Deep Learning Approaches for Sentiment Analysis Based on IMDB Movie Reviews

Lei Yang
Tulane University
lyang20@tulane.edu

Qinzheng Xu
Tulane University
qxu5@tulane.edu

1 Problem Overview

The issue at hand is using deep learning models to automatically figure out how people feel about movies in reviews. The goal is to correctly find, separate, and label the feelings behind movie reviews from the IMDB database as either positive, negative, or neutral. The subjective and often complicated nature of human language, which includes sarcasm, idiomatic expressions, and different emotional states, makes this job hard. To make things even more difficult, you have to understand the context and semantics of the texts in order to accurately classify the mood. The end goal is to use the information gathered from this research to help entertainment companies improve their services, make their marketing more effective, and make viewers happier. Achieving high accuracy in sentiment analysis can make these insights more open to everyone, so more people can understand and use the feelings and opinions shared in movie reviews.

2 Data

This sentiment study is based on information from the IMDB database, which has 50,000 movie reviews. This dataset was carefully chosen for tasks that involve natural language processing and text analytics. Its main focus is on binary mood classification, which means putting reviews into two groups: positive and negative. The data has a lot of it, which makes it a great resource for training and testing deep learning models that are meant to analyse mood. This dataset is unique because it is so big. Compared to other standard datasets in the same field, it lets you train and test models in more ways.

There are two parts to the dataset: a training set and a testing set. Each has 25,000 movie reviews. This fair split makes sure that both positive and negative feelings are represented equally, which is important for teaching computers to correctly sort feelings without any bias. The data comes from the IMDB library and can be found on Kaggle (Maas et al., 2011), a website for data science and machine learning projects. In the fields of mood analysis and natural language processing, this makes it easier to repeat studies and make sure that results are correct.

3 Related Work

When researchers use sentiment dictionaries to study sentiment analysis, the first thing they need to do is label the expression trend of sentiments. This means that they need to make the emotional meanings of words clearer. The study is based on predefined semantic rules and the sentiment scoring of the lexical expressions in the sentiment dictionary. A series of text processing is used to get the general sentiment tendency of the text. Some researchers, including Tureny et al. (Turney, 2007), look at and figure out how similar words are by adding mutual information intent. They also rebuild the dictionary by adding polar semantics, which makes the dictionary's lexical construction more complex. Yang et al. (Chen and Yang, 2013) start with a sentiment dictionary that already exists and use the LDA model tool to find the topic words in that dictionary. This makes the sentiment tendencies more expressive and gives topic words with multiple meanings a lot of extra information. Pang et al. (Pang et al., 2002) were the first to use machine learning to solve the sentiment analysis binary classification problem. They combined standard bag-of-words methods with machine learning to make the classification work better. Wikarsa et al. (Wikarsa and Thahir, 2015) gathered a huge number of Twitter comments as a dataset and then put emotions into groups based on their meanings. Once the groups were made, the authors used standard machine learning algorithms to figure out how people felt about the tweets, taking into account the emojis and words in the dataset to improve the algorithm's performance and accuracy. Govindarajan et al. (Govindarajan, 2013) made a big step forward in combining genetic algorithms and traditional machine learning algorithms in a natural way. They used the combined algorithm as a new way to solve the problem of sentiment binary classification and got better results in binary classification effects. Chikersal et al. (Chikersal et al., 2015) changed the emotional rules of emojis and polysemous words in comment texts by mixing rule processing and SVM and processing Twitter corpora. This made text semantic prediction more accurate. But the text opinion research based on machine learning has hit a wall. This is because the research field is getting too big. When it comes to current long text jobs that involve semantics, the combination of dictionary rules and machine learning often doesn't work as planned, which slows down research and development.

4 Method

4.1 Algorithms Used

We used three different types of neural networks to figure out how people felt about IMDB movie reviews: Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and Transformer models. Each of these models takes a different approach to dealing with and making sense of the complex structure of textual data, and each has its own benefits when it comes to catching the subtleties of language and emotion.

