Sentiment Classification Using a Fine-Tuned BERT Language Model

Mohammed Nihad Kaipalli
Student ID:22081746
Git Hub Link
Google Colab Link

1 Introduction

Transformer-based language models have redefined the landscape of natural language processing (NLP), enabling remarkable performance on tasks such as sentiment analysis, text classification, question answering, and more. BERT (Bidirectional Encoder Representations from Transformers) is one of the most influential models in this family, pre-trained on large corpora using masked language modeling and next sentence prediction to learn deep bidirectional representations of text (Devlin et al., 2018).

In this Assignment, I fine-tuned a pre-trained BERT model on the Amazon Polarity dataset to classify customer reviews as either positive or negative. The aim was to understand the model's ability to generalize from pre-trained representations and adapt to domain-specific classification. This report details the task setup, dataset insights, training procedure, results, and key learnings.

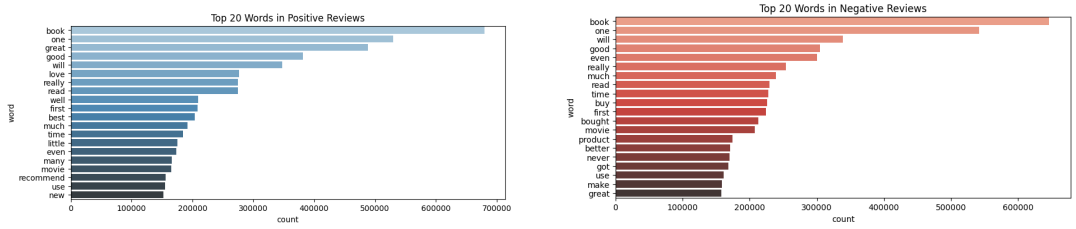
2 Task and Dataset Overview

The Amazon Polarity dataset, introduced by Zhang et al. (2015), consists of millions of product reviews labeled with binary sentiment: 0 for negative and 1 for positive. For computational efficiency, I used a smaller subset of 20,000 samples for training and 4,000 for evaluation. Each sample contains the review's content as input text and a sentiment label. The simplicity of this task makes it an ideal candidate to evaluate the effectiveness of transformer-based fine-tuning.

Before training, I performed tokenization using the bert-base-uncased tokenizer from Hugging Face's transformers library. Reviews were truncated or padded to standardize input length using dynamic padding.

3 Exploratory Data Analysis

To gain insights into the dataset's structure, I analyzed the distribution of text lengths and frequent word usage across sentiment classes. The average review length was higher for negative reviews compared to positive ones, suggesting that dissatisfied customers tend to elaborate more. Using word frequency plots and bar charts, I visualized the top 20 words in both sentiment categories after removing stopwords. Words like "waste", "disappointed", and "refund" appeared frequently in negative reviews, while "love", "excellent", and "perfect" dominated positive ones.



(a) Top 20 Words in positive reviews (b) Top 20 Words in negative reviews

Figure 1: distribution of words in both Positive & Negative Reviews

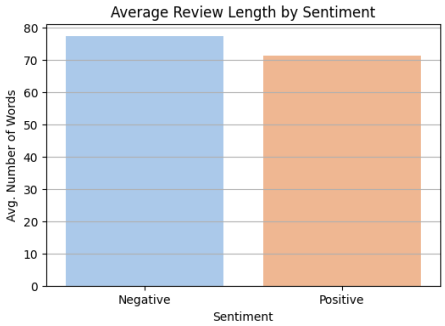


Figure 2: Average Review Length

Such patterns confirm the dataset's suitability for sentiment classification and also help in interpreting model behavior post-training.

4 Model Training

For the model, I used `BertForSequenceClassification`, initialized from `bert-base-uncased` **?**. The model architecture consists of a pre-trained encoder followed by a linear classification head for binary output. The fine-tuning was carried out over 4 epochs using the Hugging Face Trainer API. Key hyperparameters are summarized in Table 1.

