Sentiment Analysis. Optimism-pessimism identification

Boboc George
george.boboc@s.unibuc.ro

Dana Mihai-Razvan
mihai-razvan.dana@s.unibuc.ro

Dirva Nicolae
nicolae.dirva@s.unibuc.ro

Panait Ana-Maria
ana.panait1@s.unibuc.ro

Abstract

This study explores sentiment analysis to identify optimistic and pessimistic sentiments in IMDb movie reviews using machine learning techniques. We preprocess reviews for analysis, employ TF-IDF for feature extraction, and compare multiple models for sentiment classification including Multinomial Naive Bayes, Feed Forword Neural Network, Simple Recurrent Neural Network, Long Short Term Memory Network, Stacked Long Short-Term Memory Network, Universal Sentance Encoder and transformers (fine tunning pretrained model from HuggingFace Model Hub). Our results indicate that using the pretrained model from HuggingFace is most effective, achieving superior performance in accuracy and F1-score. This research highlights the potential of text classification techniques to discern complex emotional expressions in user-generated content, offering insights beneficial to the entertainment industry.

1 Introduction

Sentiment analysis is an increasingly critical task in natural language processing, as it helps to discern the emotional undertones conveyed within vast amounts of text data. This research focuses specifically on identifying optimistic and pessimistic sentiments in IMDb movie reviews. The ability to automatically classify these sentiments is significant for applications ranging from enhancing recommendation systems to understanding consumer behavior and even to monitoring mental health trends. Our choice to explore this topic was driven by the nuanced challenge it presents: optimism and pessimism are complex emotions that can subtly influence and reflect public perceptions and decisions.

The choice of IMDb reviews as our data set was intentional; as a rich source of expressive and opinionated content, these reviews provide a robust platform for analyzing sentiments.

In our opinion, the advancement of sentiment analysis techniques, particularly through deep learning and transformer models, holds substantial promise for enhancing our understanding of complex emotional expressions in text. By identifying not just whether a sentiment is positive or negative but exploring the shades of optimism and pessimism within, we can offer more nuanced insights into the emotional responses of individuals, which is of great value to sectors like the entertainment industry, marketing, and beyond. This paper presents our findings and discusses the implications of employing advanced sentiment analysis models to tackle this intricate task.

2 Related Work

Sentiment analysis has progressively evolved from simplistic lexicon-based methods to sophisticated deep learning techniques. This evolution is marked by significant milestones in both methodological approaches and application domains.

Early Developments: The foundational work by Pang et al. (Pang et al., 2002) demonstrated the effectiveness of machine learning techniques in sentiment analysis, specifically using SVMs for classifying movie reviews. This study set the stage for further exploration of statistical models in sentiment analysis.

Advancements with Deep Learning: The shift towards deep learning was notably marked by the adoption of recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks. Hochreiter and Schmidhuber's (Hochreiter and Schmidhuber, 1997) introduction of LSTMs addressed the challenges of long-range dependencies in text, a crucial aspect for understanding sentiment. These models have been extensively applied to sentiment analysis, significantly improving the ability to capture contextual nuances compared to earlier techniques.

Transformers and BERT: The introduction of

the transformer architecture by Vaswani et al. (Vaswani et al., 2017) represented a paradigm shift in NLP. This was quickly followed by the development of BERT (Bidirectional Encoder Representations from Transformers) by Devlin et al. (Devlin et al., 2018), which further refined the approach by leveraging bidirectional training of transformers. This model has set new state-of-the-art benchmarks for a variety of NLP tasks, including sentiment analysis.

More recent studies have focused on fine-tuning and adapting pre-trained models like BERT for specific sentiment analysis tasks. For instance, Sun et al. (Sun et al., 2019) demonstrated how fine-tuning BERT on sentiment analysis tasks could achieve state-of-the-art results across several benchmarks. Additionally, the exploration of multimodal sentiment analysis, which integrates textual, audio, and visual data, has opened new research avenues. Xia et al. (Li et al., 2020) highlighted how cross-modal attention mechanisms could enhance sentiment detection in multimodal contexts.