- **ANN Model**: As a starting point, the ANN model is used to simulate a sped-up version of the neural processes seen in the human brain. It works by learning patterns and connections in the data through layers of neurones. We used a structure with thick layers and dropout layers in between to keep the model from overfitting and make sure it works well with data it hasn't seen before.

- **RNN Model**: Textual data is usually stored in a certain order, so the RNN model takes that into account when it comes to processing inputs. The RNN can remember long-term relationships thanks to its Long Short-Term Memory (LSTM) units. This makes it perfect for sentiment analysis, where the order and context of words have a big effect on their meaning.

- **Transformer Model**: The Transformer architecture introduces an attention mechanism, allowing the model to focus on different parts of the input sequence when making predictions. This approach is highly effective for understanding context and relationships within the text, offering advantages in processing the complex structure of movie reviews.

4.2 Front-end Interaction Tool

A Strong open-source Python library Streamlit is praised for its ability to speed up the development of web applications for data science and machine learning. It was designed to provide data scientists and developers more power by enabling them to transform data-centric scripts into minimally coded interactive web apps. Nowadays, there are many different data visualization tools available, each with their own advantages. These include code libraries like Plotly and Bokeh as well as the well-known Tableau and Power BI. However, in the field of interactive applications, Streamlit has become a strong competitor for both developers and data scientists. The core design principle of Streamlit is simplicity. Just a few lines of Python code can be used to create an interactive web application. A Streamlit program can be created with very little changes from an existing Python script, in contrast to some technologies that need complex settings or dashboard design. Furthermore, Streamlit's instantaneous feedback loop, which allows code modifications to be reflected quickly in the browser, encourages quick prototyping. Compared to programs like Tableau, where iterative design might not be as flexible, this is a benefit.

4.3 Libraries and Variations

We used well-known Python tools for deep learning, like TensorFlow and Keras, to put these models into action. These libraries make the development process faster by giving you a lot of tools and features for creating and training neural network models.

- We used normal configurations from the Keras library for the ANN and RNN models, but changed the number of units, dropout rates, and sequence processing to make them work best for sentiment analysis.

- Because TensorFlow is so flexible, the Transformer model implementation added custom layers and attention mechanisms to make it work for our text classification job.

4.4 Evaluation Methods

A confusion matrix, which is a powerful way to see how well a model can classify things into different groups, was used to judge the models' success. Accuracy, precision, recall, and the F1 score are some of the key measures we got from the confusion matrix. These metrics give a full picture of how well the model can predict the future and tell the difference between happy and negative emotions.

- **Accuracy** measures the overall effectiveness of the model in correctly identifying sentiments.
- **Precision** and **Recall** offer insights into the model's ability to correctly predict positive sentiments and minimize false positives.
- **F1 Score** serves as a balance between precision and recall, providing a single metric to assess the model's overall performance in handling both aspects.

4.5 Baselines for Comparison

A comparison with well-known benchmarks in sentiment analysis was part of our research to make sure that our models worked. By comparing our results to these standards, we wanted to show how our method led to progress and possible improvements. We specifically looked at standard metrics that regular machine learning methods and past neural network models got right in sentiment analysis tasks. We then used these as a starting point to see how much better our ANN, RNN, and Transformer models were.

5 Preliminary Experiment Results

5.1 Label Distribution

We started our exploratory data analysis by looking at the sentiment label distribution in the dataset.

This visualization, which takes the shape of a bar chart, shows how the two sentiment labels—positive and negative—are distributed equally. Each of the two groups has 25,000 occurrences, which is the same amount for both. Because it eliminates the requirement for data balancing strategies and indicates that the dataset is suitable for assessing classification models without running the danger of bias towards a more frequent label, this parity in distribution is helpful for our binary sentiment classification. Given that all classes are equally represented in the dataset, accuracy may also be a trustworthy statistic for assessment.

