UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Artificial Intelligence and Machine Learning (6CS012)

Sentiment Analysis of Hotel Reviews with Recurrent Neural Network and It's Variant.

Student ID: 2329861

Student Name: Anurag Sharma

Group: L6CG6

Tutor: Mr. Shiv Kumar Yadav

Module Leader: Mr. Siman Giri

Cohort: 9

Submitted on: 13/05/2025

Abstract

This project focuses on the sentiment classification of hotel reviews with the goal of using deep learning techniques like Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), and LSTM combined with pretrained Word2Vec embeddings to predict user ratings between 1 and 5. In order to provide business insights, the main objective is to analyze customer feelings from textual reviews.

In order to reduce the skew, methods such as oversampling underrepresented categories were manually implemented due to the dataset's notable class imbalance. Bi-directional versions of the models were used to improve learning from both preceding and following words in a phrase. Tokenization, stop word removal, and sequence padding were among the preparatory procedures in the data preparation pipeline.

The LSTM model using Word2Vec embeddings performed better in terms of accuracy than the other two models, according to the experimental data after all three models were trained. The model's practical application was demonstrated by the development of a graphical user interface (GUI) that allows for real-time sentiment prediction from user-input hotel reviews. According to the results, LSTM performs better than conventional RNN models. For better training, future developments might include adding a bigger and more varied dataset.

Table of Contents

Table of Figures

1. Introduction

Sentiment analysis, such as analyzing hotel reviews in Natural Language Processing (NLP), is essential for companies looking to comprehend consumer feedback for advertising. This project aims to classify hotel ratings on a scale of one to five using deep learning techniques such as simpleRNN, LSTM, and LSTM with Word2Vec embeddings.

A study by Abid Ishaq (Ishaq, 2020), titled "Extensive Hotel Reviews Classification using Long Short-Term Memory", explored the use of LSTM with word embeddings for classifying hotel reviews based on guest sentiment. The dataset, obtained from Kaggle, contained 515,000 reviews from 1,493 luxury hotels across Europe. Preprocessing techniques such as lowercasing, punctuation removal, tokenization, and stopword removal were applied to clean the data. The research also compared various machine learning algorithms, like Logistic Regression and Decision Tree, with deep learning models such as Bi-directional LSTM (BiLSTM) and Gated Recurrent Units (GRU). Reviews in the dataset were categorized into three different formats to evaluate model performance effectively.

The reviews were categorized into three ways:

a. 10 – class (ratings from 1 to 10)
b. 3 – class (Negative: 1 – 4, Neutral: 5-7, Positive: 8-10)
c. 2 – class (Negative: 1-5, Positive: 6-10)

The results showed that the LSTM model performed best under the binary classification setting, and the training was conducted on a PC with a 2GB GPU. It obtained an F1-score of 81.69%, 97% accuracy, 87% precision, and 77% recall. This outperformed the other models that were examined, such as Logistic Regression and Bi-directional LSTM (BiLSTM).

Link to the notebook: https://github.com/AnuragSharma8/6CS012/upload/main

2. Dataset

The Dataset used in this project is publicly available on Kaggle (Source: https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews) and was created by Larxel. It contains 20,491 entries with two columns "Review" and "Rating". The maximum

length of the text in the review is 13501 and the rating distribution is depicted below:

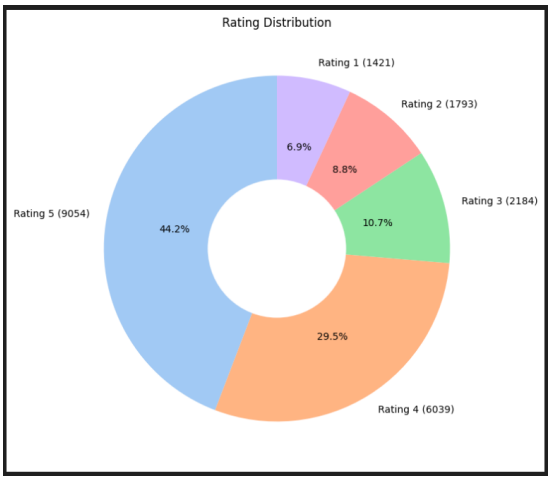


Figure 1: Distribution of Rating

The dataset was divided for model evaluation, with 20% set aside for testing and the remaining 80% for training. Techniques like oversampling and the use of calculated class weights were used to make sure that underrepresented classes received the proper attention during training because of the imbalance in the class distributions.

3. Methodology

The model is trained using two different approaches due to data imbalance.

a. Classifying 1-5 classes using SoftMax activation
b. Classifying 0 and 1 by grouping 1 and 2 to negative and 4 and 5 to positive classes.

3.1 Data Preprocessing

The preprocessing pipeline included the following steps:

a. Converting all text to lowercase
b. Removing URLs, mentions, hashtags, numbers, and special characters
c. Expanding contractions
d. Removing stopwords while keeping sentiment-relevant stopwords
e. Lemmatization
f. Tokenization
g. Sequence padding

3.2 Model Architecture:

3.2.1 SimpleRNN

The SimpleRNN model served as the foundational architecture, starting with an embedding layer that transformed input text into vector form so the model could understand word semantic associations. A SimpleRNN layer with 32 units to record sequential context comes next. Strategies like L2 regularization and dropout were used to enhance model performance and avoid overfitting. Using a SoftMax activation function, the last dense layer generates predictions for five different rating categories. A similar structure is used in the binary classification scenario, which uses a sigmoid activation function and binary cross-entropy loss for classification. However, the SimpleRNN layer is unidirectional and has 64 units.