Emerging Trends: The exploration of zero-shot and few-shot learning paradigms, as discussed by Brown et al. (Brown et al., 2020) with the introduction of GPT-3, showcases the potential of performing sentiment analysis with minimal training data. These approaches are particularly promising for languages and domains where labeled data is scarce.

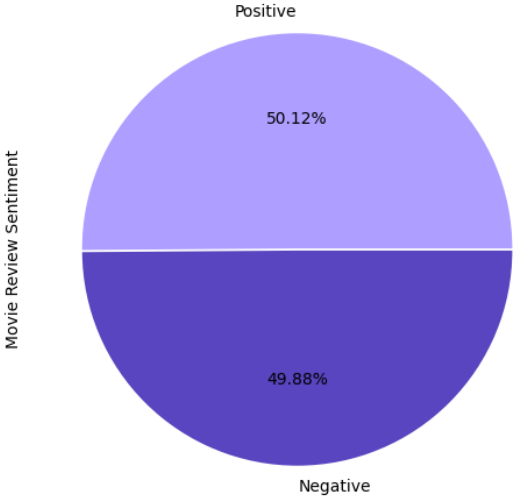

Figure 1: Distribution of target

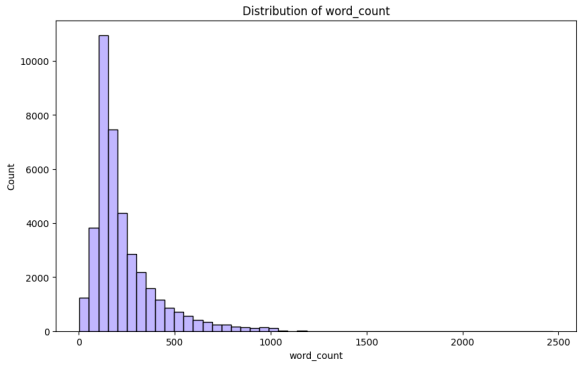

Figure 2: Distribution of Various word counts

3 Method

This section details the methods used in our study to identify optimistic and pessimistic sentiments in IMDb movie reviews. We preprocessed the reviews, utilized TF-IDF for feature extraction, and employed various machine learning models for sentiment classification. Our approach compared these methodologies to establish which is most effective for nuanced sentiment classification. We discussed each method's advantages and disadvantages, focusing on factors like computational efficiency, model complexity, and performance metrics such as accuracy and F1-score. This comparative analysis helps highlight the suitability of each model for specific applications within sentiment analysis.

3.1 Dataset

The dataset used in this study is the "IMDB Movie Ratings Sentiment Analysis" dataset (H., 2018), sourced from Kaggle. This dataset comprises 40,000 rows and 2 columns, where each row represents a movie review along with its associated sentiment label (positive or negative). The dataset contains 277 duplicate values which need to be removed to ensure the quality and integrity of the analysis. There are no missing values (NaN) in the dataset, which simplifies the preprocessing step.

The dataset is balanced, meaning that the number of positive and negative reviews is approximately equal as shown in Figure 1. The balanced nature of the dataset ensures that the model will not be biased towards either positive or negative sentiments. The distribution of the word count in the reviews is right-skewed and contains many outliers as shown in Figure 2. Most reviews are relatively short, with a few very long reviews. This skewness and the presence of outliers need to be considered when preprocessing the text data.

