Sentiment Analysis of Google Play Store Reviews using BERT

Advait Hirlekar, Amogh Amin, Rutuja Jangle

Abstract

Sentiment analysis is a critical task in natural language processing (NLP) that offers insights by interpreting subjective information within text data. This project proposal explores the potential of using BERT (Bidirectional Encoder Representations from Transformers) to enhance sentiment classification. The project aims to investigate how BERT's contextualized embeddings, capturing deep semantic relationships, could improve the accuracy and robustness of sentiment analysis compared to traditional approaches. This proposal outlines the approach for evaluating BERT's performance in sentiment analysis, with the goal of establishing a foundation for its practical applications across various real-world text classification tasks.

Keywords: Sentiment Analysis, Natural Language Processing, BERT Transformer, Classification, Class Imbalance, Fine-Tuning.

1. Introduction

Sentiment analysis plays a crucial role in applications where understanding user feedback is essential, such as in Google Play Store reviews [4]. Analyzing the sentiment in these reviews provides developers and stakeholders with insights into user satisfaction and app performance, helping to guide improvements based on user feedback. Traditional methods for sentiment analysis, such as Support Vector Machines (SVM) and Naive Bayes, have proven effective to an extent; however, they struggle to handle the complex, nuanced language often present in user-generated content [5]. These earlier models generally rely on bag-of-words or word embeddings techniques that lack deep contextual understanding, making them less effective in capturing sentiment from informal, app-specific language.

Recent advancements in NLP, particularly BERT, address these limitations. BERT's bidirectional training approach allows it to understand context by capturing semantic relationships across an entire sentence rather than processing text in a unidirectional manner [1]. This contextual depth is invaluable for analyzing sentiment in diverse, complex user reviews. By applying BERT to Google Play Store app reviews, this project aims to overcome the limitations of traditional models, enhancing the accuracy of sentiment classification.

2. Methods

2.1 Data Collection and Preprocessing:

The dataset used in this study consists of user reviews from the Google Play Store App. It includes three sentiment categories: Positive, Neutral, and Negative. To prepare the data for

analysis, reviews were preprocessed using the Natural Language Toolkit (NLTK) [2]. This involved tokenizing the text, converting all tokens to lowercase, removing punctuation and stop words, and applying lemmatization to reduce words to their base forms [7]. This preprocessing pipeline ensured uniformity and removed noise from the data, improving the quality of inputs to the BERT model. The cleaned reviews were tokenized and encoded using the BERT tokenizer, which converts the text into input IDs and attention masks compatible with the model.

Figure 1 shows the initial distribution of the Positive, Negative and Neutral Sentiments in the Google Play Store Reviews dataset after all the preprocessing steps were applied to the dataset.