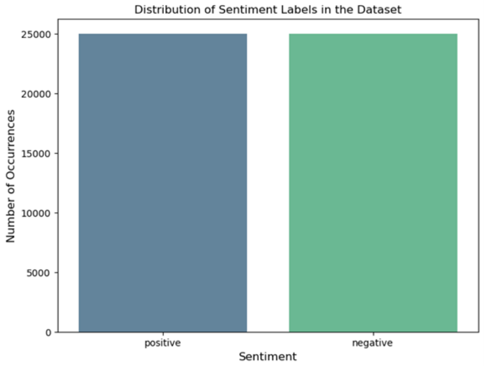


Figure 1: Label Distribution

5.2 Word Cloud

A word cloud analysis was part of our exploratory data analysis. It showed us which words were used most often in both the positive and bad movie reviews in our dataset.

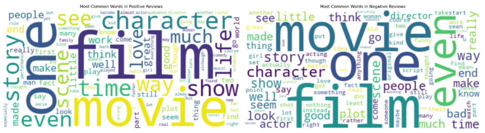


Figure 2: Word Cloud for Positive/Negative Reviews

Word clouds show how often a word appears in the evaluations; the bigger a word is, the more often it appears in the evaluations. Positive scores use words like ”love,” ”great,” ”best,” ”good,” and ”amazing” a lot, which shows that people usually say nice things about things. The words ”story,” ”character,” and ”film” are also used a lot, which shows how important story and character are to good reviews.

The negative rating word cloud, on the other hand, has words like ”bad,” ”worst,” ”nothing,” and ”waste,” which all make it clear that the user is unhappy. Furthermore, it shows a strong presence of words like ”plot,” ”acting,” and ”script,” which suggest criticism aimed at the movie's acting and story.

The word cloud analysis not only helps you find the most common words, but it also shows you which parts of movies are most often linked to both positive and negative feelings. This information shows the basic ideas and phrases that might be able to tell us how people feel about movies, which could be useful for future work with sentiment analysis and prediction models.

5.3 Text Features for Extraction

We will extract the following text features, each offering unique insights into the text data:

1. **Polarity**: This is a measure ranging from -1.0 to 1.0. A score of 0 denotes neutral sentiment, -1 signifies extremely negative sentiment, and +1 reflects highly positive sentiment.

2. **Subjectivity**: Expressed as a float within the range of 0.0 to 1.0. A score of 0.0 indicates a highly objective stance, whereas 1.0 suggests a highly subjective viewpoint.

3. **Word Count**: The total number of words present in each review.

4. **Part-Of-Speech (POS) Tag Ratios**: We will calculate the ratio of specific POS tags to the total word count in a review. The tags include:

   - PROPN (Proper Noun), e.g., John, London, NATO.
   - PUNCT (Punctuation), e.g., ., (,), ?, !.
   - NOUN, e.g., woman, tree, beauty.
   - ADJ (Adjective), e.g., big, yellow, sparkling.
   - VERB, e.g., eat, running.

5. **Uppercase Ratio**: The proportion of uppercase words to the total word count in the review.

6. **Digits Ratio**: The proportion of digits to the total word count in the review.

5.4 Polarity Distribution

The histogram shows the orientation scores of reviews that are both positive and negative. When reviews are good, the polarity scores are more concentrated on the right side of the scale, reaching their highest point in the range that shows strong positive sentiment. On the other hand, bad reviews have a polarity distribution that leans to the left and has a peak in the negative spectrum.

The histogram presented captures the polarity scores of both positive and negative reviews. For positive reviews, the polarity scores cluster more densely towards the right side of the scale, peaking in the range that indicates strong positive sentiment. This suggests that the language used in positive reviews is consistently associated with higher positivity in sentiment analysis.

