ADVANCE MACHINE LEARNING

ASSIGNMENT 4: IMDB Review Classification Using RNNs

Under the guidance of:

Prof. Chaojiang (CJ) Wu, Ph.D.

Department of Business Analytics

November 12th, 2025

Submitted by:

Name: Shrishty Kashyap

Student Id: 811299714

Course code: 64061-001

INTRODUCTION

This assignment focuses on applying Recurrent Neural Networks (RNNs) on the IMDB movie review dataset to understand how different word embedding strategies perform when we have limited training data. The reviews are shortened to 150 words, and the vocabulary is restricted to the top 10,000 most frequent tokens. At first, the model is trained on only 100 samples while keeping 10,000 reviews for validation. Two different embedding methods are compared before feeding into the Bidirectional LSTM layer, one using a standard trainable embedding layer and the other using pretrained GloVe word embeddings. After that, the same experiment is repeated by gradually increasing the number of training samples to see how the performance changes. The main goal is to identify the point where the trainable embedding starts performing better than the pretrained GloVe embedding. Through this analysis, the assignment aims to find which word representation method is more effective in improving the overall prediction accuracy for text classification tasks like sentiment analysis.

DATASET OVERVIEW AND PREPROCESSING

The IMDB movie review dataset includes 50,000 labeled reviews, equally divided between positive and negative sentiments. For this assignment, each review was shortened to a maximum of 150 words to keep the input length consistent and make the training process more efficient. The vocabulary was limited to the top 10,000 most frequent words to manage sparsity and focus on the most meaningful terms. After preprocessing, the reviews were tokenized, converted into integer sequences, and padded to maintain equal sequence lengths for all inputs. To create a low-data scenario, only 100 samples were used for training, while 10,000 samples were kept for validation. This setup helps analyze how the model performs when data availability is extremely limited and allows for a fair comparison of embedding strategies under such restricted conditions.

MODEL DEVELOPMENT

Two RNN-based models were created for this assignment, differing only in how they handle word embeddings. The first model used a trainable Keras Embedding layer, allowing the network to learn word representations directly from the IMDB data during training. The second model used pretrained GloVe embeddings that were kept fixed (non-trainable) while being passed through the same Bidirectional LSTM setup. Both models followed an identical structure a Bidirectional LSTM layer with 32 units, followed by a dropout layer to help reduce overfitting, and a sigmoid activation layer for binary classification of sentiment. Each model was trained for 10 epochs with validation monitoring enabled to prevent overfitting, and their performances were evaluated using both validation and test datasets to compare how each embedding strategy influenced accuracy under different data conditions.

METHODOLOGY

The experiment was designed as a controlled comparison between the two embedding approaches. The steps followed in this process are outlined below:

- **Initial Training:** Both models were first trained using only **100 samples** to examine their performance under extremely limited data conditions.
- **Incremental Scaling:** The training size gradually increased to **1,000**, **3,000**, **5,000**, **10,000**, and **20,000** samples to identify the point where the trainable embedding starts outperforming the pretrained GloVe embeddings.
- **Consistency in Setup:** For every training size, the **same model architecture**, **hyperparameters**, and **preprocessing steps** were maintained to ensure fair and unbiased comparison.
- **Performance Tracking:** Throughout all experiments, key metrics **training accuracy**, **validation accuracy**, **test accuracy**, and **validation loss** were recorded and visualized to analyze performance trends as the amount of training data increased.

RESULTS

We compared the performance of the two models across different training sample sizes to see how each embedding approach behaves as more data is available. With very small data (100 samples), the pretrained embeddings performed better, since they already carry rich word knowledge from a large external corpus. As the data increased to 1,000–5,000 samples, the results fluctuated, showing that this is a transition phase where neither model consistently outperforms the other.

A clear trend emerged with larger datasets. At 10,000 and 20,000 samples, the trainable embeddings gave higher test accuracy than the pretrained ones. With more data, the model could learn sentiment-specific word patterns directly from the IMDB reviews, giving it an edge over fixed pretrained embeddings. In short, pretrained embeddings work best with very small datasets, but trainable embeddings become better when enough training data is available.