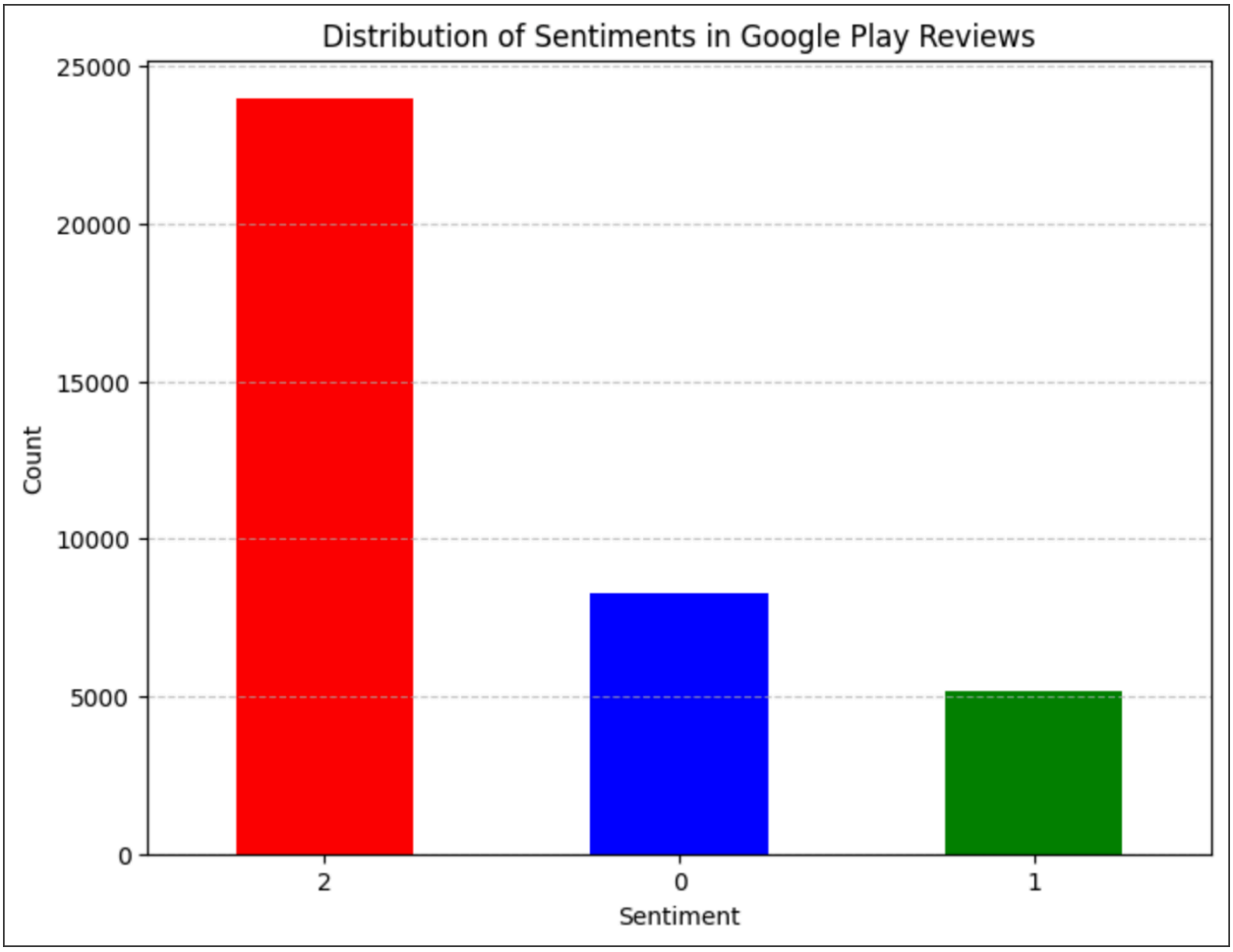

Figure 1. Original Distribution of Sentiments in Google Play Store Reviews Dataset

2.2 Model Architecture:

The pre-trained BERT model (bert-base-uncased) was fine-tuned for sentiment classification. The architecture includes a transformer-based encoder that generates contextual embeddings for input text [3]. A classification head, consisting of a fully connected layer, was added to BERT's

output to map the contextual embeddings to three sentiment classes: Positive, Neutral, and Negative. The pre-trained weights were retained for the encoder, while the classification head's weights were initialized and optimized during fine-tuning [1].

2.3 Weighted Loss Function:

Given the imbalance in sentiment distribution having a higher prevalence of Positive reviews, class weights were computed to penalize errors for underrepresented classes more heavily. These weights were incorporated into the CrossEntropyLoss function, ensuring balanced learning [6]. Specifically, weights were inversely proportional to the frequency of each class, assigning higher importance to Neutral and Negative classes. This adjustment mitigated the risk of the model being biased toward the majority class.

2.4 Training Procedure:

The dataset was split into training (80%) and validation (20%) sets to evaluate model performance on unseen data. The BERT model was fine-tuned over 10 epochs with a learning rate of 5e-5. The AdamW optimizer was employed for gradient updates, and a linear learning rate scheduler was used to gradually reduce the learning rate during training. The data was batched using PyTorch's DataLoader, with a batch size of 16 to optimize memory usage. Each training step involved forward propagation to compute predictions, backward propagation to compute gradients, and optimizer steps to update model weights. The weighted loss function was applied at each step to ensure balanced contributions from all sentiment classes.

2.5 Evaluation Metrics:

Model performance was evaluated using standard metrics: accuracy, precision, recall, and F1-score. These metrics provided a comprehensive view of the model's effectiveness across all sentiment classes. Additionally, a confusion matrix was generated to visualize class-wise predictions and misclassifications. This analysis helped identify specific areas where the model performed well and where improvements were needed, such as distinguishing Neutral from Positive sentiments.

