Sentiment Analysis on Movie Reviews using machine learning models

Group 23: Zheng Qiao, Dongxiao Li, Bingyang Ke, Haoran Guo

1. Introduction

Sentiment analysis, which is also called opinion mining, is the most exhaustive task in Natural Language Processing (NLP) that intends to identify the sentiment mentioned in a specific text. This project focuses on the evolution of sentiment classification in the domain of movie reviews with a target of classifying reviews as positive, negative, or neutral. The primary problem of sentiment analysis is context vagueness along with utilizing domain-specific language, which would influence the performance of classification. To cope with these difficulties, people commit to the accomplishment of several machine learning models, from which we include traditional linear models like Logistic Regression and also deep learning architectures like Multi-Layer Perceptrons (MLP), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM), to more modern and data-saved models such as BERT. Our measurements will be based on accuracy, F1-score, and the swiftness of training, which will be studied through the use of ablation to analyze the impact of different model architectures, hyperparameter tuning, and feature representations. The outcomes of this study will guide developers in finding the right approach that will be effective for sentiment analysis and, in a way, select which one is the most efficient model for real industry needs.

2. Dataset

We will use the IMDB movie review dataset, a widely-used benchmark collection for sentiment analysis. The dataset contains 50,000 movie reviews, evenly divided between training and testing sets (25,000 each). This public dataset is available on Kaggle under the name "IMDB Dataset of 50K Movie Reviews."

Link:
https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

3. Methodology

We will use Python machine learning libraries such as NLTK, Scikit-learn, and PyTorch to develop a sentiment analysis model that identifies the emotional tone of the text and classifies the result as positive or negative. For feature extraction, the TF-IDF matrix, Positive Pointwise Mutual Information matrix, Word2Vec embeddings, and N-grams will be used for capturing deep semantic information and word relationships from the dataset. Real-world data is always high-dimensional. Therefore, we will apply dimensionality reduction techniques, including Principal Component Analysis and Singular Value Decomposition, before training. We will experiment with several machine learning models, such as Recurrent Neural Network(RNN), Long Short-Term Memory(LSTM), Support Vector Machine(SVM), Bidirectional Encoder Representations from Transformers(BERT), and compare their performance. The models will be evaluated based on metrics such as F1-score, Accuracy, and Precision. T-SNE plots in 2D space will help to illustrate the result, Word Cloud will display common words associated with positive, negative, and neutral tones, and confusion matrices will analyze the classification results.

3.1 RNN Model

This model uses a recurrent neural network (RNN) to classify the sentiment of movie reviews.

We used the RNN model to classify the sentiment of IMDB movie reviews. As a classic method for sequence modeling, RNN can model temporal order information in natural language processing. This model explored the performance of basic RNN in sentiment analysis tasks and compared it with more complex models to observe its limitations when processing long texts.

We built a model based on PyTorch that includes an embedding layer, a single-layer RNN, and a fully connected output:

1. Data preprocessing: text is unified into lowercase, punctuation and special symbols are removed, encoded using a vocabulary, and padded to a uniform length.
2. Model structure: Embedding → RNN → Linear, where the RNN unit outputs the hidden state of the last time step for classification.
3. Training settings: Adopting the cross entropy loss function and the Adam optimizer, training for 5 cycles, and a batch size of 64.

Results and Evaluation:

RNN with learning rate = 0.001, embedding size = 128, hidden size = 128:

- **Loss and Accuracy**:

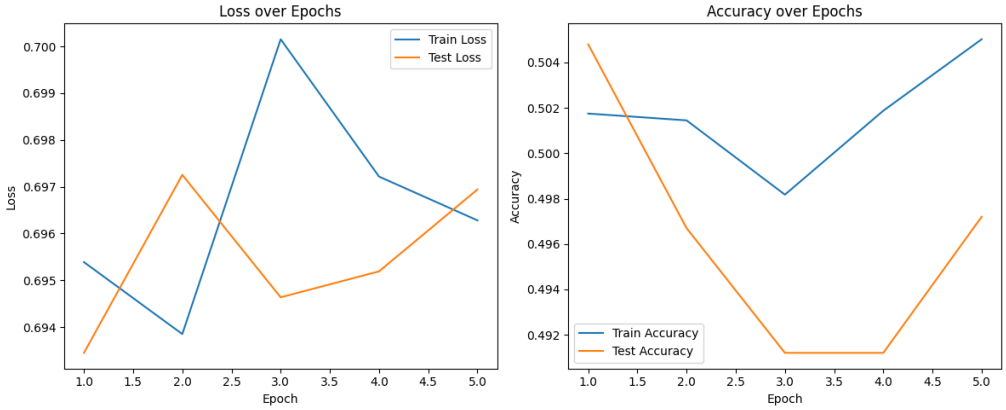
  Epoch 1/5
  Train Loss: 0.6954, Accuracy: 0.5018
  Test Loss: 0.6934, Accuracy: 0.5048