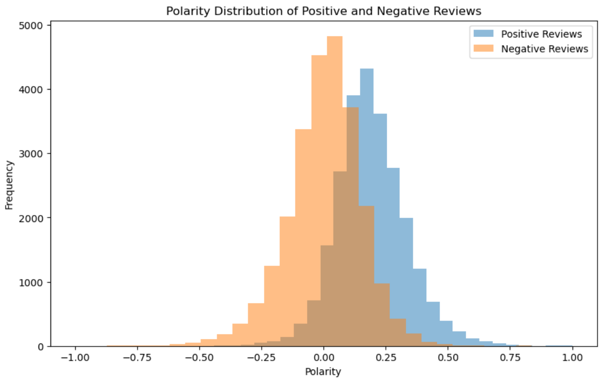


Figure 3: Polarity Distribution

Conversely, the polarity distribution of negative reviews skews towards the left, with a peak in the negative spectrum, underscoring a trend of more negative sentiment expressions. The presence of a few negative reviews with a slightly positive polarity score may indicate either a nuanced view within the reviews or potential noise in the data. This bimodal distribution highlights the clear demarcation between the sentiments of positive and negative reviews. Understanding the distribution of polarity scores is essential for refining sentiment analysis algorithms, as it demonstrates the typical sentiment range for positive and negative feedback. Such insights can improve the accuracy of sentiment prediction models by emphasizing the polarity features that most strongly correlate with each sentiment category.

5.5 Word Count Distribution Analysis

Continuing with our exploratory data analysis, we next turn our attention to the word count distribution within the movie reviews. This analysis is crucial as it provides insights into the verbosity and the level of detail present in the reviews.

6 Model Training Result

6.1 ANN Model

The performance of the Artificial Neural Network (ANN) model is very good. A confusion matrix shows that there are a lot of true positives and true negatives and almost no false positives and false negatives in the training set. This means that the

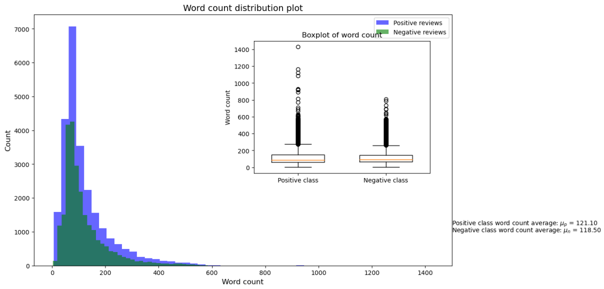
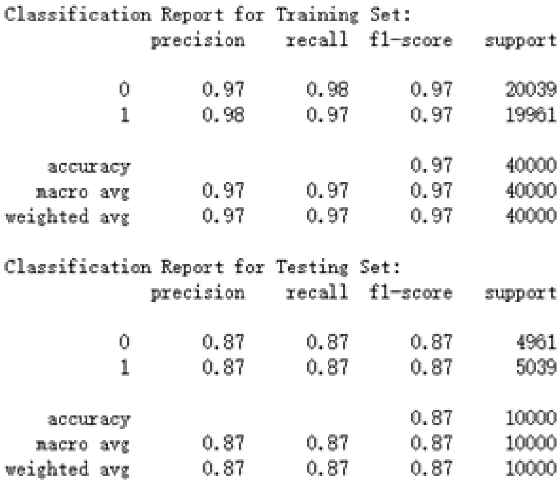


Figure 4: Polarity Distribution

model is very good at classifying things. The area under the curve (AUC) of the Receiver Operating Characteristic (ROC) curve is 0.99, which means that the models can be separated very well.

The model keeps its good classification accuracy for the testing set, but there are a few more wrong classifications than for the training set. The confusion matrix shows a fair classification, with about the same number of true positives and false negatives. The ROC curve's AUC of 0.94 also shows strong performance. The high AUC is in line with the fact that the accuracy, recall, and F1 scores are almost perfect for the training set. These measures always stay at 0.87 for both classes on the testing set. This is lower than the training set, but it still shows that the model did well on data it had not seen before.



```
Classification Report for Training Set:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97     20039
           1       0.98      0.97      0.97     19961

    accuracy                           0.97     40000
   macro avg       0.97      0.97      0.97     40000
weighted avg       0.97      0.97      0.97     40000

Classification Report for Testing Set:
              precision    recall  f1-score   support

           0       0.87      0.87      0.87      4961
           1       0.87      0.87      0.87      5039

    accuracy                           0.87     10000
   macro avg       0.87      0.87      0.87     10000
weighted avg       0.87      0.87      0.87     10000
```