3.2.2 LSTM

The Long Short-Term Memory (LSTM) network is the second model being used. It starts with an embedding layer that resembles the SimpleRNN model. LSTM, in contrast to SimpleRNN, is more successful at capturing long-term dependencies because it has internal memory and gating mechanisms that allow it to save pertinent information and eliminate noise. For deeper context awareness, the model can analyze input sequences both forward and backward thanks to the usage of a 32-unit bidirectional LSTM layer. A dense layer with a SoftMax activation function is then applied to the output in order to classify it into five rating categories. The binary classification setup follows the same structure as SimpleRNN, with the two-class output changes required.

3.2.3 LSTM with Word2Vec

In the third model, the LSTM architecture is expanded by adding a 64-unit LSTM layer, although other settings remain the same as in the earlier models. By initializing the embedding layer with a pretrained Google News Word2Vec model that uses 300-dimensional word vectors, the model can take use of extensive semantic knowledge and perhaps enhance performance. Learning rates were set at 0.0003 for the first two models and 0.00005 for the third, and the Adam optimizer and categorical cross-entropy as the loss function were used to train all three models. Callbacks such as EarlyStopping and ReduceLROnPlateau were used to optimize training, and a batch size of 32 was used for 20 epochs of training.The LSTM with Word2Vec, like the previous models, uses the same architecture for the binary classification setup.

4. Experiments and Results

4.1 RNN vs. LSTM Performance

The RNN and LSTM models' performance comparison reveals significant variations in training behavior, accuracy, and generalization. Limited generalization is indicated by the SimpleRNN model's training accuracy, which gradually rises above 80%, while validation accuracy stays relatively constant at or below 60%. Similarly, the validation loss stays constant at 1.2%, indicating overfitting, even while the training loss continuously decreases. By comparison, the validation accuracy and loss of the LSTM model are marginally better than those of the RNN, but not significantly. The extreme class imbalance, especially the absence of adequate data for minority classes, which neither regularization nor oversampling strategies could adequately address, seems to be the main source of overfitting in both models.

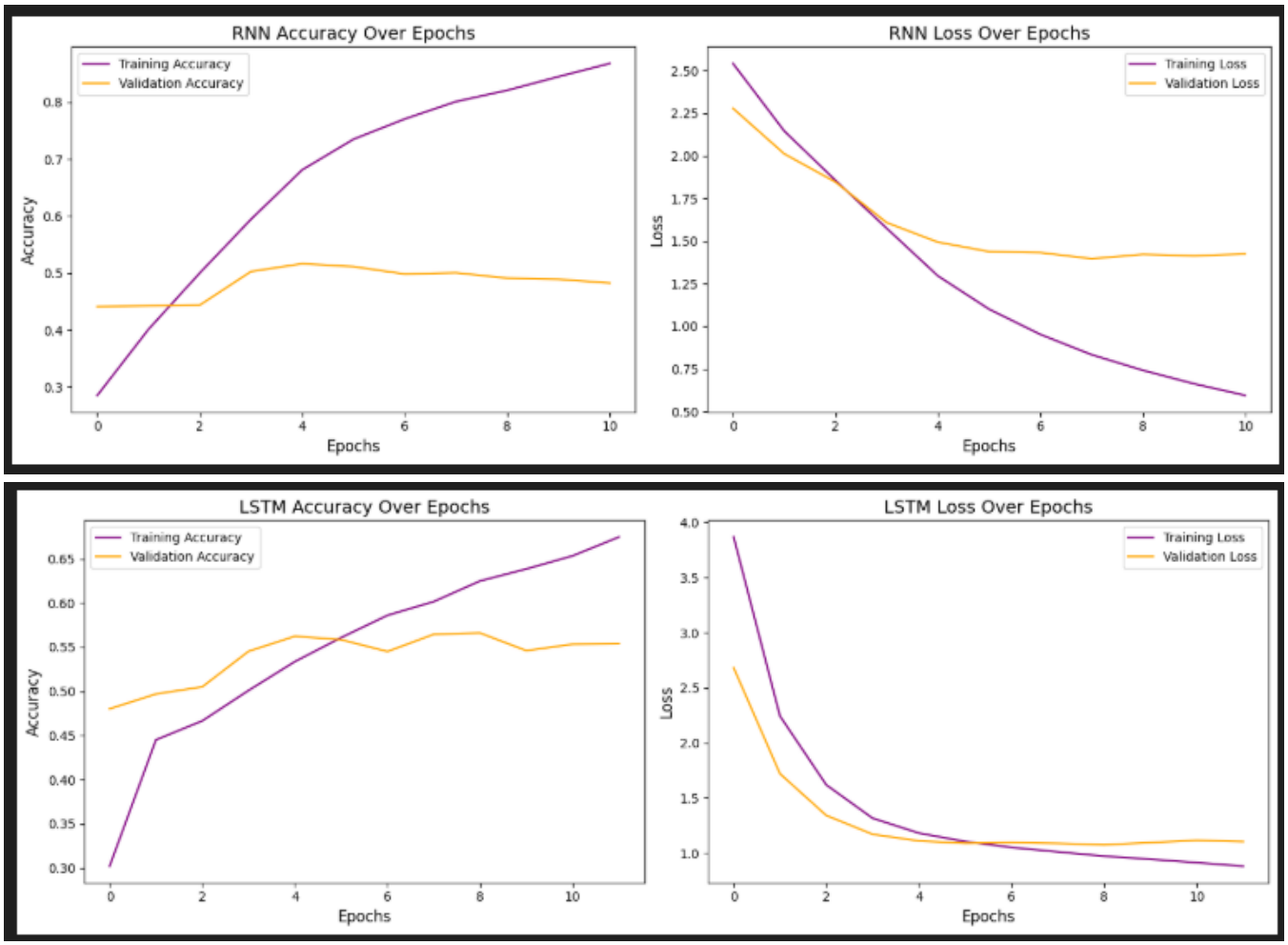


Figure 2: Five classes training and validation curve

The LSTM model performs noticeably better than the RNN model in the binary classification situation. Unlike the more unpredictable behavior shown in the RNN model, the training and validation curves for the LSTM model show steady and consistent learning. RNN manages a little over 70% validation accuracy, whereas LSTM attains almost 90%. Additionally, during training, the LSTM model logs smaller loss values. The training time of the LSTM model is comparable to that of the more straightforward RNN model because of its faster convergence despite its more intricate design.