  Epoch 2/5
  Train Loss: 0.6939, Accuracy: 0.5014
  Test Loss: 0.6973, Accuracy: 0.4967

  Epoch 3/5
  Train Loss: 0.7002, Accuracy: 0.4982
  Test Loss: 0.6946, Accuracy: 0.4912

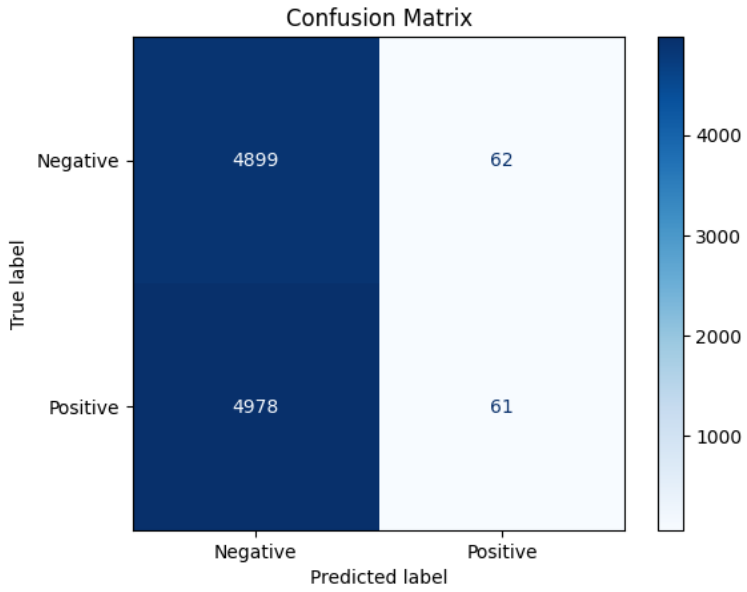
  Epoch 4/5
  Train Loss: 0.6972, Accuracy: 0.5019
  Test Loss: 0.6952, Accuracy: 0.4912

  Epoch 5/5
  Train Loss: 0.6963, Accuracy: 0.5050
  Test Loss: 0.6969, Accuracy: 0.4972



The initial parameters for RNN are with learning rate = 0.05, embedding size = 64, hidden size = 128. The model's prediction is around 50% accuracy indicates the random classification. In addition, the loss does not decrease while training. So I decrement the learning rate to 0.001 to avoid gap steps while learning, and increment the dimension to 128 to capture more features. However, the model's performance does not improve.

- **Confusion matrix**:



- **Scores Report**:

	precision	recall	f1-score	support
Negative	0.4960	0.9875	0.6603	4961
Positive	0.4959	0.0121	0.0236	5039

The recall score indicates the model predicts labels are almost negative and failing to detect positive samples. F-1 score shows the bias toward negative class. In addition, the precision score around 0.5 suggests random.

Conclusions

RNN can process sequence information and is suitable for text classification tasks, but it has the problem of "vanishing gradient" when processing long texts, which affects the model's learning of long-distance dependencies. Although the accuracy is acceptable, it does not perform as well as LSTM or BERT.

3.2 LSTM Model

The LSTM (Long Short-Term Memory) model in the recurrent neural network is used to perform sentiment classification tasks on the IMDB movie review dataset.

We use the LSTM model to classify the sentiment of movie reviews. LSTM is a model proposed to solve the long-term dependency problem of traditional RNN and is suitable for long text modeling. In sentiment analysis tasks, LSTM can capture semantic changes more accurately and has practical application value.

The model structure consists of an embedding layer, a single-layer LSTM, and a linear output:

1. Data preprocessing: Same as RNN, first clean the text and then convert it into an integer index sequence and unify the length;
2. Model structure: Embedding → LSTM → Linear, LSTM adopts a unidirectional structure, and outputs the last hidden state as the basis for classification;
3. Training settings: 5 rounds of training, using the Adam optimizer, and Dropout to prevent overfitting.

Compared with RNN, LSTM significantly improves generalization ability and reduces overfitting.

Results and Conclusions

LSTM with learning rate = 0.005, embedding size = 128, hidden size = 128:

- **Loss and Accuracy**:

Epoch 1/5
Train Loss: 0.6942, Accuracy: 0.5006
Test Loss: 0.6935, Accuracy: 0.5040

