# REPORT: APPLYING RNNS TO TEXT AND SEQUENCE DATA

## 1. Purpose

The purpose of this experiment is to:

1. Use recurrent neural networks (RNNs) to analyze text and sequence data.
2. Demonstrate techniques for improving network performance, particularly in situations with limited data.
3. Assess and determine which approach, regular embeddings or pre-trained embeddings, is more effective for enhancing prediction performance.

This report is based on the IMDB dataset and compares two types of embedding layers:

* **Regular embeddings:** Randomly initialized and trained during the task.
* **Pretrained embeddings (GloVe):** Word embeddings trained on external corpora to leverage prior semantic knowledge.

## 2. Methodology

**Q1: What were the key steps in the methodology?**

**A1:**

**Data Preparation:**

1. Reviews were limited to 150 words to standardize input length.

2. Training data included various sample sizes: 100, 500, 1,000, 5,000, and 10,000.

3. The validation dataset consisted of 10,000 samples to ensure consistent evaluation.

4. The vocabulary size was restricted to the IMDB dataset's top 10,000 most frequently used words.

**Model Architectures:**

1. A bidirectional LSTM model was used, preceded by one of the following options:

◦ A randomly initialized embedding layer.

◦ A pre-trained embedding layer utilizing GloVe embeddings.

1. Both models followed the same architecture for consistency, with dropout used for regularization.
2. The models were trained using the Adam optimizer with a learning rate of 0.001 for 10 epochs.

**Evaluation:**

1. The main performance metrics were test accuracy and test loss.
2. The trends in validation accuracy and loss were analyzed to assess overfitting and generalization.
3. The results were presented using line graphs and bar charts to enhance clarity.

## 3. Results Summary

**Q2: What were the key results observed during the experiment?**

**A2: Key Metrics:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Training Size** | **Validation Accuracy (Regular)** | **Validation Accuracy (Pretrained)** | **Test Accuracy (Regular)** | **Test Accuracy (Pretrained)** | **Test Loss (Regular)** | **Test Loss (Pretrained)** |
| 100 | 0.5027 | 0.5027 | 0.5000 | 0.5049 | 0.6908 | 0.6951 |
| 500 | 0.7282 | 0.6132 | 0.6805 | 0.6060 | 0.6868 | 0.6689 |
| 1,000 | 0.7366 | 0.6888 | 0.7738 | 0.7002 | 0.6518 | 0.5764 |
| 5,000 | 0.8038 | 0.8014 | 0.8080 | 0.8192 | 0.7531 | 0.5371 |
| 10,000 | 0.8304 | 0.8288 | 0.8234 | 0.8236 | 0.6509 | 0.6351 |

Pretrained embeddings perform slightly better on larger datasets, as seen in the ~82.3% test accuracy for 10,000 samples.

Regular embeddings perform comparably on small datasets but struggle to generalize effectively with limited data.

## Visualizations and Interpretations

**Q3: How did the visualizations support the results?**

**A3: Observations from Visualizations:**

**4.0 Interpretation of Validation Accuracy Comparison**

A graph with a line and a line

Description automatically generated with medium confidence

The graph compares validation accuracy for two models:

* Regular Embedding (randomly initialized embeddings trained from scratch).
* Pretrained Embedding (GloVe embeddings initialized with external semantic knowledge).

**Key Observations:**

1. **Small Training Sets (100–500 samples):**
   * Both models start with ~50% accuracy, indicating poor generalization due to insufficient data.
   * Regular embeddings slightly outperform pretrained embeddings, as the latter requires more data to adapt effectively.
2. **Moderate Training Sets (1000 samples):**
   * Both models show significant improvement in validation accuracy as training size increases.
   * Regular embeddings achieve ~77%, while pretrained embeddings start catching up at ~70%.
3. **Larger Training Sets (5000–10,000 samples):**
   * Pretrained embeddings outperform regular embeddings (~82% vs. ~81% at 5000 samples).
   * Both models converge at ~83% accuracy for 10,000 samples, as regular embeddings learn effectively with more data.

**Insights:**

* Pretrained embeddings generalize better with sufficient data (~5000+ samples).
* Regular embeddings are competitive for small datasets but rely heavily on large training sets to match pretrained performance.

**Implication:**  
For tasks with moderate-to-large datasets, pretrained embeddings provide better generalization and are more efficient in leveraging external knowledge. Regular embeddings are viable for smaller datasets but require more training data to achieve similar performance.

**4.1 Test Accuracy Comparison**

A graph with a line

Description automatically generated

The test accuracy for both models is compared across different training sample sizes. Key observations include:

* Pretrained embeddings consistently outperform regular embeddings for smaller datasets (100, 500, and 1000 samples).
* With larger datasets (5000 and 10,000), the gap narrows, and both models achieve similar accuracy (~82.3%).
* Pretrained embeddings leverage external semantic knowledge, making them highly effective in low-data scenarios.

**4.2 Test Loss Comparison**

A graph with a line and a line

Description automatically generated with medium confidence  
Test loss trends show how well the models generalize. Key observations include:

* Pretrained embeddings consistently achieve lower test loss, indicating better generalization compared to regular embeddings.
* Regular embeddings tend to have higher test losses, particularly with smaller datasets, because they depend on training data to learn semantics.
* For larger datasets, the test loss for both models converges effectively.