Figure 5: Evaluation Statistics for ANN Model

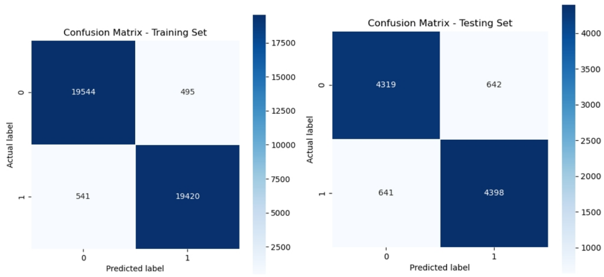


Figure 6: Confusion Matrices for ANN Model

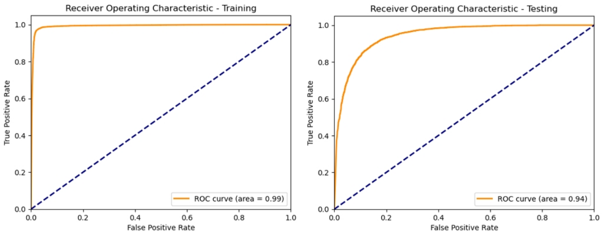


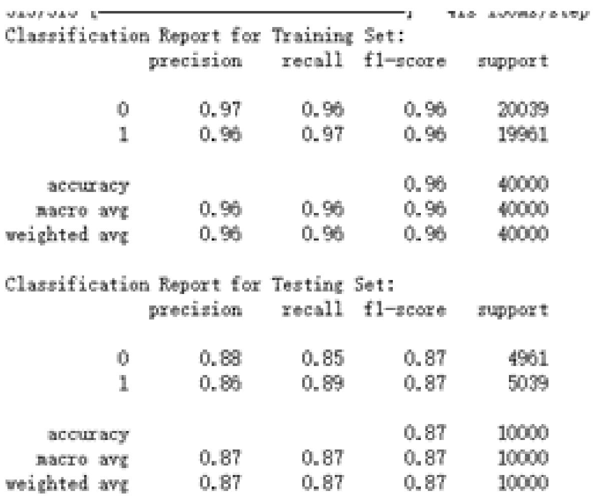
Figure 7: ROC Curve for ANN Model

6.2 RNN Model

The performance metrics and visualizations for an LSTM (Long Short-Term Memory) model show that it has been trained to correctly identify two groups (label 0 and label 1) with a high level of accuracy on the training set and a fair level of accuracy on the testing set. We can see from the confusion matrix that the model got 4218 out of 4483 of the predictions it made on the testing set right. There were, however, 556 false negatives (where it thought label 0 was true label 1) and 743 false positives (where it thought label 1 was true label 0). With an area under the curve (AUC) of 0.87, the Receiver Operating Characteristic (ROC) graph shows that the model can clearly tell the difference between the two groups. An AUC number close to 1 means that the model works very well, so 0.87 is a pretty good value for many uses. The classification reports tell us a lot about how well the model worked. The model does a great job on the training set, showing high accuracy, recall, and F1-score for both classes, which stays around 96 percentages. These numbers are a little lower for the testing set, averaging around 87 percentages. This is to be expected, since models tend to do worse on new data that they haven't seen before.

6.3 Transformer Model

The confusion matrix and classification reports show that the Transformer model did well on both the training set and the testing set. Precision and memory did change a bit between the two groups, though. To check if the model was right, the confusion matrix shows that it correctly predicted 4470 instances of class 0 and 4152 instances of class 1. But 491 of the guesses turned out to be wrong. They said that class 0 was class 1 and class 1 was



```
Classification Report for Training Set:
              precision    recall  f1-score   support

           0       0.97      0.96      0.96     20039
           1       0.96      0.97      0.96     19961

    accuracy                           0.96     40000
   macro avg       0.96      0.96      0.96     40000
weighted avg       0.96      0.96      0.96     40000

Classification Report for Testing Set:
              precision    recall  f1-score   support

           0       0.88      0.85      0.87      4961
           1       0.86      0.89      0.87      5039

    accuracy                           0.87     10000
   macro avg       0.87      0.87      0.87     10000
weighted avg       0.87      0.87      0.87     10000
```