3.2 Preprocessing

To prepare the IMDB movie reviews dataset for sentiment analysis, several preprocessing steps were applied. These steps were chosen to standardize the text data, reduce noise, and improve the performance of the sentiment analysis model. The preprocessing methods implemented are as follows:

- **Removing Line Breaks and Special Characters**, ensuring noise and dimensionality reduction
- **Tokenization**, ensuring enhanced text analysis and feature extraction
- **Converting to Lowercase**, reducing the vocabulary size and preventing the model from learning case-sensitive patterns
- **Removing Stopwords**, enhancing its ability to identify sentiment-laden terms and thus improving its accuracy
- **Lemmatization**, further reducing the vocabulary size and help the model generalize better by treating different forms of a word as the same token
- **Removing Duplicates**, preventing overfitting to repeated data and ensuring a more diverse training set

These preprocessing techniques make the data more manageable and reliable, making it easier to spot patterns and trends. Additionally, we standardized the text by converting it to lowercase and ensured uniform input lengths through padding.

3.3 Models

Following the preprocessing steps, we proceeded with model training and evaluation to analyze the sentiment of IMDb reviews.

- **Multinomial Naive Bayes (Model 0)**
  - We used a Multinomial Naive Bayes model to define the baseline accuracy.
  - We obtained an accuracy score of 85.84%, precision 86%, recall 86% and F1 score of 86%.
- **Feed-Forward Neural Network (Model 1)**
  - The initial approach in our sentiment analysis study involved a feed-forward neural network.
  - This model was designed using TensorFlow and Keras, employing a sequence of text preprocessing, vectorization, embedding, and dense layers. The architecture is as follows:
    - **Input Layer**: Accepts input in the form of text strings.
    - **Text Vectorization**: Transforms text input into integer sequences, capping vocabulary to the top 10,000 words while marking out-of-vocabulary words with an <oov> tag.
    - **Embedding Layer**: Maps the integer sequences to 128-dimensional vectors.
    - **Global Average Pooling**: Averages out the embeddings to reduce the dimensionality and to handle variable input sizes.
    - **Dropout (15%)**: A dropout layer to prevent overfitting by randomly setting input units to 0 during training at a rate of 15%.
    - **Output Layer**: A dense layer with sigmoid activation function for binary classification.
  - **Hyperparameters** were set as follows:
    - Batch size: 32
    - Epochs: 5
    - Optimizer: Adam
    - Loss Function: Binary Crossentropy
  - **Hyperparameter tuning** was conducted to optimize model performance. This involved varying the learning rate of the Adam optimizer, adjusting the dropout rate, and experimenting with different sizes for the embedding dimension. Each configuration was evaluated based on its impact on validation accuracy and loss, helping identify the most effective settings for our specific dataset.
  - The model's performance **metrics** on the test dataset were 89.71% accuracy, 90% precision, 90% recall, and an F1 score of 0.90. The relatively high accuracy suggests either a model well-suited to the dataset characteristics.
  - In comparison to state-of-the-art (SOTA) models in sentiment analysis, our approach integrates less complexity than