3. Results

3.1 Training Performance:

The model exhibited steady improvement during training, as indicated by the reduction in loss over 10 epochs in Figure 2. Initially, the training loss was approximately 2.5 over 3 epochs, reflecting the model's difficulty in making accurate predictions at the start. The training accuracy was 96% denoting that the model was overfitting as it did not accurately classify new instances

and the loss value further proved it's inefficiency. So, after fine-tuning when we selected 10 epochs, by the final epoch the loss had dropped to near-zero levels, demonstrating effective convergence and the model's ability to learn meaningful patterns in the data. This consistent decline highlights the model's stability and robust optimization during fine-tuning. This fine-tuned balanced model was then tested against the new instances and it accurately classified them into their respective sentiments thus denoting that the overfitting problem was solved.

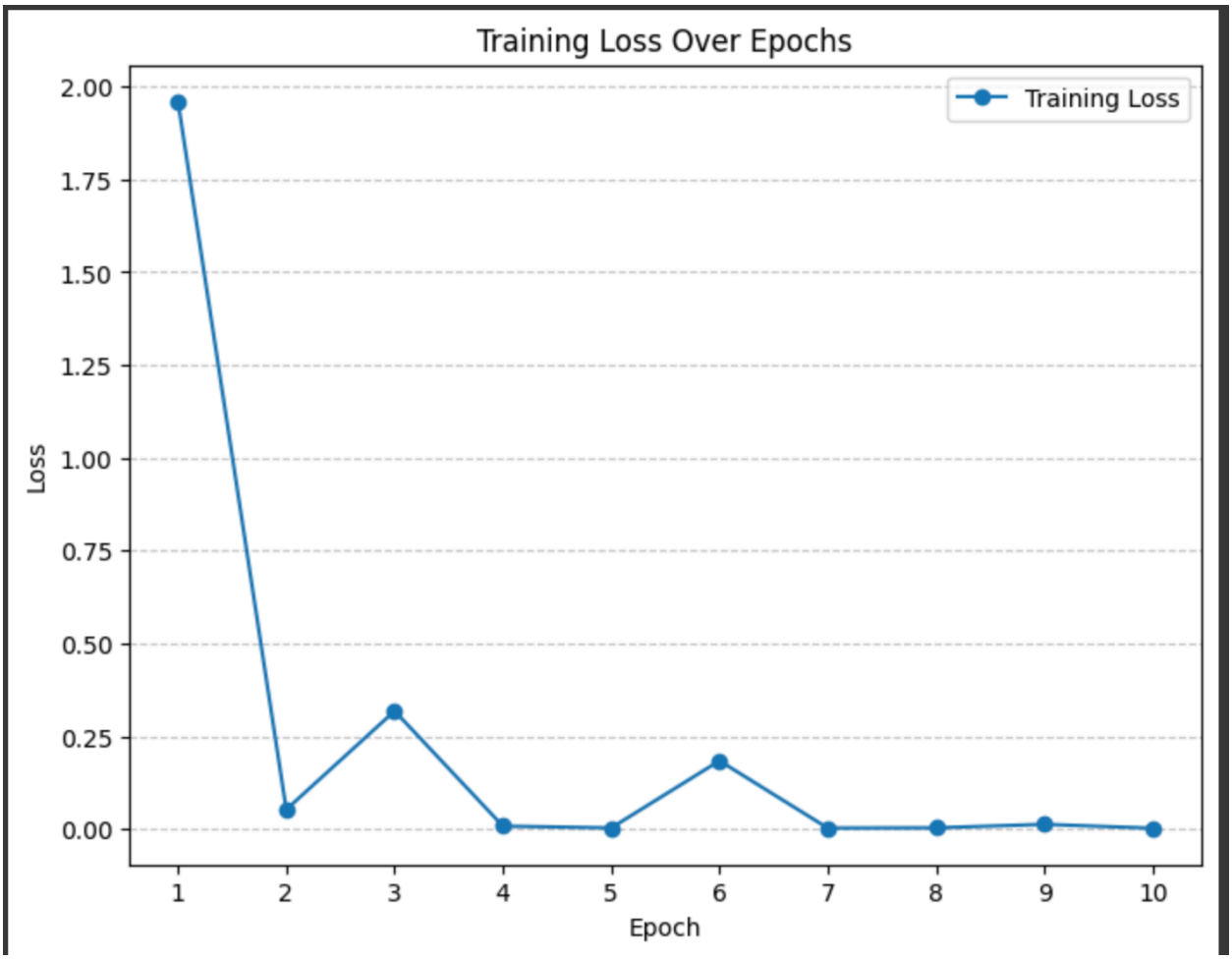

Figure 2. Training Loss Curve

3.2 Sentiment Distribution and Weights:

To address the imbalanced sentiment distribution in the dataset, class weights were computed based on the frequency of each sentiment class. Positive sentiment, being the most frequent, received the lowest weight, while Neutral sentiment, being the least frequent, received the highest weight. The adjusted weights ensured fairer contributions during training, improving the model's performance on underrepresented classes as shown in Figure 3. This weighting