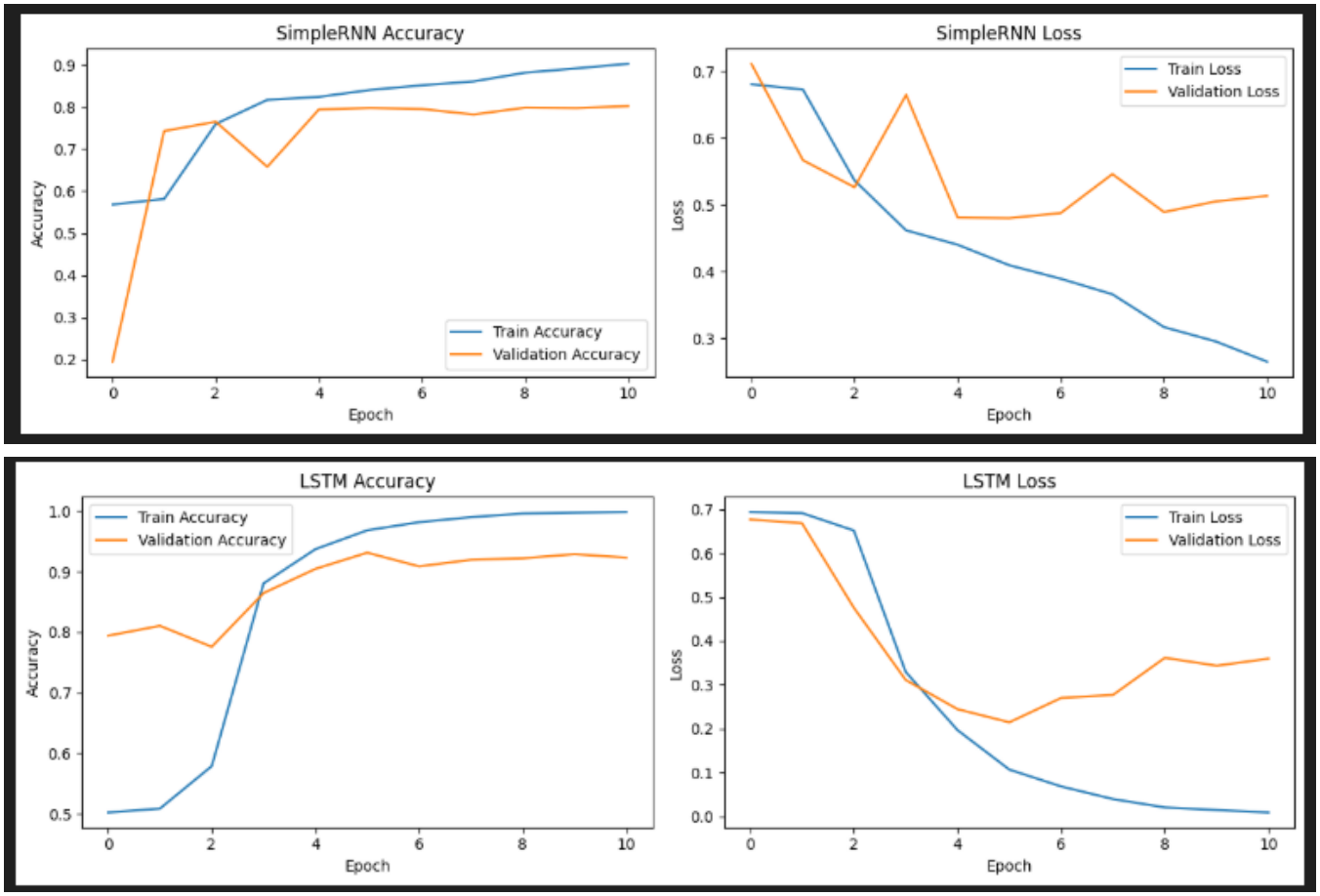


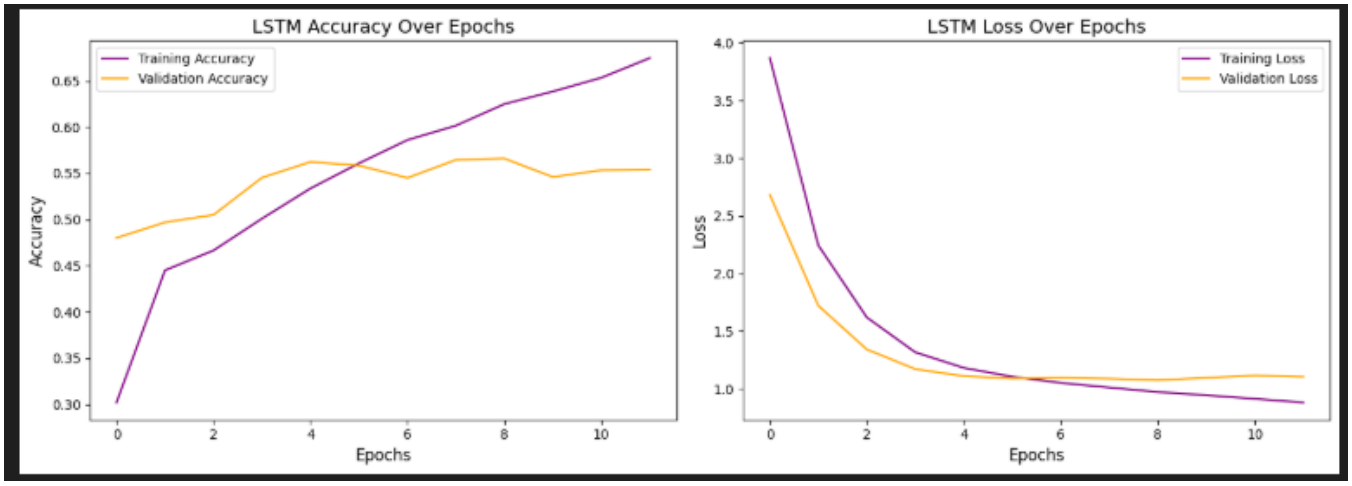
Figure 3: Two classes, Training and Loss Curve

4.2 Computational Efficiency

The relatively small size of the dataset meant that the text categorization task did not require a lot of computer resources. The T4 GPU from Google Colab was used for model training, offering hardware acceleration for faster processing. The RNN and LSTM models were trained in about two minutes under this setup, but it took around five minutes to train the more intricate LSTM model that used Word2Vec embeddings.