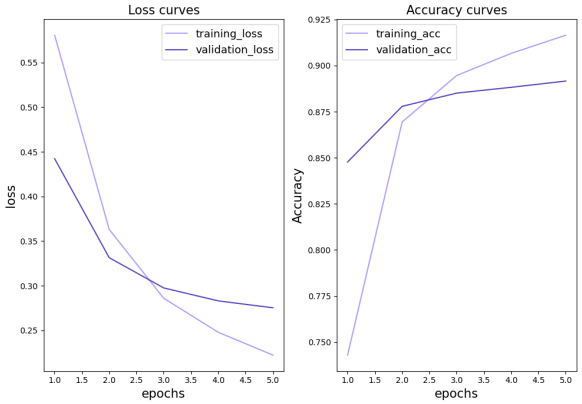

Figure 3: Loss and accuracy for model 1

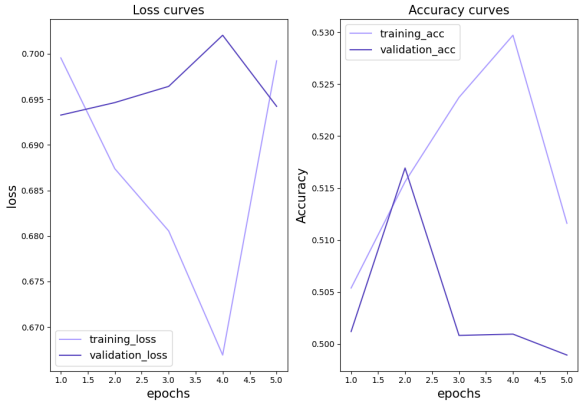

Figure 4: Loss and accuracy for model 2

recurrent neural networks (RNNs) and transformer models, which are known for their efficacy in capturing sequential dependencies and contextual nuances in text. However, the simplicity of feed-forward networks can be advantageous for faster training times and lower resource consumption, making them suitable for applications where real-time performance is critical.

- **Recurrent Neural Network (Model 2)**
  - We also implemented a recurrent neural network (RNN) to explore how well these architectures perform with sequential data, such as text. The RNN model incorporates a SimpleRNN layer that is particularly suited to capturing the temporal dependencies within the input text.
  - The architecture consists of the following layers:
    - **Input Layer**: Accepts input in the form of text strings.
    - **Text Vectorization**: Similar to the feed-forward network, it converts text inputs into a sequence of integers.
    - **Embedding Layer**: Maps the integer sequences to 128-dimensional vectors.
    - **SimpleRNN Layer**: A SimpleRNN with 32 units processes the sequence data, maintaining internal states that capture temporal information.
    - **Dropout (20%)**: To reduce overfitting, a dropout of 20% is applied after the RNN layer.
    - **Output Layer**: A dense layer with sigmoid activation function for binary classification.
  - The performance **metrics** for this model were as follows:
    - Accuracy: 49.89%
    - Precision: 48%
    - Recall: 50%
    - F1 Score: 35%
  - The model was compiled using the Adam optimizer and binary crossentropy as the loss function. The primary metrics for evaluation were accuracy, precision, recall, and F1 score. It was trained over 5 epochs with a batch size of 32, and both the training and validation datasets were processed in a similar manner as the feed-forward model.
  - These results indicate significantly lower performance compared to the feed-forward model. The accuracy close to 50% suggests that the RNN model might be performing no better than random guessing on this particular task. Several factors could contribute to this outcome:
    - The SimpleRNN's limitations in handling long sequences might lead to vanishing gradients.
    - There is insufficient network depth or complexity to capture the intricacies of the sentiment analysis.
    - Overfitting despite the dropout, possibly due to not adequately capturing the dependencies or the complexity of the input data.
- **Long Short-Term Memory Network**

(Model 3)

- Our third model in the sentiment analysis series employs a Long Short-Term Memory (LSTM) network, which is designed to overcome the limitations of traditional RNNs by better handling long-range dependencies in text data. The model integrates an LSTM layer to effectively capture both short-term and long-term dependencies within the input sequences.
- The model architecture is as follows:
  - **Input Layer**: Takes raw text strings as input.
  - **Text Vectorization**: Converts text to sequences of integers.
  - **Embedding Layer**: Translates integer sequences into 128-dimensional embeddings.
  - **LSTM Layer**: A 64-unit LSTM layer processes the sequence data, designed to retain information over longer sequences and reduce the impact of gradient vanishing.
  - **Dropout (10%)**: A dropout layer follows the LSTM layer to help mitigate overfitting by randomly omitting 10% of the units in the learning phase.
  - **Output Layer**: A dense layer with a sigmoid activation function for binary classification.
- The performance **metrics** for this model were as follows:
  - Accuracy: 50.08%
  - Precision: 50%
  - Recall: 50%
  - F1 Score: 36%
- This model was compiled with the Adam optimizer and binary crossentropy loss. It was trained over 5 epochs with a batch size of 32, maintaining consistency in training conditions across models to facilitate direct comparison. The metrics used to evaluate the model included accuracy.
- The LSTM model demonstrated gradual improvements in both training accuracy and loss across the epochs, indicating a positive learning trajectory. However, the validation results displayed an increase in loss and a plateau in accuracy, suggesting some overfitting to the training data despite the use of dropout.