Figure 8: Evaluation Statistics for RNN Model

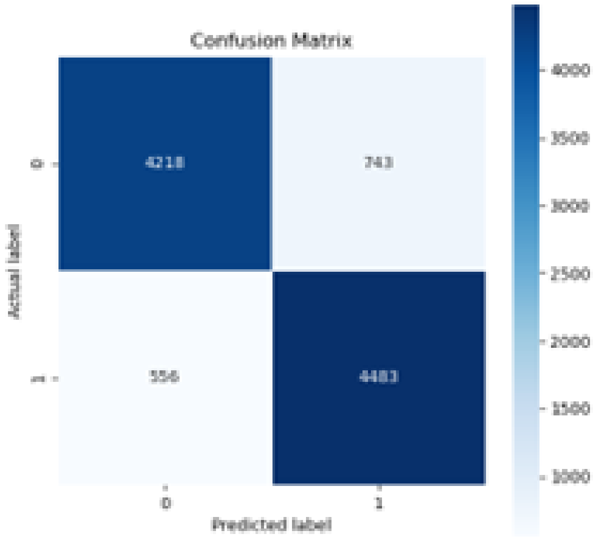


Figure 9: Confusion Matrices for RNN Model

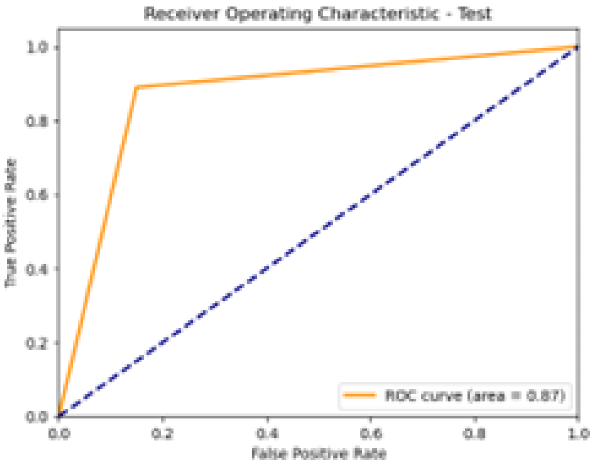
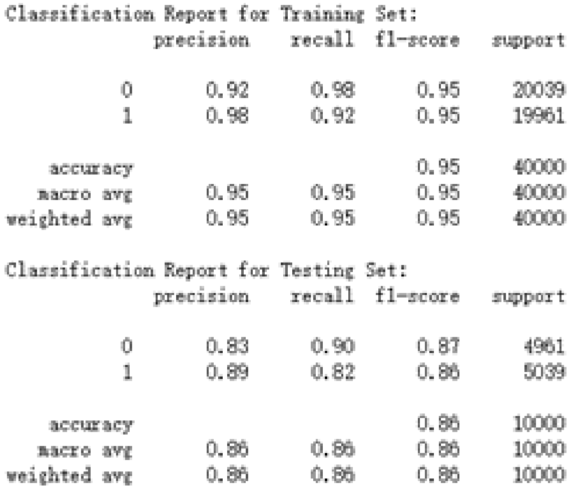


Figure 10: ROC Curve for RNN Model



```
Classification Report for Training Set:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     20039
           1       0.98      0.92      0.95     19961

    accuracy                           0.95     40000
   macro avg       0.95      0.95      0.95     40000
weighted avg       0.95      0.95      0.95     40000

Classification Report for Testing Set:
              precision    recall  f1-score   support

           0       0.83      0.90      0.87      4961
           1       0.89      0.82      0.86      5039

    accuracy                           0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000
```