4.3 Training with different Embedding's

The LSTM and LSTM with Word2Vec models perform somewhat differently in the five-class classification task. With improved loss optimization, the Word2Vec-enhanced LSTM achieved a somewhat higher accuracy of 61% as opposed to the regular LSTM's 59%. But during the course of the learning process, the regular LSTM displayed more consistent training behavior.



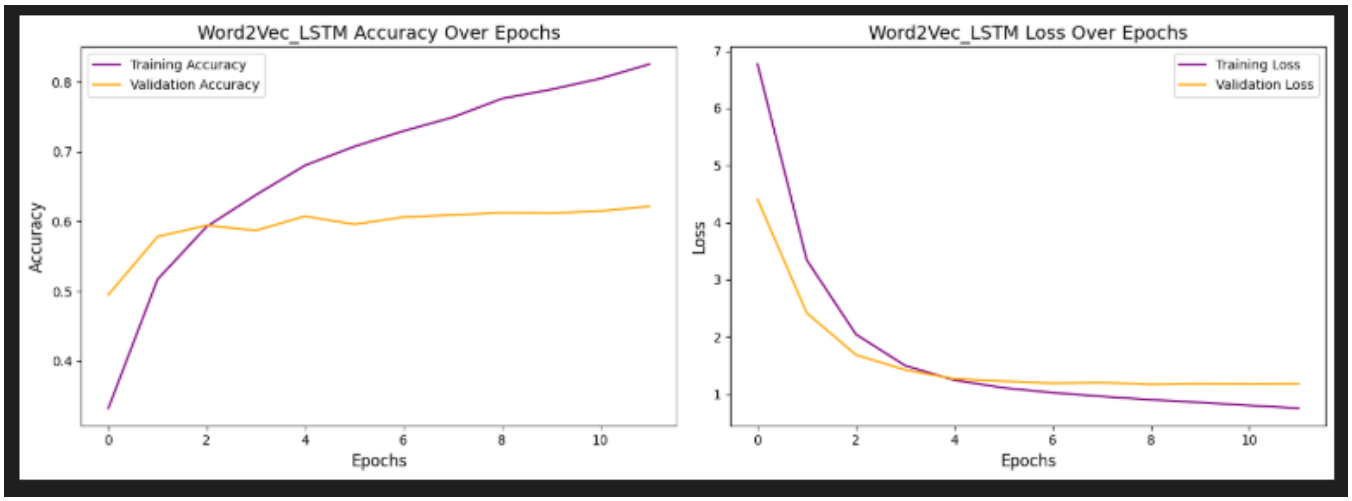
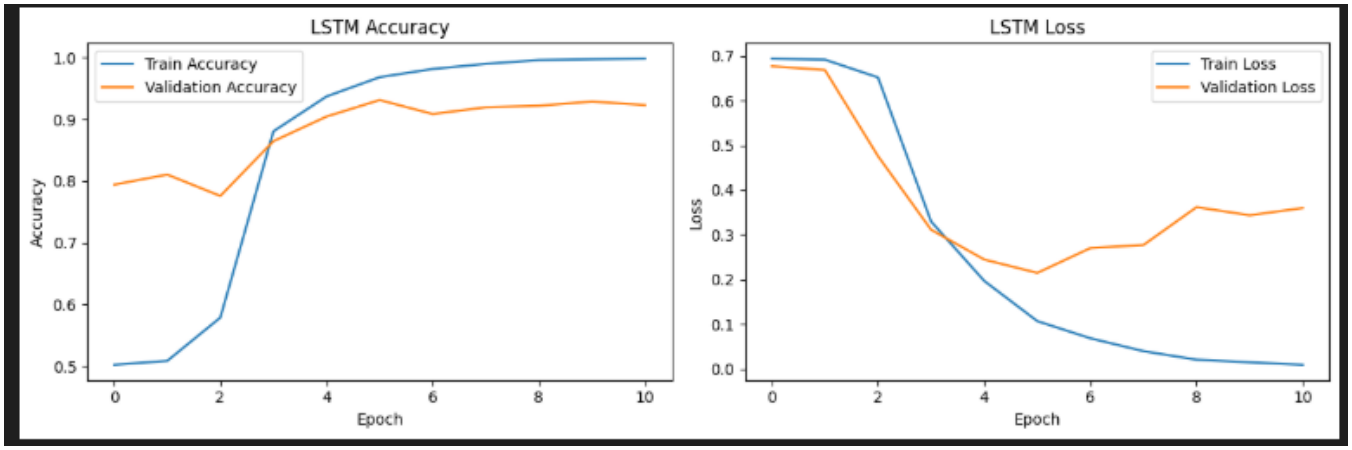