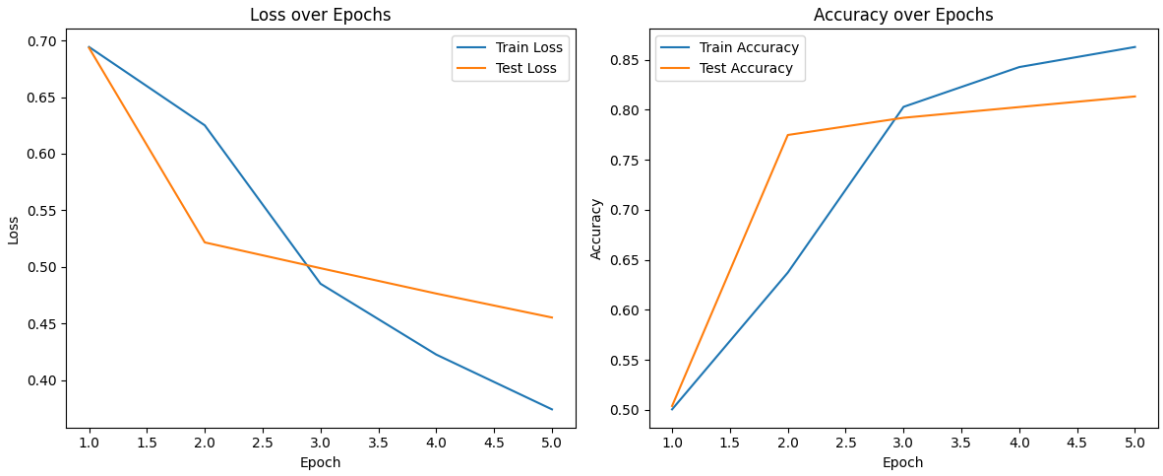
Epoch 2/5
Train Loss: 0.6250, Accuracy: 0.6373
Test Loss: 0.5217, Accuracy: 0.7747

Epoch 3/5
Train Loss: 0.4852, Accuracy: 0.8028
Test Loss: 0.4989, Accuracy: 0.7919

Epoch 4/5
Train Loss: 0.4226, Accuracy: 0.8426
Test Loss: 0.4765, Accuracy: 0.8026

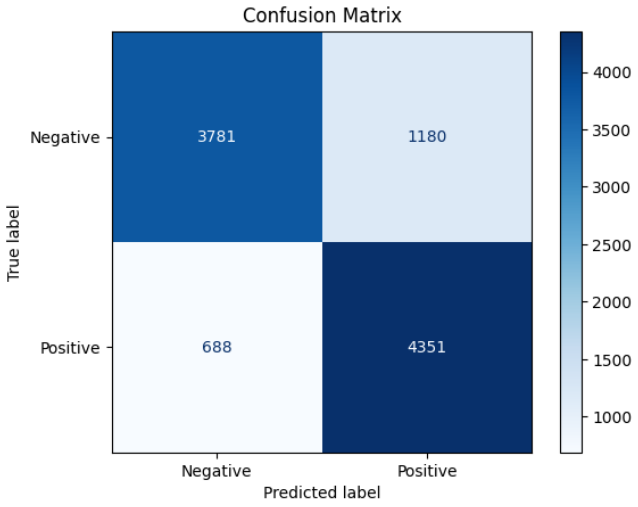
Epoch 5/5
Train Loss: 0.3742, Accuracy: 0.8626

Test Loss: 0.4553, Accuracy: 0.8132



The LSTM model showed a significant improvement in both accuracy and loss after the first complete pass through the entire dataset, and its performance continued to improve with each following epoch. The loss consistently decreased for both the training and test sets, while the accuracy improved to 86% on the training set and 81% on the test set. The relatively small difference between training and test performance indicates that the model generalizes well and it is not overfitting.

- **Confusion matrix**:



- **Scores Report**:

	precision	recall	f1-score	support
Negative	0.8461	0.7621	0.8019	4961
Positive	0.7867	0.8635	0.8233	5039

The precision score is a little higher for predicting “negative” label. The recall score indicating the model recovers more positive examples. And the F1-score is above 0.8 suggests that for both classes, the performance is strong balanced.

Conclusions
LSTM can capture longer-range text dependencies and improve the accuracy of sentiment classification. Its gating mechanism effectively alleviates the gradient vanishing problem

Why using RNN and LSTM?
Simple classification models such as traditional SVM model will ignore the sequential relationships between words. Therefore, we apply sequential model RNN to extract the sequence information of the text and preserve the information across time steps, allowing the model to better at capturing semantic meaning.
However, because of the gradient vanishing problem, RNN cannot remember information over long sequences. As a result, we apply the LSTM model to make a further improvement. LSTM model uses a gating mechanism that allows the model to choose whether to retain or discard information, thus effectively preserves important content across long dependencies, and providing better performance in solving complex language patterns.

3.3 SVM Model

This code aims to build a sentiment classification model based on text features, and use support vector machines (SVM) to distinguish the positive and negative emotions of movie reviews. The whole process includes data preprocessing, feature extraction, dimension compression, model training and evaluation, and visual analysis.

We used support vector machines (SVM) to classify movie reviews into two categories. SVM is a traditional machine learning method that is suitable for small and medium-sized data, has a clear structure, and is fast to train. It is an important control experiment for neural network models.

The process includes two parts: feature extraction and classification modeling:
1. Text vectorization: Use TF-IDF to encode the cleaned text and convert it into a sparse matrix representation;
2. Model structure: Use sklearn's linear kernel SVM for binary classification training;
3. Training settings: 5000 training sets, 3000 test sets, and use default parameters for model fitting.