Figure 11: Evaluation Statistics for Transformer Model

class 0. To be accurate and remember things, the model got a score of 0.95 for both classes in the training set. Class 1 is a little more accurate than Class 0, but Class 0 is better at remembering things. This proves that the model can reliably change based on the learned facts. This set of tests shows that class 0 is less accurate than class 1, but class 0 has better memory than class 1. The F1 score for class 0 is 0.87, and the F1 score for class 1 is 0.86. In other words, the model is a little better at finding things that belong to class 0. With a score of 0.86 on the test set, the Transformer model does pretty well all around. When the tests were given, there was a small difference in the accuracy and recall of the two groups. This suggests that there may be more room for improvement. One way to do this would be to change the decision level or the weights for each class.

7 Web App Dashboard

7.1 Manual Mode

The person using this interface can pick a machine learning model from a dropdown menu. In this case, they picked an Artificial Neural Network (ANN) to study. This time, the method of input was set to ”Manual Input” from another drop-down menu.

A text box called ”Enter Review Text” is below the dropdowns. This is where the user can write a review of a movie. The example text ”I Love this movie” has been written, which means the person likes the movie.

The user can click the ”Detect Sentiment” button after writing the review to look at how the review

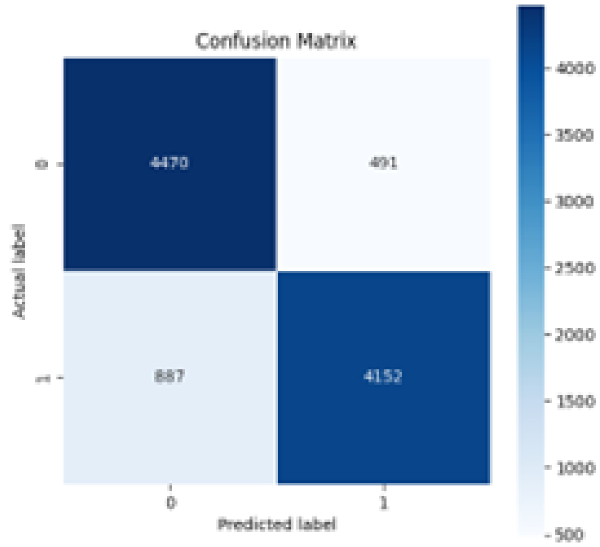
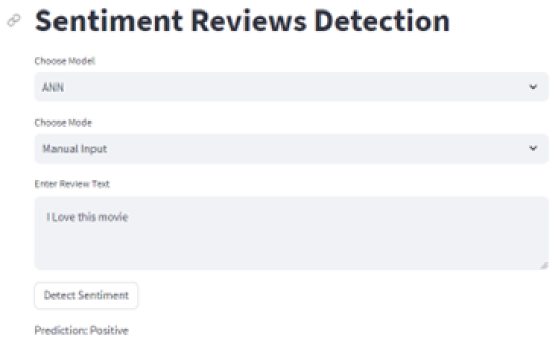


Figure 12: Confusion Matrices for Transformer Model


Sentiment Reviews Detection

Choose Model: ANN

Choose Mode: Manual Input

Enter Review Text: I Love this movie

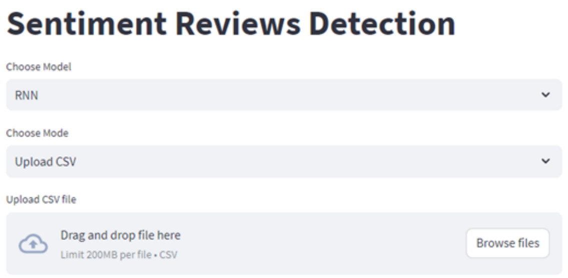
Prediction: Positive

Figure 13: Web Interface with Manual Input

makes them feel. The app uses the chosen ANN model to process the text you enter and shows the outcome below the button. According to the model, the review's sentiment was positive, and the forecast ”Positive” backs this up. This matches the positive sentiment in the review text. This shows that the ANN model correctly figured out how the reviewer felt about the data in this case.