Figure 4: Five Classes LSTM vs. LSTM Word2Vec

At first look, the performance of LSTM and LSTM with Word2Vec in the two classes' classification appears to be comparable, however, it is evident that the Word2Vec model is more stable. While the accuracy of both models is 93 percent, Word2vec outperforms LSTM in the categorization report.



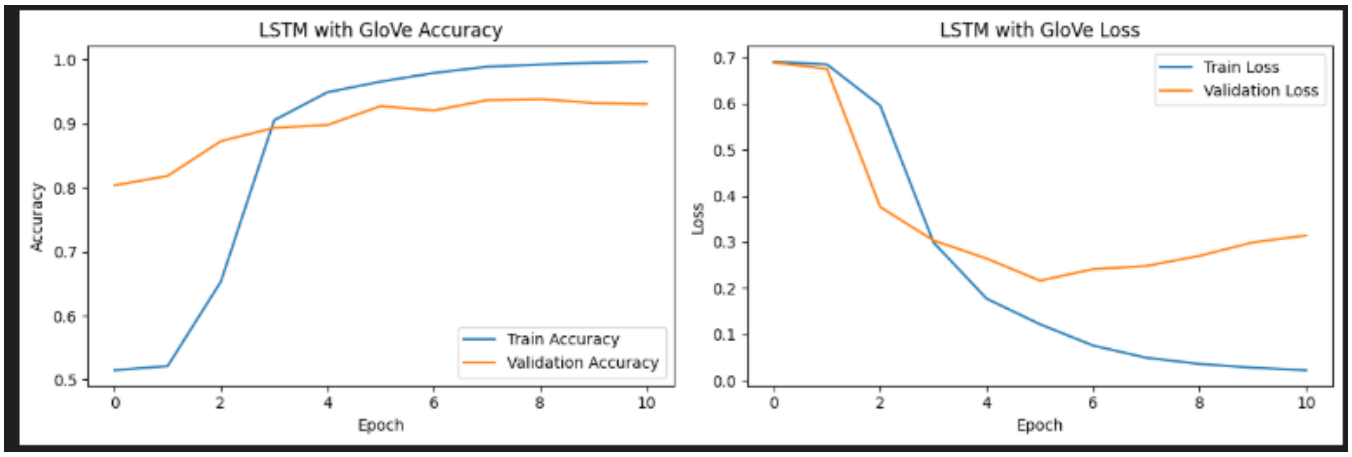


Figure 5: Two Classes LSTM vs. LSTM Word2Vec

Overall, the improvement observed is that word2vec training is stable and optimizes loss better than LSTM.

4.4 Model Evaluation

Among all six models (3 from five classes classification and 3 from two classes classification), the word2vec model outperforms all models in terms of accuracy,

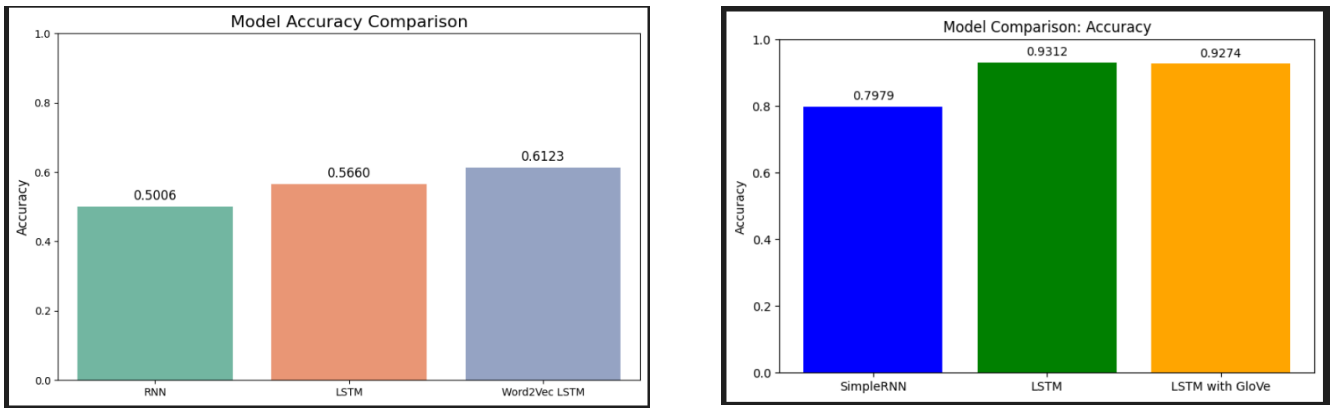


Figure 6: All six models' accuracy

The model's performance varies significantly across the five classes, which are based on 4,099 samples. The SimpleRNN model demonstrated modest generalization across classes, achieving an accuracy of 0.5006 with a macro average F1-score of 0.39. With a macro average F1-score of 0.47 and an accuracy of 0.5660, the LSTM model outperformed this, demonstrating more balanced predictions and superior contextual awareness. With a global average F1-score of 0.56 and an accuracy of 0.6123, the LSTM with Word2Vec embeddings outperformed the other two models to demonstrate the advantages of using pretrained semantic information for multi-class sentiment classification.