Results and Evaluation:

Accuracy: The model achieved an overall accuracy of 85.71%. This indicates that the classifier correctly predicts the sentiment of a movie review approximately 86 out of 100 times. This level of accuracy is generally good and suggests that the model is well-fitted to the data.

Precision and Recall:

Negative Reviews:

Precision: 0.86 - This score tells us that when the model predicts a review as negative, it is correct 86% of the time.

Recall: 0.85 - This means that the model successfully identifies 85% of all actual negative reviews.

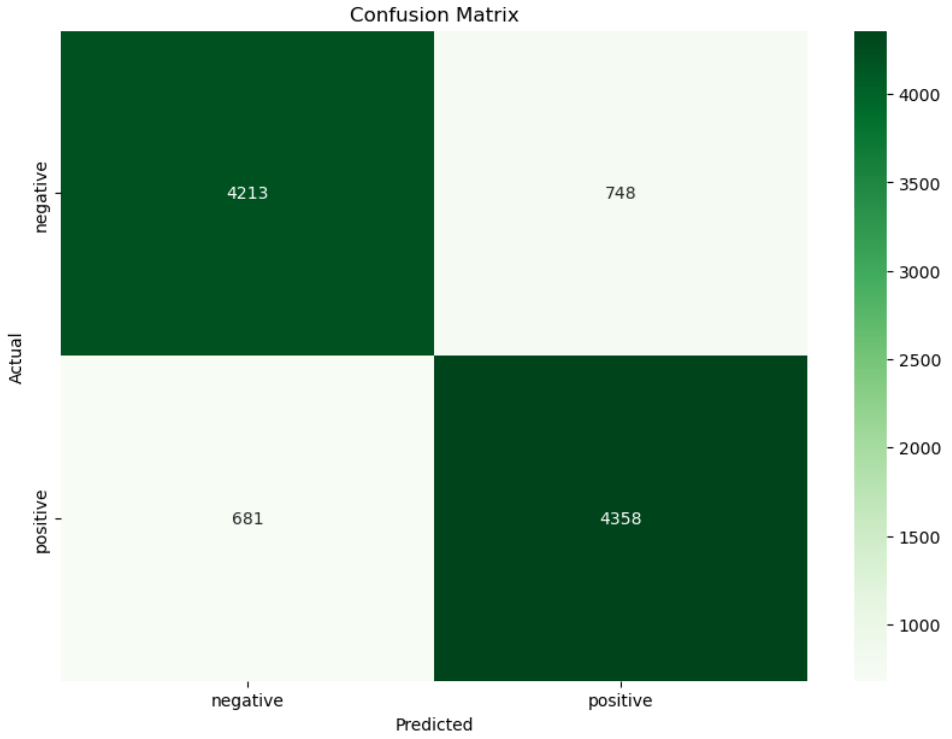
Positive Reviews:

Precision: 0.85 - Indicates that the model's predictions of positive reviews are correct 85% of the time.

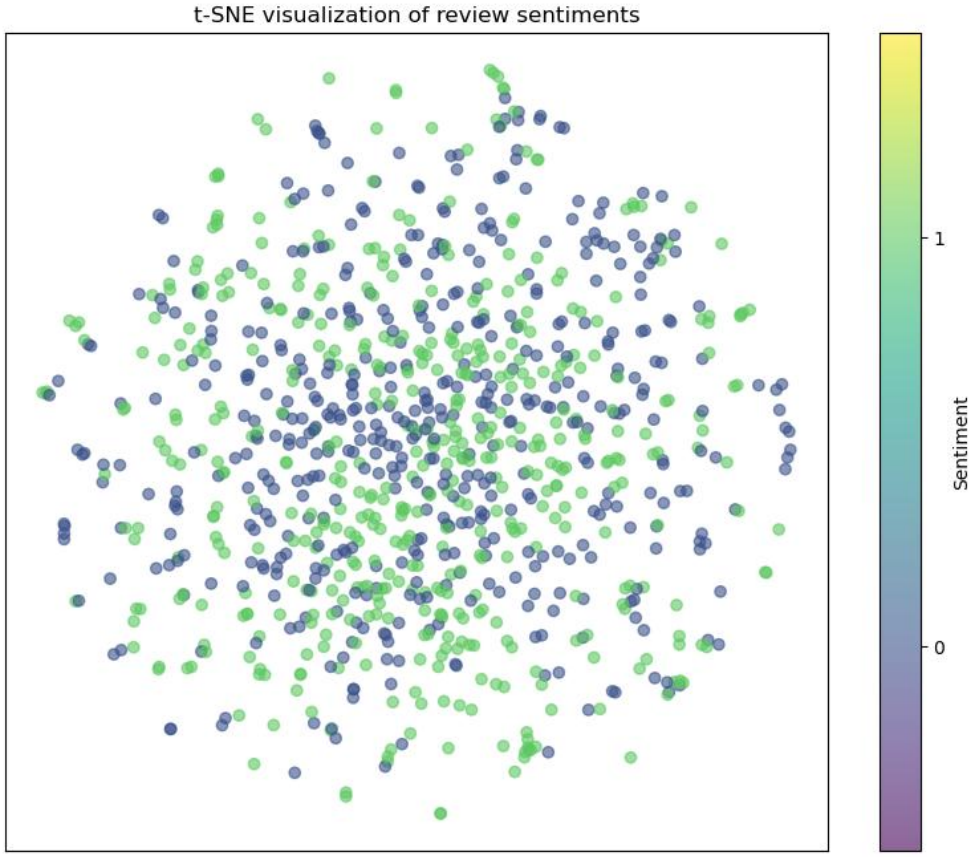
Recall: 0.86 - Shows that the model identifies 86% of all actual positive reviews.