7.2 CSV Mode

In the CSV file mode. A dropdown menu let the user choose the Recurrent Neural Network (RNN) model. They then posted a CSV file called ”test.csv” for sentiment analysis. Once the program has finished processing the file, it shows the sentiment predictions for each review in the file. It shows the outcomes of two reviews in this case, both of which are marked as ”Negative.” It's likely that these results come from how the RNN model understands the words used in the reviews.


Sentiment Reviews Detection

Choose Model: RNN

Choose Mode: Upload CSV

Upload CSV file

Figure 14: Web Interface with Manual Input

There is a bar chart below the predictions called ”Sentiment Count” that shows how many negative reviews were posted. The bar chart shows that there are two reviews, and both have been marked as negative because there is only one bar for the negative count and it is set at a height of 2, which matches the two negative forecasts shown above.

This visual aid makes it easy for users to quickly grasp how the reviews in the CSV file's feelings are spread out. On the other hand, it looks like the X-axis labels on the bar chart are missing or can't be seen. These labels should show the different types of emotion, like ”Positive” and ”Negative.”

8 Discussion and Conclusion

8.1 Discussion

The results from the Artificial Neural Network (ANN), Recurrent Neural Network (RNN), and Transformer models give a full picture of what mood analysis can do now. Each model did a good job of processing and sorting mood data. The ANN had almost perfect training results, and both the RNN and the Transformer were strong at dealing with complex patterns in sequence data. The ANN model learned a lot from the training set because it almost always got the right answer and had a high ROC AUC score. But the small drop in speed on the test set makes me wonder if the model might be too good. With its LSTM design, the RNN model also did very well, especially when dealing with sequential text data, which can be hard because of the subtleties of language. With its attention mechanisms, the Transformer model seems to be very good at capturing contextual connections in data. It kept up its high accuracy and F1-scores, which shows that it could use the training data to make new predictions.

8.2 Limitations

Even though the results look good, there are some things that need to be thought about. The difference in accuracy between training and testing shows that the models may be fitting the training data too well. For the Transformer model, the difference in accuracy and recall between the classes also points to possible biases in the model or the data. It's also possible to get the results wrong because the application's user interface, especially in CSV mode, doesn't have clear labels in the visualizations. Also, the models haven't been tried with a wider range of data or more complex emotions, like sarcasm or irony, which are hard for sentiment analysis models to understand.

8.3 Conclusion

Overall, the models that were made are a good base for using mood analysis in other situations. The web app dashboard makes these models easier to use and more accessible by letting you do real-time mood analysis with both manual and CSV inputs.

In future work, fixing the overfitting problem by using regularization techniques or more advanced data addition methods could make the model more stable. It's also important to make the choices made by the models easier to understand, especially for uses that need to know a lot about the model's thinking.The app will be more useful if the user experience is improved so that data can be seen better and more preprocessing is added so that it can handle a wider range of emotions. Transfer learning or multilingual models could also be looked into to make the app more useful in more languages and areas.

As natural language processing and machine learning models keep getting better, it's a great chance to make mood analysis tools even better. In the future, the app can give even more accurate and useful sentiment analysis by using bigger datasets, more complex models, and learning from user feedback.

References

L Chen and Y Yang. 2013. Emotional speaker recognition based on i-vector through atom aligned sparse representation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7760–7764. IEEE.

P. Chikersal, S. Poria, and E. Cambria. 2015. Sentu: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 647–651.

M. Govindarajan. 2013. Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm. *International Journal of Advanced Computer Research*, 3(4):139.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*.

B Pang, L Lee, and S Vaithyanathan. 2002. Thumbs up: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics.

B. W. Turney. 2007. Anatomy in a modern medical curriculum. *The Annals of The Royal College of Surgeons of England*, 89(2):104–107.

L Wikarsa and S. N. Thahir. 2015. A text mining application of emotion classifications of twitter's users using naive bayes method. In *2015 1st International Conference on Wireless and Telematics (ICWT)*, pages 1–6. IEEE.