RNN Evaluation:
Accuracy: 0.5006099048548427

Classification Report:

	precision	recall	f1-score	support
1	0.40	0.52	0.45	284
2	0.33	0.26	0.29	359
3	0.27	0.19	0.22	437
4	0.42	0.23	0.30	1208
5	0.58	0.80	0.68	1811
accuracy			0.50	4099
macro avg	0.40	0.40	0.39	4099
weighted avg	0.47	0.50	0.47	4099

LSTM Evaluation:
Accuracy: 0.5659917052939741

Classification Report:

	precision	recall	f1-score	support
1	0.54	0.71	0.62	284
2	0.36	0.21	0.26	359
3	0.31	0.40	0.35	437
4	0.51	0.34	0.41	1208
5	0.68	0.80	0.74	1811
accuracy			0.57	4099
macro avg	0.48	0.49	0.47	4099
weighted avg	0.55	0.57	0.55	4099

Word2Vec_LSTM Evaluation:
Accuracy: 0.6123444742620151

Classification Report:

	precision	recall	f1-score	support
1	0.70	0.65	0.68	284
2	0.46	0.50	0.48	359
3	0.33	0.47	0.39	437
4	0.54	0.42	0.47	1208
5	0.76	0.79	0.77	1811
accuracy			0.61	4099
macro avg	0.56	0.57	0.56	4099
weighted avg	0.62	0.61	0.61	4099

Figure 7: Classification Report of Five Classes

The classification report across two classes reveals notable differences compared to the five-class models. The SimpleRNN achieved an accuracy of 0.80 with a macro average F1-score of 0.72, while both the LSTM and LSTM with GloVe models performed similarly, achieving an accuracy of 0.93 and a macro average F1-score of 0.88. The LSTM with Word2Vec also demonstrated high performance with an accuracy of 0.93 and a macro average F1-score of 0.88, highlighting the effectiveness of embedding-based models in improving performance for binary classification tasks.

.

SimpleRNN Evaluation:
Accuracy: 0.7979246313489896

Classification Report:

	precision	recall	f1-score	support
0	0.45	0.76	0.57	643
1	0.94	0.81	0.87	3019
accuracy			0.80	3662
macro avg	0.70	0.78	0.72	3662
weighted avg	0.86	0.80	0.82	3662

LSTM Evaluation:
Accuracy: 0.931185144729656

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.76	0.79	643
1	0.95	0.97	0.96	3019
accuracy			0.93	3662
macro avg	0.89	0.86	0.88	3662
weighted avg	0.93	0.93	0.93	3662

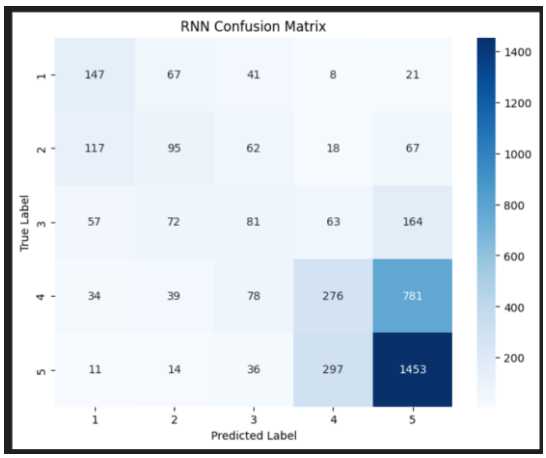
LSTM with GloVe Evaluation:
Accuracy: 0.9273620972146368

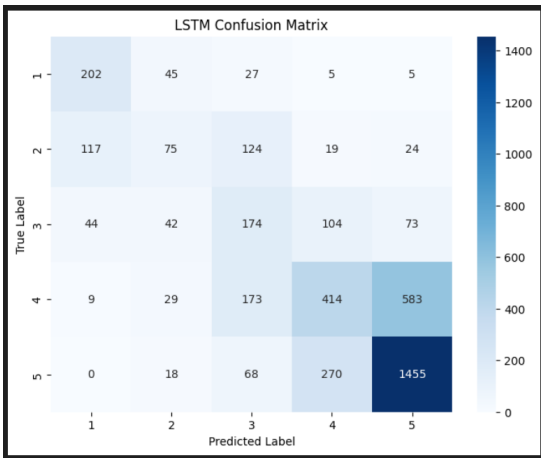
Classification Report:

	precision	recall	f1-score	support
0	0.78	0.82	0.80	643
1	0.96	0.95	0.96	3019
accuracy			0.93	3662
macro avg	0.87	0.89	0.88	3662
weighted avg	0.93	0.93	0.93	3662

Figure 8: Two-Class Classification Report

The confusion matrices for the 5-class classification challenge show significant variations in the performance of SimpleRNN, LSTM, and LSTM with Word2Vec. While SimpleRNN does well in class 5 (1453 accurate predictions) and class 4 (781 correct), it struggles in class 1 and class 3, frequently misclassifying several cases. Accuracy in classes 5, 4, and 3 is improved by LSTM, however it still sometimes mistakes class 3 with class 4 and class 4 with class 5. Overall, LSTM with Word2Vec produces the greatest results, performing especially well in classes 4 and 5, while also lowering errors in classes 3 and 1. In general, SimpleRNN performs the least effectively, while LSTM with Word2Vec accomplishes the most obvious class separation.