mechanism played a crucial role in balancing the model's predictions across all sentiment categories.

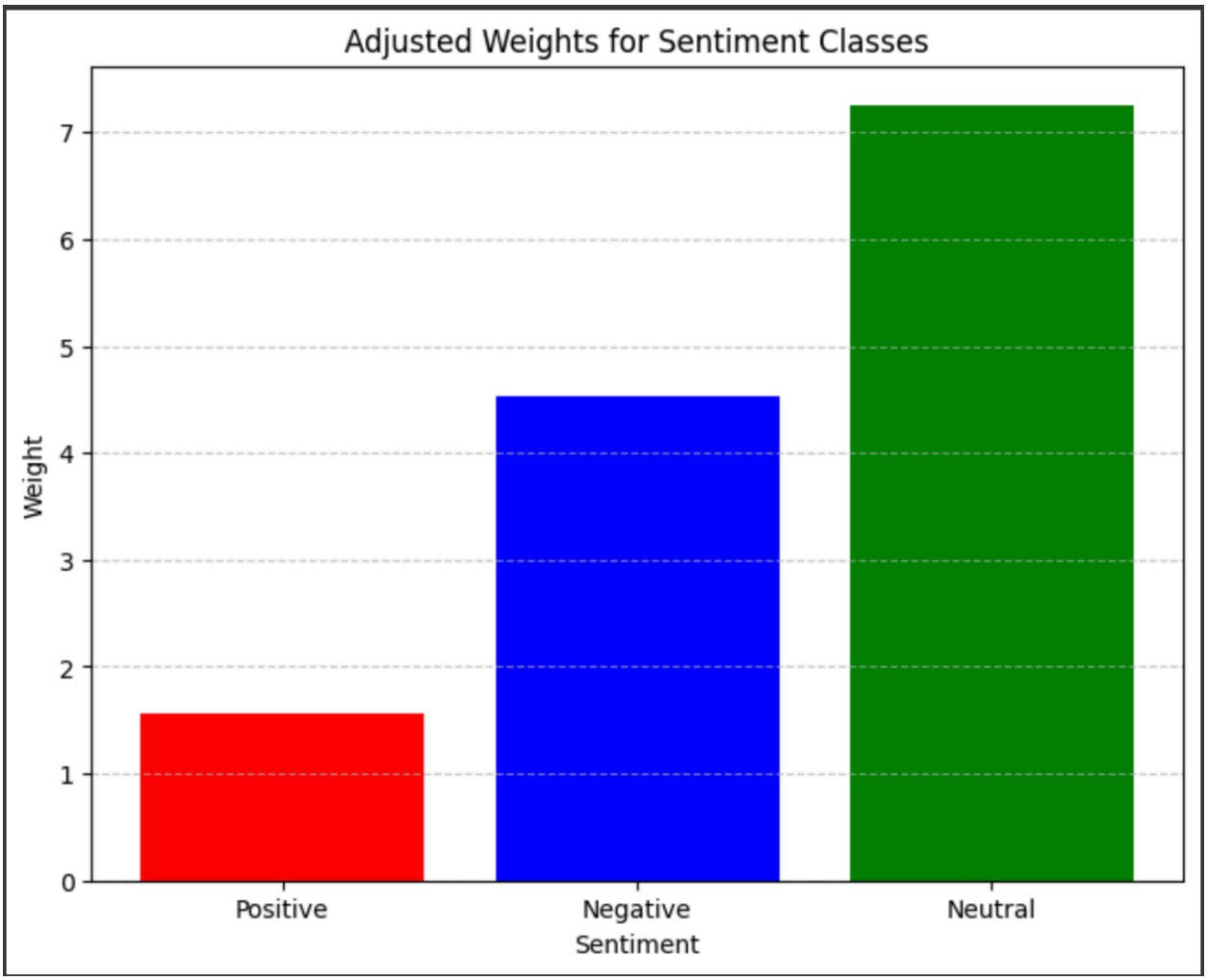

Figure 3. Adjusted Weights after fine-tuning the model.