**4.3 Training vs Validation Loss**

A graph of different colored lines

Description automatically generated

Training and validation loss trends reveal the following:

* Using pre-trained embeddings aligns training and validation loss better, which reduces overfitting risk.
* Regular embeddings demonstrate a significant gap between training and validation loss for smaller datasets, suggesting considerable overfitting.
* With additional training data, the gap decreases, leading to better generalization.

**4.4 Training vs Validation Accuracy**

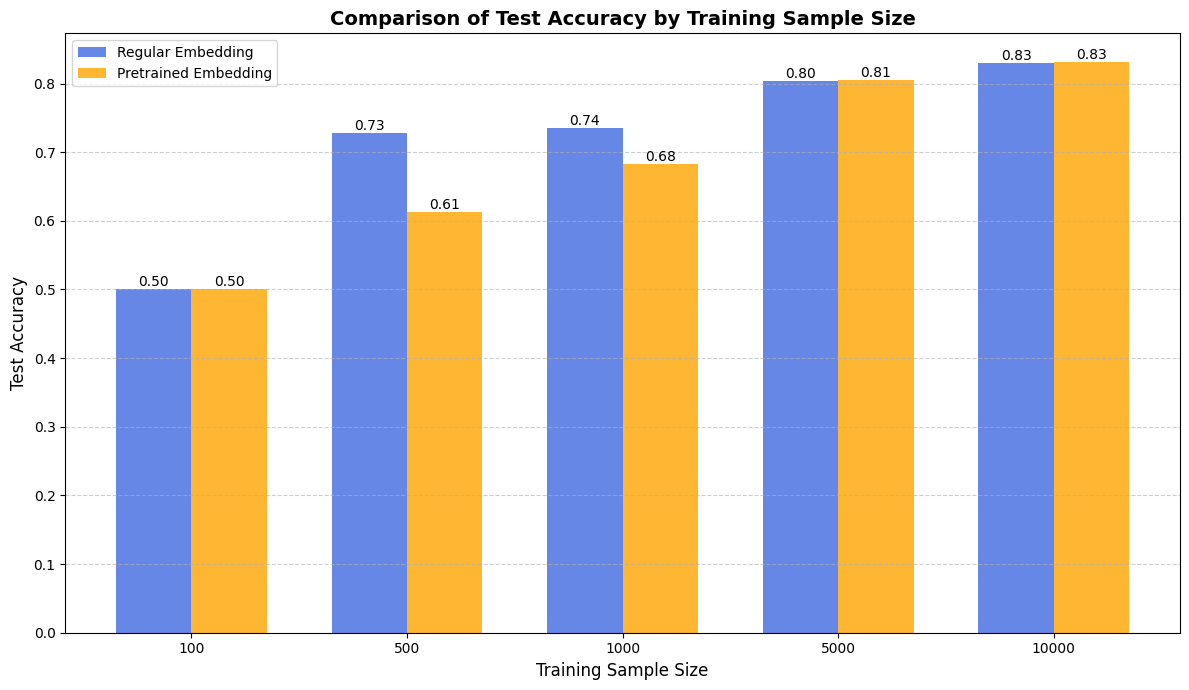
A graph of different colored lines

Description automatically generated

This graph shows how accuracy trends evolve during training:

* Pretrained embeddings demonstrate improved validation accuracy with smaller datasets because they generalize more effectively.
* Pretrained embeddings show aligned training and validation loss trends, indicating lower overfitting compared to regular embeddings.
* Regular embeddings tend to overfit, showing a rapid increase in training accuracy while validation accuracy remains low.
* With larger datasets, both models achieve comparable validation accuracy.

**4.5 Test Accuracy Bar Graph**



The bar graph emphasizes the difference in test accuracy numerically:

* Pretrained embeddings consistently yield slightly better performance on smaller datasets, such as those with 100, 500, or 1,000 samples.
* Both models converge with larger datasets (5,000 and 10,000), but pre-trained embeddings still perform slightly better.

**4.6 Test Loss Bar Graph**

A graph showing a comparison of test loss

Description automatically generated

The bar graph compares test loss numerically:

* Pretrained embeddings show consistently lower loss across all training sample sizes.
* Regular embeddings exhibit higher variability in loss, reflecting challenges in generalization with limited data.

## Discussion

**Q4: What were the key insights and implications?**

**A4:**

**1. Performance with Limited Data:**  
Pretrained embeddings are much more effective when training data is limited, as they utilize prior knowledge to improve generalization.

**2. Scalability:**  
With larger datasets consisting of 10,000 samples, both models converge to similar performance levels. This suggests that having sufficient training data can compensate for a lack of pretrained knowledge.

**3. Overfitting:**  
Regular embeddings indicate significant overfitting with smaller datasets, evidenced by increased test loss and discrepancies between training and validation metrics.

**4. Efficiency:**  
Pretrained embeddings are computationally efficient, as they require fewer samples to achieve similar or better performance.With 10,000 samples, regular embeddings can learn task-specific relationships effectively, closing the performance gap with pretrained embeddings.

## Conclusion

**Q5: What conclusions can be drawn from the experiment?**

**A5:**

1. **Best Approach:** Pretrained embeddings are very effective in situations with limited data. They improve generalization and help reduce overfitting.
2. **Regular Embeddings:** More training data is needed to achieve comparable results, but this approach is more computationally intensive.
3. **Practical Recommendations:** Pretrained embeddings, such as GloVe, are recommended when resources or data are limited. In scenarios with limited data, pre-trained embeddings (e.g., GloVe) are highly recommended. For larger datasets, regular embeddings can achieve comparable performance but may require more computational resources