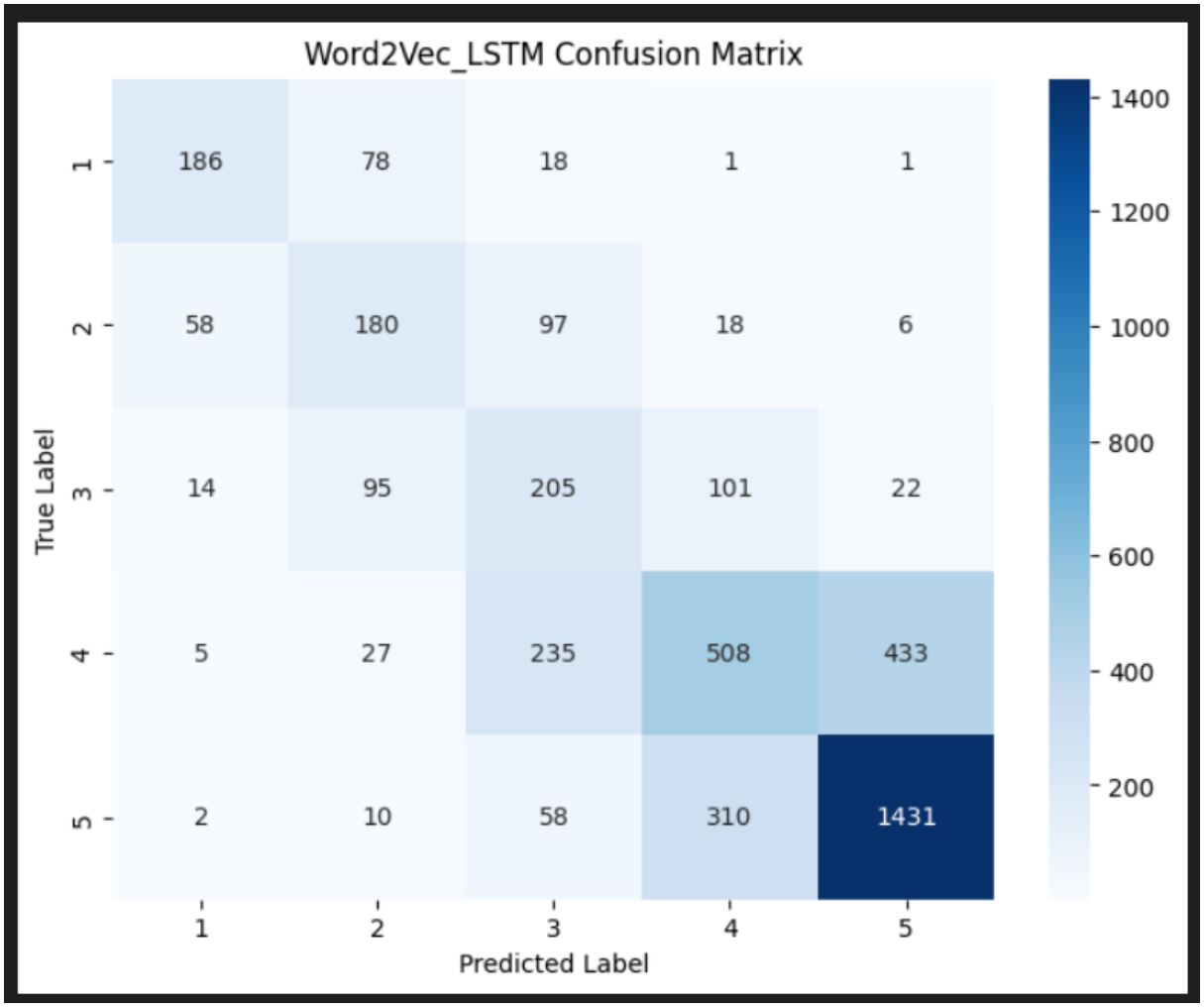
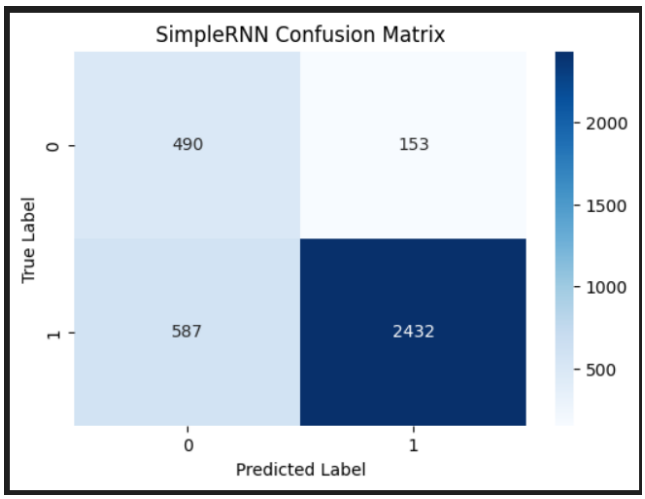
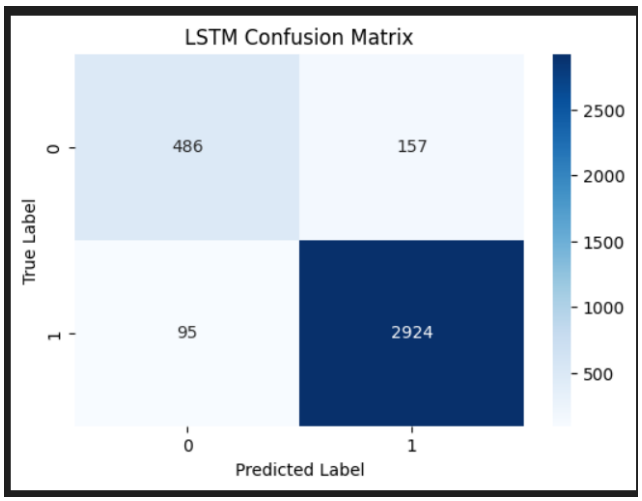


Figure 9: Five Classes Confusion Matrix

The confusion matrices for a binary classification problem show clear performance differences between SimpleRNN, LSTM, and LSTM with GloVe. Despite its significant faults, which include misclassifying 153 instances of class 0 and 587 occurrences of class 1, SimpleRNN correctly predicts 490 instances of class 0 and 2432 instances of class 1. With fewer errors, LSTM outperforms this, accurately detecting 486 class 0 and 2924 class 1 occurrences. LSTM with GloVe exhibits the fewest misclassifications and the best performance, correctly predicting 530 class 0 and 2866 class 1 cases. In conclusion, LSTM combined with GloVe provides the finest balance and accuracy in both classes.