3.3 Model Evaluation:

The fine-tuned BERT model achieved outstanding performance on the validation set, with an overall accuracy of 94%. Detailed metrics for each sentiment class are presented below in Figure 4 showing the classification report.

Negative Sentiment: Precision = 0.89, Recall = 0.90, F1-Score = 0.89 (Support = 1653 reviews)
Neutral Sentiment: Precision = 0.92, Recall = 0.92, F1-Score = 0.92 (Support = 1049 reviews)
Positive Sentiment: Precision = 0.96, Recall = 0.96, F1-Score = 0.96 (Support = 4784 reviews)

Classification Report:

	precision	recall	f1-score	support
Negative	0.89	0.90	0.89	1653
Neutral	0.92	0.92	0.92	1049
Positive	0.96	0.96	0.96	4784
accuracy			0.94	7486
macro avg	0.92	0.92	0.92	7486
weighted avg	0.94	0.94	0.94	7486

Figure 4. Classification Report

3.4 Confusion Matrix Analysis:

The confusion matrix provides deeper insights into the model's predictions. The majority of Positive reviews were classified correctly, with minimal spillover into the Neutral or Negative categories. Neutral reviews, despite being less frequent, were predicted with high precision and recall, reflecting the effectiveness of class weighting. Negative reviews showed slight misclassification as Neutral, likely due to the subtle overlap in language used in some reviews. These observations underscore the model's strong capability while also revealing areas for improvement. The macro-averaged precision, recall, and F1-score were 92%, demonstrating the model's balanced performance across all sentiment classes. The confusion matrix in Figure 5 highlights the model's accuracy in classifying Positive sentiments, with minor misclassifications observed between Neutral and Negative sentiments.

