İstanbul
Bilgi Üniversitesi

Transformer based Sentiment Classification

by

Devrim Mert Kivrak, 123200101

Submitted for the course

CMPE346: NATURAL LANGUAGE PROCESSING

for the assignment

Final Project

May, 2025

Abstract

In this project, I explored how five different transformer models perform on sentiment analysis tasks. The models I used were BERT, DistilBERT, RoBERTa, XLM-RoBERTa, and ALBERT. I tested each of them on three well-known datasets: IMDB, Yelp, and Amazon. For training and evaluation, I used the HuggingFace Transformers library, applying the same setup to all models to ensure fairness. Based on the accuracy and F1-score results, RoBERTa and ALBERT turned out to be the top performers. DistilBERT, on the other hand, offered decent results with less computational demand, which might be useful in practical settings. Overall, the results show how model complexity and efficiency can impact performance in text classification tasks.

Table of Contents

Abstract **ii**

1 Introduction **1**

2 Methodology **1**

3 Experiments and Findings **2**

4 Discussion **2**

5 Conclusion **3**

References **3**

1 Introduction

In this study, I focused on the fundamental NLP task of text classification, with sentiment analysis as the specific use case. I selected three commonly used sentiment classification datasets: IMDB, Yelp, and Amazon. Each dataset contains user generated reviews with associated sentiment labels, making them ideal for binary classification tasks, with the exception of Amazon dataset, where I had to fix to an extent, which I will explain later . The objective is to predict whether a given text expresses positive or negative sentiment, based on the context of the sentence.

In recent years, deep learning models such as BERT [1], DistilBERT [2], and RoBERTa [3] XLM-RoBERTa [4] and ALBERT.[5] have become standard tools for NLP classification problems. My aim is to compare multiple transformer-based models across these datasets.

2 Methodology

I used the HuggingFace `transformers` library to implement and fine-tune five transformer models: BERT, DistilBERT, RoBERTa, XLM-RoBERTa, and ALBERT. All models were loaded using the `AutoModelForSequenceClassification` class with a binary classification head (num_labels=2) , (positive/negative).

Each dataset was loaded and preprocessed using my custom `preprocessing.py` module. I applied truncation and padding using a maximum token length of 256, which is sutiable for BERT models. The training was managed via the HuggingFace `Trainer` API.

In my `train.py` module, I set the following training configuration to train my models:

- Epochs: 1
- Batch size: 16
- Save strategy: each epoch

In my implementation, I defined a general purpose model loader function that handles both classification and generation tasks.

While classification models such as BERT, RoBERTa, and DistilBERT are loaded using the AutoModelForSequenceClassification class designed for outputting discrete class labels, generation models like T5 and mBART require a different loading method.

T5 and mBART are encoder-decoder architectures originally pre-trained for text generation tasks such as translation, summarization, or question answering. As such, they are instantiated using the T5ForConditionalGeneration and MBartForConditionalGeneration classes, respectively.

This difference is important because generation models produce full textual outputs rather than single label predictions. Attempting to use them within a classification pipeline without proper handling would result in incompatible input/output expectations.

Therefore, my model loading function includes a condition that checks whether the selected model is intended for classification or generation, and loads it using the appropriate HuggingFace class.

While most datasets used in this project, such as IMDB and Yelp, consist of two columns: text and label, the Amazon dataset follows a different structure.

It contains three columns: label, title, and content. To maintain consistency with the other datasets and ensure compatibility with the classification pipeline, I preprocessed the Amazon dataset by concatenating the title and content fields into a single input text.

This preprocessing step was handled through a separate function named load_amazon_dataset(), which reformats the raw Amazon data into the standard text- label format expected by the tokenizer and training functions.

3 Experiments and Findings

I trained all five models on three seperate datasets. Below are the accuracy and F1-score results:

Model	IMDB (F1)	Yelp (F1)	Amazon (F1)
BERT	0.7085	0.8082	0.9331
DistilBERT	0.8398	0.9038	0.9205
RoBERTa	0.8898	0.8491	0.9488
XLM-RoBERTa	0.3554	0.8430	0.705
ALBERT	0.8839	0.8655	0.9533

Table 1: F1-Scores across datasets

Model	IMDB (Accuracy)	Yelp (Accuracy)	Amazon (Accuracy)
BERT	0.7114	0.8317	0.9279
DistilBERT	0.8517	0.9078	0.9138
RoBERTa	0.8898	0.8597	0.9459
XLM-RoBERTa	0.6293	0.8537	0.5451
ALBERT	0.8958	0.8798	0.9499

Table 2: Accuracy scores across datasets

Due to the relatively small size of the selected datasets (around 1000 samples for train set, and 200 samples for test set), the training process was completed within a relatively short amount of time. While this allowed me to compute results faster , it may have limited the generalization capability of the models, potentially resulting in overfitting, where models memorize training data rather than actually learning to make accurate predictions.

4 Discussion

After running the experiments, I noticed that most of the transformer models performed quite well on sentiment classification, although the results varied a bit depending on the dataset.

Out of all five models, RoBERTa and ALBERT stood out the most. They were especially strong on the Amazon dataset, reaching F1 scores of 0.9488 and 0.9533. It seems like these models are pretty good at picking up on sentiment patterns in product reviews.

DistilBERT also gave solid results. Even though it's a smaller and lighter model, it managed to do well on both IMDB and Yelp. Its performance suggests that smaller models can still be useful without giving up too much accuracy, which could make them more practical in real-world situations.

XLM-RoBERTa didn't do as well as the others, especially on IMDB, where it got an F1 of 0.3554. Its scores were a bit better on Amazon, but still behind the others. This might be because the model originally built to handle multiple languages, which may effect its performance when fine tuned on English-only datasets like the ones used here.

Another thing I also noticed that models generally performed better on the Amazon dataset. That might be because the sentiment in product reviews is more clearly stated or because the data was better balanced. In the other hand, IMDB seemed to be the most difficult, probably because the reviews there are longer and more subtle in tone.

Overall, these results suggest that how well a model performs doesn't just depend on its design, but also on the kind of data it's trained on and evaluated with.

5 Conclusion

Throughout this project, I compared five transformer models: BERT, DistilBERT, RoBERTa, XLM-RoBERTa, and ALBERT, using three different sentiment analysis datasets: IMDB, Yelp, and Amazon.

Based on the results, RoBERTa and ALBERT gave the most consistent and strong performance overall. DistilBERT also did quite well, especially considering how efficient it is in terms of speed and resource usage, which makes it a solid choice for practical use.

In the other hand, XLM-RoBERTa didn't perform as well on English only datasets, like IMDB and Amazon. It seems that its multilingual focus may have affected how well it adapted to this specific task.

Another thing I noticed was how much the dataset itself can affect the outcome. For example, models generally performed best on the Amazon dataset, while IMDB appeared to be more difficult, probably because of longer and more complex reviews.

If I were to continue this work, I'd like to try it on larger and more varied datasets, and also experiment with different hyperparameter settings to see if I can push the results further.

References

[1] Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.

[2] Sanh et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2019.

[3] Liu et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019.

[4] Lan et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. 2019.

[5] Conneau et al. Unsupervised Cross-lingual Representation Learning at Scale. 2020.