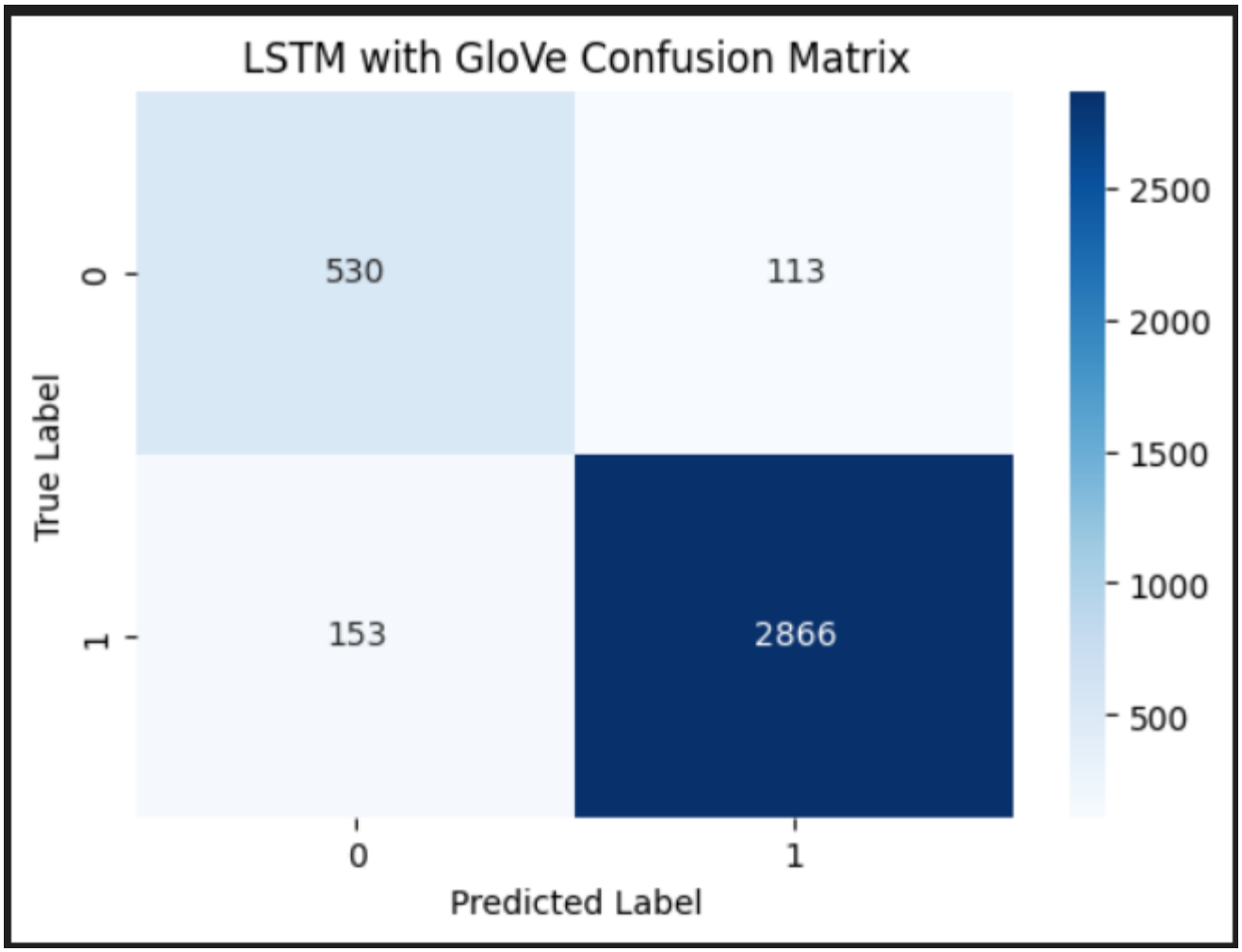


Figure 10: Two Classes Confusion Matrix

5. Conclusion and Future Work

The sentiment analysis project demonstrates the LSTM model's higher performance using Word2Vec embeddings, outperforming LSTM (93% and 59%) and SimpleRNN (80% and 58.7%) with 93% accuracy in binary classification and 61% accuracy in five-class classification. In binary tasks, LSTM's 90% validation accuracy and stable training were guaranteed by its capacity to capture long-term dependencies, as opposed to SimpleRNN's 70%. With pretrained semantics, Word2Vec improved five-class performance, but its binary impact was negligible. Due to data imbalance, overfitting, and poor convergence, continued even after oversampling. Training on a Google Colab T4 GPU takes two to five minutes. To further enhance and broaden the model's applicability, future research might incorporate a bigger dataset, hyperparameter tuning, attention processes, multilingual evaluations, or real-time commercial implementation.

6. Reference

Ishaq, A., 2020. Extensive Hotel Reviews Classification using Long Short-Term Memory. *Journal of Ambient Intelligence and Humanized Computing*, 12(10).