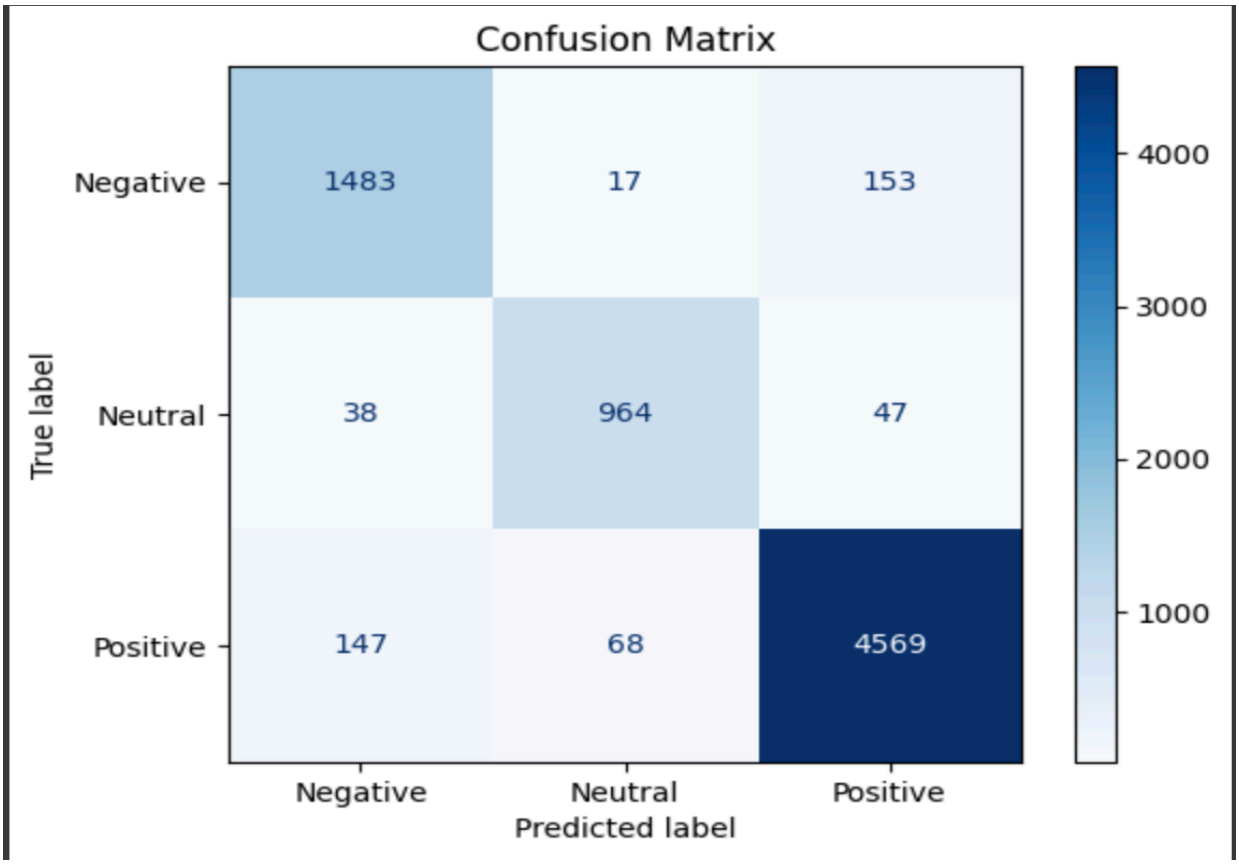

Figure 5. Confusion Matrix showing the final results.

4. Discussion

The high accuracy and F1-scores achieved by the fine-tuned BERT model demonstrate its effectiveness in handling nuanced sentiment expressions in user-generated content. The use of class weighting was instrumental in addressing the dataset's imbalance, enabling the model to perform well across all sentiment categories. Fine-tuning of the BERT model also helped in overcoming the overfitting problem by adjusting the class weights and balancing all the three sentiments equally for classification. The inclusion of preprocessing steps, such as lemmatization and stop word removal, further enhanced the model's ability to process informal, app-specific language effectively.

Limitations:

Despite its strong performance, the model struggled with ambiguous or sarcastic reviews. These cases often require additional contextual understanding or specialized features, such as sarcasm detection. Another limitation was the dataset's domain-specific nature, which may reduce the model's generalizability to reviews from other platforms or industries.

Future Work:

Sarcasm Detection: Incorporating sarcasm detection as a secondary task could help the model better handle subtle or implicit negative sentiments.

Enhanced Pre-training: Experimenting with larger pre-trained models, such as RoBERTa or DeBERTa, could further improve performance, especially on challenging examples.

Dataset Expansion: Collecting and incorporating reviews from diverse app categories or platforms would improve the model's robustness and generalizability.

5. Conclusion

This study highlights the efficacy of BERT for sentiment analysis of Google Play Store reviews. By leveraging its bidirectional training and contextual embedding capabilities, BERT demonstrated superior performance compared to traditional models, achieving an accuracy of 94% and macro-averaged F1-score of 92%. The incorporation of a weighted loss function addressed class imbalance effectively, enabling balanced performance across Positive, Neutral, and Negative sentiment categories.

The results underscore BERT's ability to handle complex, informal, and app-specific language in user reviews, making it a valuable tool for developers and stakeholders seeking to understand user feedback. While the model performed well, certain limitations, such as handling sarcasm and ambiguous language, remain. These limitations open avenues for future research, including the development of multi-task learning models and the exploration of larger, domain-specific datasets.

In conclusion, this work provides a robust framework for sentiment analysis using BERT, paving the way for its practical applications in analyzing user-generated content across various domains. The insights gained from this study contribute to the growing body of research on transformer-based NLP models and their real-world applications.

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT 2019 (pp. 4171–4186). Association for Computational Linguistics.

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[3] Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. Neural Information Processing Systems.

[4] Googleplaystore_user_reviews.csv. Retrieved from https://www.kaggle.com/code/gallo33henrique/llm-engineering-prompt-sentiment-analysis/input?select=googleplaystore_user_reviews.csv

[5] Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Machine Learning: ECML-98, 137–142.

[6] Bakırarar, Batuhan & ELHAN, Atilla. (2023). Class Weighting Technique to Deal with Imbalanced Class Problem in Machine Learning: Methodological Research. Turkiye Klinikleri Journal of Biostatistics. 15. 19-29. 10.5336/biostatic.2022-93961.

[7] SM Mazharul Islam, Xin Dong, and Gerard de Melo. 2020. Domain-Specific Sentiment Lexicons Induced from Labeled Documents. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6576–6587, Barcelona, Spain (Online). International Committee on Computational Linguistics