F1 Score: The F1 scores for both categories are around 0.85 to 0.86, which suggests a balanced performance between precision and recall. This balance is important in scenarios where both the costs of false positives and false negatives are nearly equal.

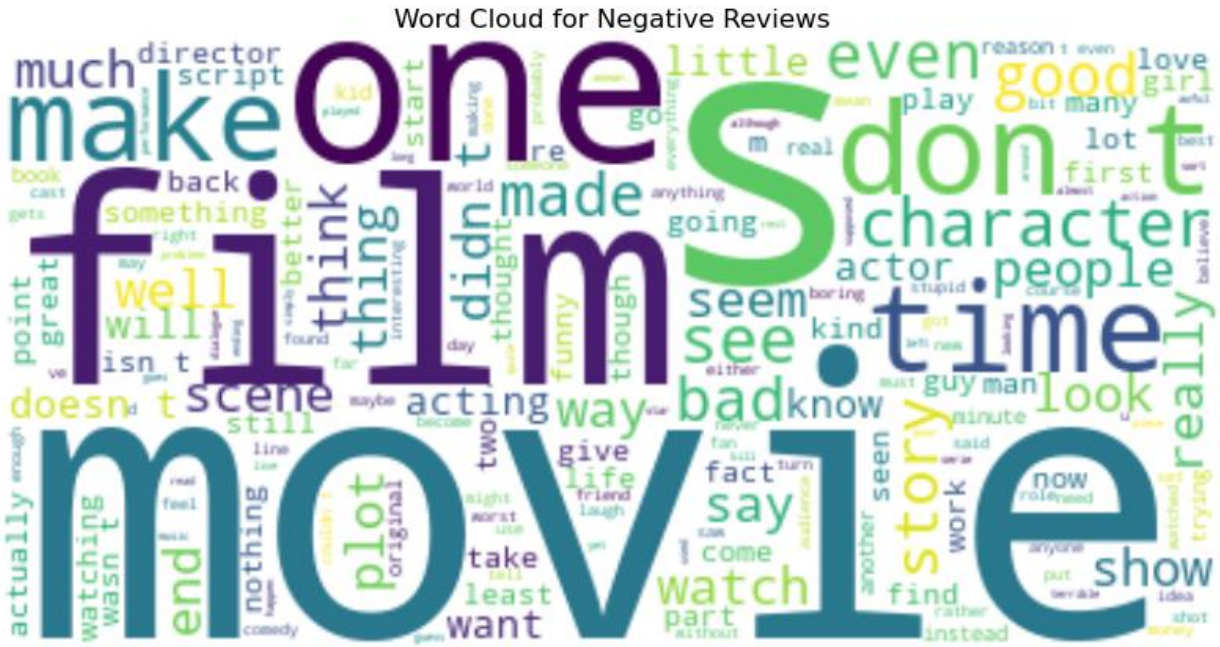
Confusion Matrix



T-SNE Visualization



WordCloud



Analysis

Balanced Performance: The model shows a relatively balanced performance in terms of precision and recall across both sentiment classes. This is ideal in cases where both types of errors (false positives and false negatives) have similar consequences.

Implications of Metrics:

The similar scores in precision, recall, and F1 across both categories suggest that the model does not exhibit a bias toward either sentiment class despite the balanced dataset.

The balanced support numbers (number of instances in each class) confirm that the model was trained and evaluated on an evenly distributed dataset, which helps in achieving an unbiased evaluation.

Potential Improvement

1. Feature Enhancement: While the current feature extraction techniques (likely TF-IDF and possibly Word2Vec) are effective, incorporating more contextual embeddings from models like BERT or GPT could capture deeper semantic meanings, potentially improving recall and precision further, especially for complex sentences or sarcasm.

2. Advanced Modeling Techniques: Experimenting with different or more sophisticated machine learning algorithms, such as ensemble methods or deep learning approaches, could further refine the model's predictions, especially in capturing nonlinear relationships.

3. Error Analysis: Conducting a thorough error analysis by examining cases where the model predictions were wrong could provide insights into specific areas where the model struggles, such as certain linguistic nuances or less frequent expressions of sentiment.