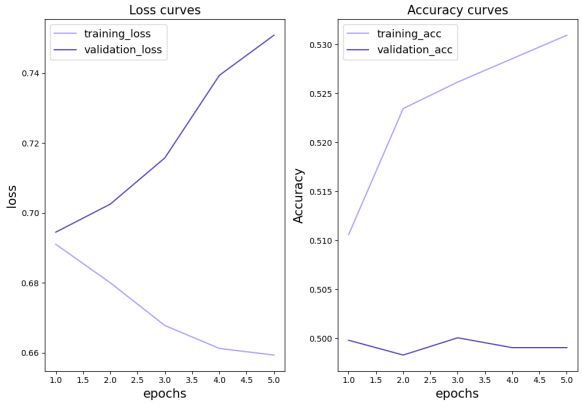

Figure 5: Loss and accuracy for model 3

- **Stacked Long Short-Term Memory Networks (Model 4)**
  - Our fourth model in the sentiment analysis project is a stacked Long Short-Term Memory (LSTM) network, designed to enhance the model's capacity to learn from complex data structures by adding multiple LSTM layers. This configuration aims to leverage the strengths of deep learning to capture more nuanced patterns in the data that may not be accessible to single-layer LSTM models.
  - The detailed architecture includes:
    - **Input Layer**: Receives text as input.
    - **Text Vectorization**: Converts text to sequences of integers.
    - **Embedding Layer**: Translates integer sequences into 128-dimensional embeddings.
    - **First LSTM Layer (64 units)**: Processes the sequence with return sequences set to True, allowing the next LSTM layer to receive sequence output rather than final state output.
    - **Dropout (20%)**: Applied after the first LSTM to reduce overfitting by randomly omitting 20% of the units.
    - **First LSTM Layer (32 units)**: Further processes the output from the first LSTM layer, designed to refine the understanding of the input data.

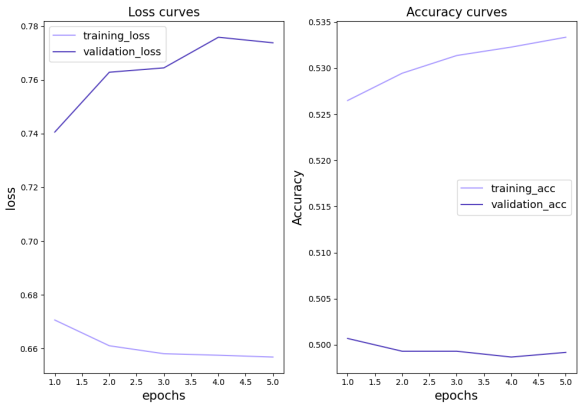

Figure 6: Loss and accuracy for model 4

  - ∗ **Dropout (20%)**: Another dropout layer follows the second LSTM layer to enhance the model's ability to generalize.
  - ∗ **Output Layer**: A dense layer with a sigmoid activation function for binary classification.
- Despite the theoretical advantages of stacked LSTMs in handling complex patterns, model 4 reported:
  - ∗ Accuracy: 49.92%
  - ∗ Precision: 49%
  - ∗ Recall: 50%
  - ∗ F1 Score: 36%
- These metrics suggest a performance level similar to that of simpler models, indicating potential issues such as:
  - ∗ Inadequate training data size or quality to effectively train deeper network architectures.
  - ∗ Possible overfitting despite the use of dropout layers, perhaps due to too high complexity in relation to the dataset's nuances.
  - ∗ Suboptimal hyperparameters, including the number of LSTM units and the rate of dropout.

• **Universal Sentence Encoder Model (Model 5)**

- Model 5 utilizes the Universal Sentence Encoder (USE), a powerful pre-trained model from TensorFlow Hub, designed to convert text into high-dimensional vectors. These vectors are then used to perform various tasks such as text classification, semantic similarity, clustering, and more. For our sentiment analysis, the USE model is leveraged as the initial layer in our network, followed by dense layers to refine and adapt the embeddings for specific sentiment classification.
- The architecture includes:
  - ∗ **Universal Sentence Encoder Layer**: Pre-trained model that encodes text into 512-dimensional vectors. This layer is initially set as non-trainable to preserve the pre-trained weights.
  - ∗ **Dense Layer (64 units, ReLU activation)**: A fully connected layer that introduces the ability to learn non-linear combinations of the features.
  - ∗ **Output Layer (Sigmoid activation)**: Produces the final prediction for binary classification.
- The model was initially compiled using the Adam optimizer with standard parameters and binary crossentropy loss. The primary metrics evaluated were accuracy, precision, recall, and F1 score. It was trained for 5 epochs with the dataset batched as per previous models.

  In its initial training phase, where the USE layer was not trainable, Model 5 achieved:
  - ∗ Accuracy: 86.65%
  - ∗ Precision: 87%
  - ∗ Recall: 87%
  - ∗ F1 Score: 87%
- These results suggest that the USE-based model performs robustly in capturing sentiment, likely benefiting from the broad and rich linguistic patterns learned during the pre-training on diverse text data.
- To further enhance the model's performance, the USE layer was set to trainable, allowing the embeddings to adapt more specifically to our sentiment analysis task. The optimizer's learning rate was reduced to 1e-4 to make smaller updates, which is crucial when fine-tuning to avoid overwriting the learned representations significantly. Additionally, callbacks for early stopping and learning rate reduction were employed to optimize training without overfitting:

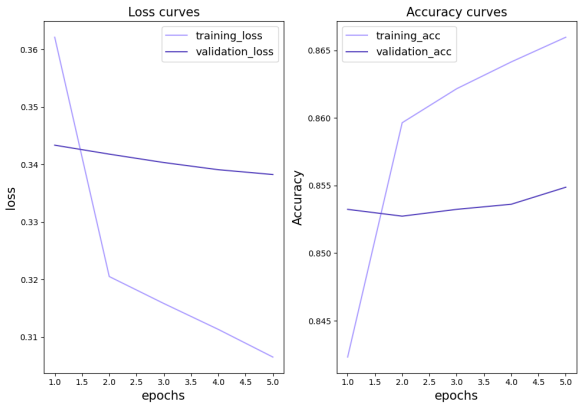

Figure 7: Loss and accuracy for model 5

  - ∗ **Early Stopping**: Monitors validation loss and stops training if no improvement is seen for 4 epochs, restoring weights from the best iteration.
  - ∗ **Reduce LR on Plateau**: Reduces the learning rate by a factor of 0.2 if no improvement in validation loss is observed for 2 consecutive epochs, with a minimum delta change of 0.001, preventing stagnation in training progress.

- **Transformers: Fine-Tuning a Pretrained Model (Model 6)**
  - For our sixth model in the sentiment analysis project, we used a transformer-based approach using a pretrained model from Hugging Face's Model Hub. Transformers are renowned for their state-of-the-art performance in numerous NLP tasks due to their ability to capture contextual information from all parts of a text simultaneously. This approach utilizes the "bert-base-uncased" model, adapted for binary sequence classification.
  - Hyperparameters and Training:
    - ∗ Epochs: 3
    - ∗ Batch Size: 8
    - ∗ Learning Rate Schedule: A polynomial decay schedule was employed to decrease the learning rate from an initial value of 5e-5 to 0 by the end of the training.
  - The model was compiled with a binary crossentropy loss function designed for logits (pre-sigmoid outputs), an Adam optimizer with the aforementioned learning rate schedule, and accuracy as the metric.
  - The transformer model underwent fine-tuning over a subset (20%) of the training data, significantly enhancing its ability to contextualize and classify sentiments more accurately. Due to the computational intensity and efficiency considerations, only a portion of the available training dataset was used, which is a common practice when fine-tuning large models.

4 Conclusion

This project has provided valuable insights into the complexities of sentiment analysis and the diverse range of techniques available for tackling text classification tasks, and various machine learning models, from traditional algorithms like Naive Bayes to advanced deep learning models like LSTM and Transformers. We enjoyed advancing throught the models and seeing the accuracy going up as we were learning.
However, some aspects of the project were challenging, starting with finding a good enough, well balanced dataset, and finishing with optimizing hyperparameters for deep learning models.

5 Future Work

Implementing aspect-based sentiment analysis would have been an intriguing extension of the project. This approach involves analyzing the sentiment not just at the overall review level but also at a deeper level, focusing on specific aspects or components of the movie that contribute to the overall sentiment. For example, identifying whether the sentiment expressed in a review is positive towards the acting but negative towards the plot, or vice versa.

Sentiment analysis models like the ones developed in this project have a wide range of potential applications beyond just movie reviews. For example companies can use sentiment analysis to monitor social media platforms in real-time to understand public opinion about their products or services. This could help them identify emerging trends, detect customer issues, and gauge overall brand sentiment.

Ethical Statement

- Potential Unethical Uses of Research:
  - **Manipulation of Public Opinion**: The technology could generate biased reviews or posts.
  - **Invasion of Privacy**: Applying sentiment analysis without consent.
  - **Discrimination**: Reinforcing existing biases in various domains.
- Potential Biases in Our Work:
  - **Dataset Bias**: Cultural, gender, or racial biases in the reviews.
  - **Model Bias**: Learning and propagating biases from training data.
- Measures to Combat Bias and Unethical Uses:
  - **Bias Detection and Mitigation**: Balanced dataset and diverse representation.
  - **Transparency and Accountability**: Clear documentation of data sources and methodologies.
  - **Privacy Considerations**: Use of public data and anonymization.
  - **Ethical Review**: Ensured compliance with ethical guidelines.
- Sentiment analysis has significant potential but must be developed and applied ethically, prioritizing fairness, transparency, and privacy.

Limitations

Although our sentiment analysis project demonstrates promising results in classifying movie reviews as positive or negative, it is important to acknowledge its limitations. One significant constraint is that, like any other model, the performance can never be fully accurate, meaning there will always be instances where the sentiment analysis misclassifies a review. This inherent limitation underscores the importance of interpreting the model's predictions with caution and considering them as probabilistic estimates rather than definitive judgments.

Additionally, while our models are effective for short text inputs like movie reviews, they may encounter scalability issues when applied to longer texts, such as articles or essays. Moreover, the computational resources required for training and inference, particularly the reliance on large GPU resources, may pose challenges for adoption in resource-constrained environments. Addressing these limitations would inspire further investigation and refinement of our methods, ultimately enhancing the applicability and robustness of sentiment analysis across diverse linguistic and textual contexts.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Journal or Conference Title (replace this placeholder with actual source)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Google AI Language.

Yasser H. 2018. Imdb movie ratings sentiment analysis. https://www.kaggle.com/datasets/yasserh/imdb-movie-ratings-sentiment-analysis. Accessed: 2024-05-13.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yueying Li, Hui Cao, Xiaotian Xia, and Quan Song. 2020. Multi-modal sarcasm detection via crossmodal attention mechanism. *Journal or Conference Title (replace this placeholder with actual source)*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-2002 Conference on Empirical Methods in Natural Language Processing*, volume 10, pages 79–86. Association for Computational Linguistics.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *Journal or Conference Title (replace this placeholder with actual source)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.