Table 1: Fine-Tuning Hyperparameters

Parameter	Value
Learning rate	2×10^{-5}
Batch size	16
Epochs	4
Weight decay	0.01
Evaluation	End of each epoch
Device	GPU (if available)

Training utilized dynamic padding and was accelerated on GPU hardware. The training loss decreased steadily (see Figure 3), while validation accuracy stabilized around 94%.

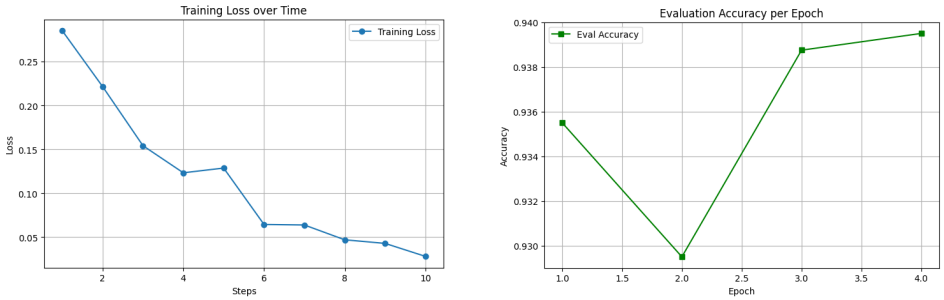


Figure 3: Training loss & Eval Accuracy.

5 Evaluation and Predictions

After training, I evaluated the model on the test subset of 4,000 reviews. The classification report yielded an overall accuracy of 94%, with precision and recall scores near or above 0.92 for both classes. The classification report is shown in Table 2.

Table 2: Classification Report on Test Set

Class	Precision	Recall	F1-Score	Support
Negative	0.95	0.91	0.93	1962
Positive	0.92	0.96	0.94	2038
Overall	0.94	0.94	0.94	4000

To test real-world generalization, I ran the model on custom-written reviews. Below are some examples with predicted outputs:

Input: "This product was amazing, exceeded my expectations!"
Prediction: Positive

Input: "Totally not worth the money, very disappointed."
Prediction: Negative

Input: "The item arrived broken and I couldn't return it."
Prediction: Negative

These predictions demonstrate the model's ability to recognize sentiment cues, even when phrased in varied ways. The model was generally confident and accurate on unlabelled, naturally phrased inputs.

6 Discussion and Future Work

The high accuracy and balanced F1-scores confirm that the model successfully learned to distinguish positive and negative sentiment from customer reviews. This success is primarily due to BERT's contextualized embeddings and deep bidirectional attention, which allow it to understand not just individual word meaning but their roles in broader contexts.

That said, there are areas where the model could be improved. First, sarcasm or mixed sentiment cases were occasionally misclassified. Introducing domain-adaptive pretraining Gururangan et al. (2020) or multi-head classification could help the model better handle such complexity. Additionally, tools like attention visualization (e.g., BertViz) could provide interpretability and help debug misclassifications.

Scaling up the training set and performing hyperparameter tuning would likely yield further improvements. Fine-tuning distilled models (e.g., DistilBERT) could offer a good trade-off between speed and accuracy for production use.

7 Conclusion

This project confirmed the power of transformer-based fine-tuning for text classification tasks. The fine-tuned BERT model achieved high performance on the Amazon Polarity dataset, providing a strong baseline for product review sentiment analysis. Continued refinements can further improve accuracy and applicability in production scenarios.

References

Devlin, J., Chang, M., Lee, K. and Toutanova, K. (2018), 'BERT: pre-training of deep bidirectional transformers for language understanding', *CoRR* **abs/1810.04805**.
URL: *http://arxiv.org/abs/1810.04805*

Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. and Smith, N. A. (2020), 'Don't stop pretraining: Adapt language models to domains and tasks', *CoRR* **abs/2004.10964**.
URL: *https://arxiv.org/abs/2004.10964*

Zhang, X., Zhao, J. J. and LeCun, Y. (2015), 'Character-level convolutional networks for text classification', *CoRR* **abs/1509.01626**.
URL: *http://arxiv.org/abs/1509.01626*