4. Cross-Validation: Implementing more rigorous validation techniques, like k-fold cross-validation, could provide a more robust estimate of model performance and stability across different subsets of data.

Conclusion

Overall, the model demonstrates strong performance in sentiment analysis with an ability to fairly and accurately classify both positive and negative reviews. This capability makes it a valuable tool for automatic sentiment analysis tasks, potentially aiding in business decisions, content filtering, and consumer insights analysis. Further refinements and exploration of advanced techniques could elevate its performance, making it an even more effective tool in natural language processing tasks.

3.4 BERT Model

This study uses a BERT-based sentiment classification model to perform binary classification modeling (positive/negative) on the IMDB movie review dataset. The overall process is divided into five stages: data preprocessing, text encoding, model building, training and evaluation.

We used the BERT model to classify the sentiment (positive/negative) of movie reviews. With the increasing popularity of pre-trained language models, applying BERT to sentiment classification tasks can not only improve accuracy, but also analyze the performance of the model in text understanding. We selected the IMDB dataset and completed an end-to-end experimental process through steps such as text cleaning, tokenization, encoding, modeling, training, and visualization.

We built a binary classifier structure based on the pre-trained bert-base-uncased model:

1. Data preprocessing: remove HTML tags and punctuation, and unify to lowercase.
2. Tokenization and encoding: Use BertTokenizerFast to tokenize the text, unify the length, and build the input tensor.
3. Model structure: Add Dropout and linear layers on the pooled vector (pooler_output) output by BERT to output logits for two categories.
4. Training settings: 5000 training data and 3000 test data, Adam optimizer, cross entropy loss function, training for 2 rounds.

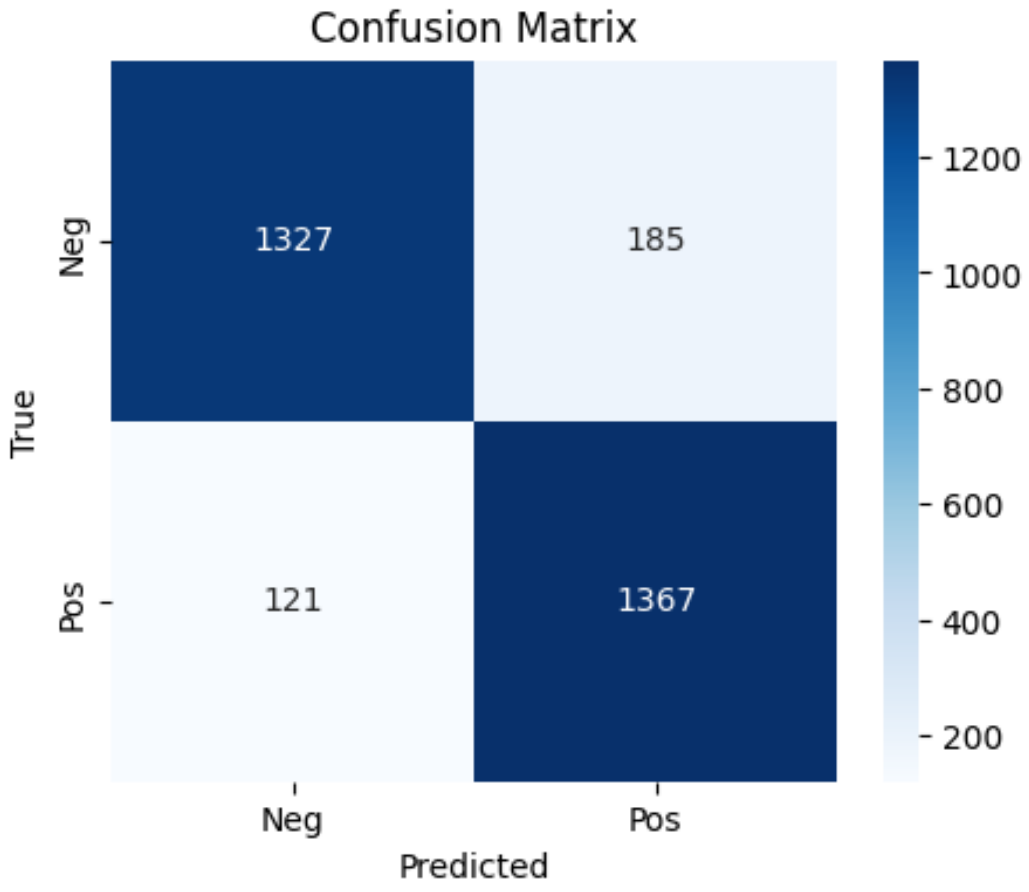
In addition, through the confusion matrix and T-SNE dimensionality reduction map, we further verified the model's ability to distinguish between the two types of comments.