Test Accuracy Comparison Across Training Sample Sizes

Training Samples	**Trainable Embedding Test Accuracy**	**Pretrained Embedding Test Accuracy**	**Better Model**
100	0.761	0.780	Pretrained
1,000	0.787	0.772	Trainable
3,000	0.788	0.795	Pretrained
5,000	0.783	0.786	Pretrained
10,000	0.802	0.790	Trainable
20,000	0.790	0.772	Trainable

ANALYSIS OF ARCHITECTURAL CHOICE

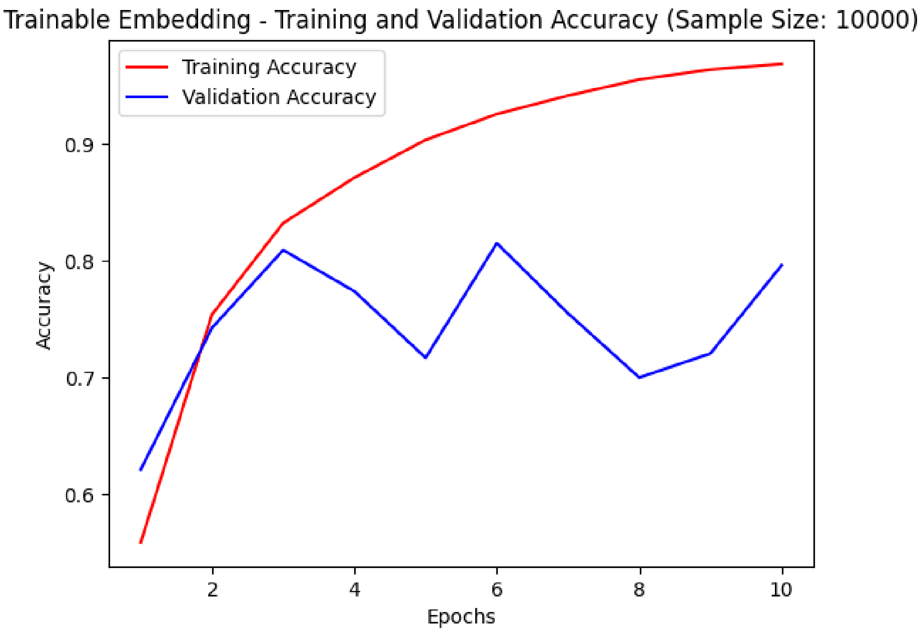
The Bidirectional LSTM architecture was selected because sentiment depends on understanding context from both directions in a review. Pretrained GloVe embeddings performed better with small training sizes because they already contain meaningful semantic relationships learned from large corpora. However, they are fixed and cannot adapt to IMDB-specific patterns. The trainable embedding layer starts with random weights but improves as more data becomes available, eventually learning task-specific word patterns. This is why it performs better once the training size becomes large.

BEST MODEL PERFORMANCE

Across all experiments, the best overall performance was achieved by the **trainable embedding model** at a training size of **10,000 samples**, reaching a test accuracy of **0.802**, which was the highest among all configurations. This indicates that once enough labeled data is available, learning embeddings directly from the IMDB reviews leads to stronger, task-specific representations compared to fixed GloVe embeddings. Although pretrained embeddings performed better in low-data settings, they could not adapt as effectively as the trainable layer when more data was introduced. This demonstrates that data availability strongly influences which embedding strategy is more effective.

VISUALIZING THE BEST MODEL: LEARNING CURVES

The learning curves for the best-performing model trainable embeddings with 10,000 training samples showed steady improvement throughout the training process. Training accuracy increased smoothly across epochs, while validation accuracy closely followed, indicating that the model was generalizing well without significant overfitting. The validation loss curve also showed a consistent downward trend, reinforcing that the model was learning meaningful patterns from the data. These curves highlight that, with enough training samples, the trainable embedding model can effectively learn sentiment-specific word relationships and maintain stable performance across both training and validation sets.



CONCLUSION

This assignment demonstrated how training data size and embedding strategies influence the performance of RNN-based sentiment classification models. Pretrained GloVe embeddings performed best when the training data was extremely limited, providing strong initial word representations that helped the model generalize early. However, as the training size increased, the trainable embedding model gradually improved and eventually achieved the highest accuracy at 10,000 samples. This shows that pretrained embeddings are ideal for low-resource settings, while trainable embeddings excel once enough data is available to learn task-specific patterns. Overall, the experiment highlights the importance of data availability and embedding choice in building effective text-classification models.