Results & Evaluation

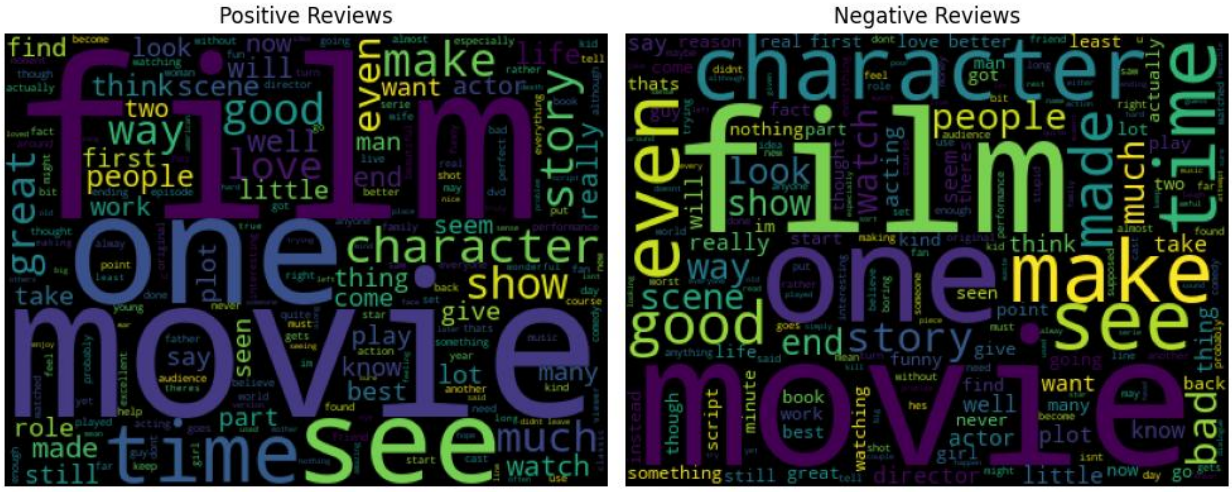
We mainly use accuracy, F1 score, precision and recall for evaluation, and the evaluation focuses on the medium-sized subset:

Accuracy is about 94.42%
Precision is 88.08%
Recall is 91.87%
F1 score is 89.93%

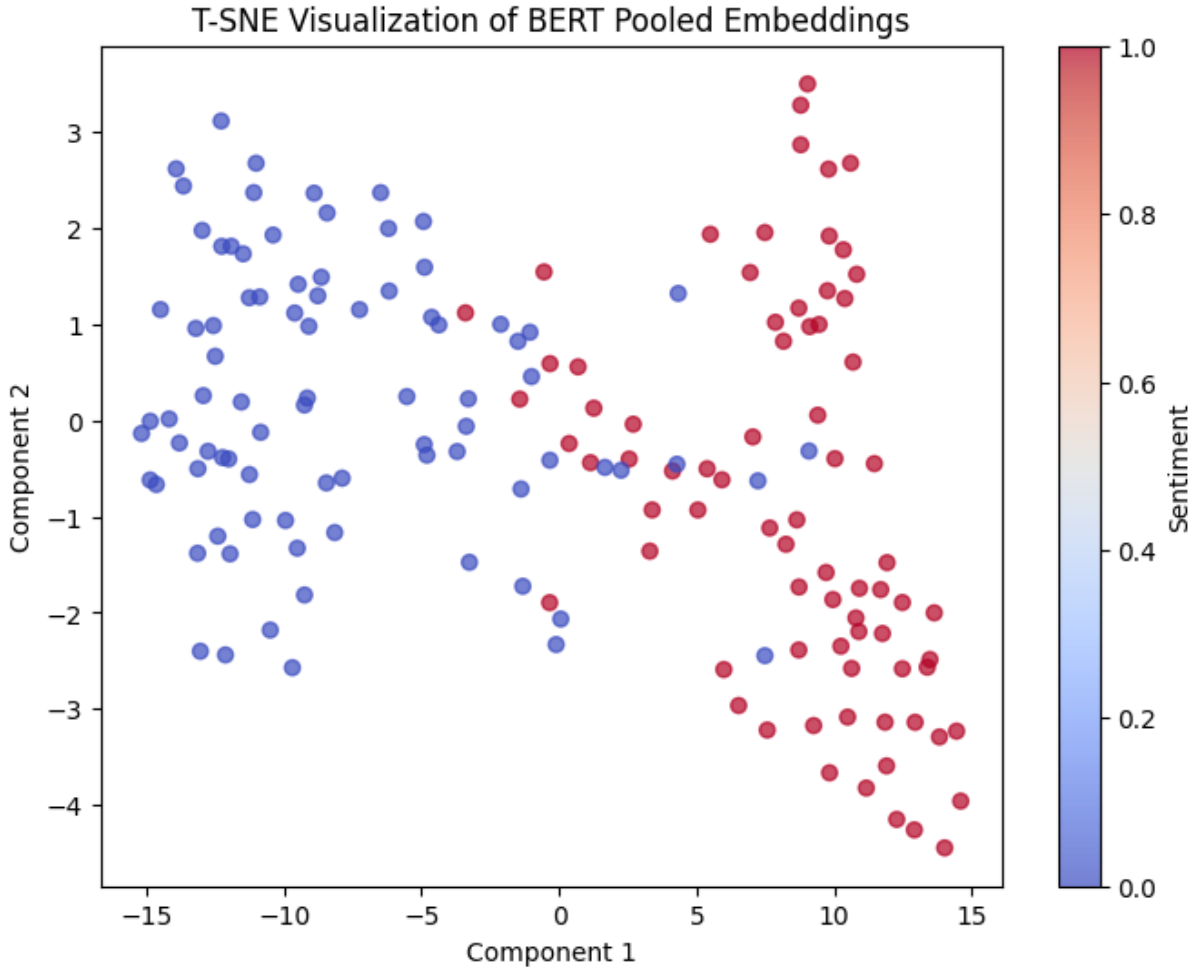
Confusion Matrix



WordCloud



T-SNE Visualization



Analysis

This BERT implementation demonstrates the significant advantage of transformer-based architectures in sentiment analysis. It satisfies the project proposal's emphasis on handling vague contextual meaning and achieving high-quality classification results, while providing robust visual explanations and metrics.

Through the implementation of BERT on the IMDB 50K movie review dataset, we observed that:

1. The model achieved high accuracy and F1-scores, even when trained on a small subset of the dataset, which confirms the efficiency of pre-trained transformers in capturing sentiment features.
2. WordCloud visualizations provided intuitive insight into the frequent positive and negative tokens, aligning well with sentiment distributions.
3. T-SNE visualization of BERT embeddings illustrated that sentiment-labeled features cluster distinctly in reduced dimensions, reinforcing that BERT captures contextual meaning effectively.

4. Summary

This project systematically compares four sentiment classification models based on the IMDB movie review dataset: SVM, RNN, LSTM, and BERT. Each model represents a stage in the evolution of natural language processing technology.

1. As a traditional machine learning model, SVM achieved an accuracy of 85.71% after combining TF-IDF features, with stable performance and balanced classification. It has a high reference value as a benchmark model.

2. Although the RNN model is theoretically suitable for sequence modeling, it performs poorly in long text tasks due to the gradient vanishing problem, with an accuracy of about 50%, which is basically random guessing, reflecting its limitations.

3. The LSTM model effectively alleviates the problems of RNN by introducing a gating mechanism, significantly improving the model performance, with a test set accuracy of 81.3%, showing stronger generalization ability and stability.

4. As a representative of pre-trained language models, the BERT model achieved the best results in this project, with an accuracy of 94.42% and an F1 value close to 90%. It can deeply understand the context and word meaning, and can achieve excellent results even with a small number of training rounds, showing the great advantages of the Transformer architecture in sentiment analysis tasks.

Overall, as the complexity of the model increases, the accuracy and robustness of sentiment classification also increase. In particular, BERT shows the latest development direction of current NLP technology.

5. Future Work

1. We can further try more complex and advanced model structures. For example, introducing a bidirectional structure (Bidirectional RNN/LSTM) based on RNN and LSTM can capture contextual information at the same time and enhance the understanding of the overall semantics of the sentence. In addition, combined with the attention mechanism (Attention), it can help the model focus on emotionally strong or key words, improving the interpretability and performance of the model. In the BERT model, ablation experiments such as layer freezing, Dropout adjustment at different positions, and optimized pooling strategy can be performed to better understand the impact of its key components on performance.

2. Currently, we only use part of the IMDB dataset for training and testing. In the future, we can consider using the complete dataset or expanding to data from more fields (such as social media, product reviews) to enhance the generalization ability of the model. On the other hand, combined with data enhancement methods (such as synonym replacement, back translation, spelling perturbation, etc.), the robustness of the model can be improved when the sample is limited. At the same time, error analysis of samples with poor model performance can help discover the challenges posed to the model by language phenomena (such as irony and double negation).

3. This project focuses on sentiment analysis of movie reviews, but the relevant methods have good transferability. In the future, we can try to apply the current model to financial texts, health forums, news reviews and other fields to explore its adaptability under different text styles and vocabulary distributions. In addition, combined with visual analysis (such as sentiment trend charts, hot word clouds) and user portraits, we can develop sentiment insight tools with commercial value to provide intelligent support for scenarios such as public opinion monitoring and product feedback analysis.

6. References

Rathi, M., et al. Sentiment analysis of tweets using machine learning approach. in 2018 Eleventh international conference on contemporary computing (IC3). 2018. IEEE.
Baid, P., A. Gupta, and N.J.I.J.o.C.A. Chaplot, Sentiment analysis of movie reviews using machine learning techniques. 2017. 179(7): p.45-49
https://www.kaggle.com/code/bayazo/transformers-comparisons
https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews