Generative AI Question 04

Kissa Zahra[1 [i21−0572]]

FAST National University of Computer and Emerging Sciences, Islamabad H-11, Pakistan kissasium@gmail.com

Abstract. Training two Vanilla RNN models for next-word prediction on Shakespeare text: one with random embeddings and another with pretrained GloVe embeddings.

Keywords: RNN · Neural Network · Language Modeling · Shakespeare · GloVe · Next-Word Prediction

1 First Section

1.1 Introduction

Implementation and analysis of a Vanilla Recurrent Neural Network (RNN) for next-word prediction on Shakespeare's text. The study focused on building a custom RNN architecture from scratch, training it on Shakespeare's works, and investigating the impact of different word embedding strategies on model performance. Specifically, we compared randomly initialized embeddings against pretrained GloVe embeddings to understand their influence on the model's ability to predict and generate Shakespearean text.

1.2 Methodology

Dataset I used the 'tiny_shakespeare' dataset from Hugging Face, containing works by William Shakespeare. After preprocessing, the data yielded a vocabulary of 21,949 unique words. The dataset was split into training (80%) and testing (20%) sets, with input sequences of length 5 created to predict the 6th word.

Architecture The implemented architecture consisted of three main components:

- **Custom RNN Cell:** We implemented a vanilla RNN cell from scratch using the formula:

$$h_t = \tanh(W_{ih} \cdot x_t + b_{ih} + W_{hh} \cdot h_{t-1} + b_{hh}) \tag{1}$$

  where h_t is the hidden state at time t, x_t is the input at time t, and W_{ih}, W_{hh}, b_{ih}, and b_{hh} are learnable parameters.

- **Embedding Layer:** An embedding layer of dimension 50 was used to convert word indices to dense vectors. Two variants were tested:
  - Randomly initialized embeddings
  - Pretrained GloVe embeddings (6B.50d version)
- **Output Layer:** A fully-connected layer mapping the hidden state to logits over the vocabulary.

Training Details The models were trained with the following parameters:

- Embedding dimension: 50
- Hidden state dimension: 128
- Batch size: 64
- Learning rate: 0.001
- Optimizer: Adam
- Loss function: Cross-entropy
- Epochs: 10

For the GloVe embeddings, we found coverage for 7,808 out of 21,949 words (35.6%) in the Shakespeare vocabulary.

Evaluation Metrics We evaluated model performance using:

- **Word-level accuracy:** Percentage of correctly predicted next words
- **Perplexity:** Exponential of the average negative log-likelihood, measuring model uncertainty
- **Qualitative assessment:** Quality of generated text samples

2 Second Section

2.1 Results

Training Dynamics Both models showed steady decreases in training loss over the 10 epochs:

- **Random Embeddings:** Training loss decreased from 7.7687 to 3.3965
- **Pretrained Embeddings:** Training loss decreased from 7.6010 to 4.7434

Validation loss increased for both models over time (from $\sim$7.5 to $\sim$9.0), indicating some degree of overfitting, which is common for language models especially with limited data and simpler architectures.

Performance Comparison The final performance metrics showed clear advantages for pretrained embeddings:

Key observations:

- The pretrained embeddings model achieved a 6% relative improvement in accuracy
- More significantly, it showed a roughly 27% reduction in perplexity
- Lower perplexity indicates the model is less "surprised" by the test data, showing better generalization

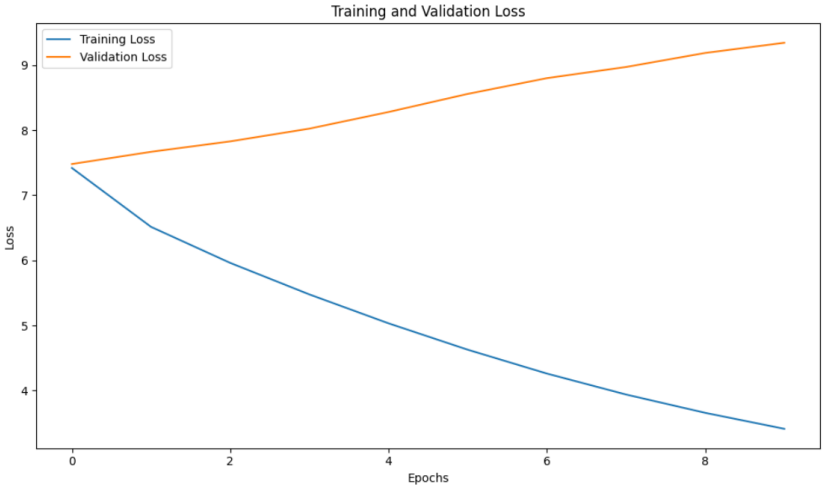

Fig. 1. Random Embeddings

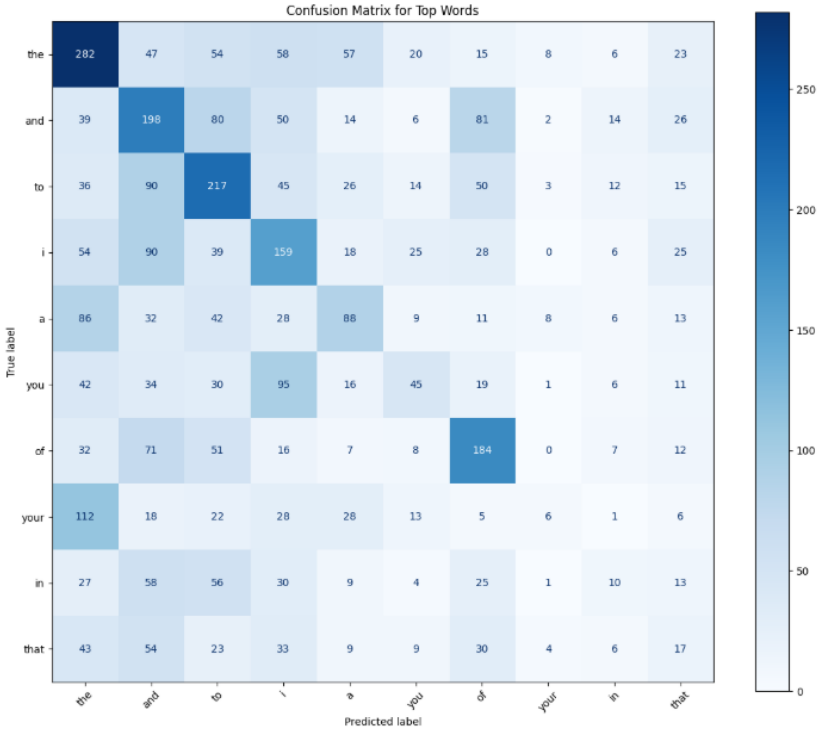

Fig. 2. Confusion Matric for Random Embeddings

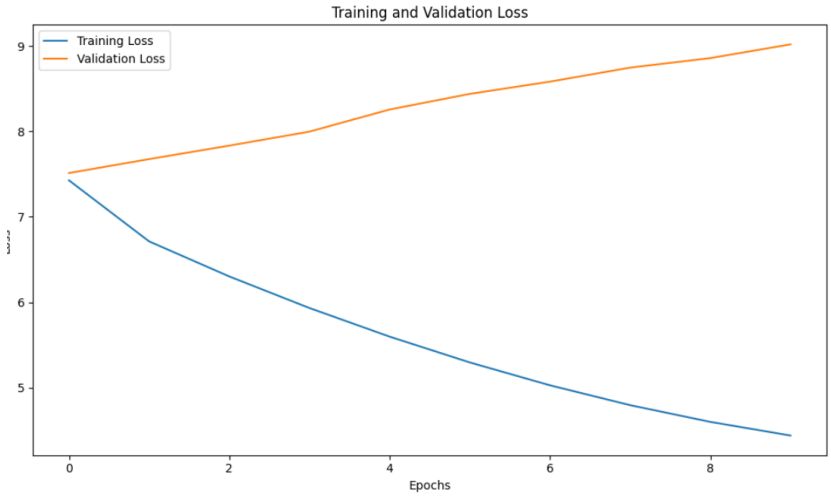

Fig. 3. Pretrained Embeddings

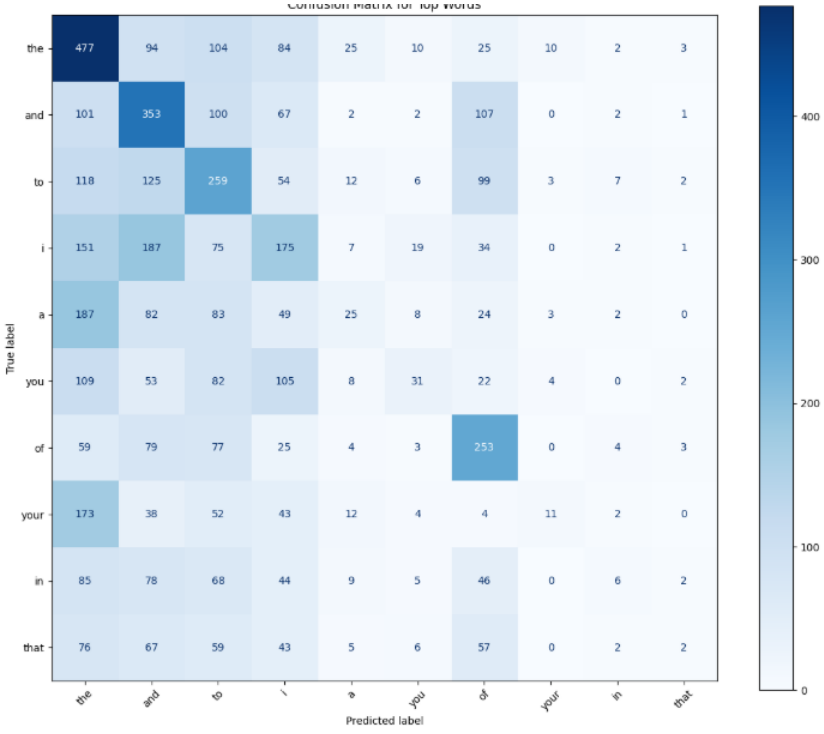

Fig. 4. Confusion Matric for Pretrained Embeddings

Table 1. Performance comparison between random and pretrained embeddings

Embedding Type	Word-Level Accuracy	Perplexity
Random Embeddings	0.0536	11,567.17
Pretrained Embeddings	0.0604	7,893.16

2.2 Text Generation

Text generated from the seed phrase "to be or not to":

- **Random Embeddings:** "to be or not to remember his lady's blazon citizens: come, come, yield you raging"
- **Pretrained Embeddings:** "to be or not to deny him truth therefore, being join'd benvolio: marry, 'tis now"

Both outputs successfully captured key elements of Shakespearean language, including character references (e.g., "benvolio"), archaic phrasings ("'tis", "being join'd"), and formal speech patterns. The models demonstrate an ability to generate text with the distinctive style and vocabulary characteristic of Shakespeare's works.

3 Third Section

3.1 Analysis

Impact of Pretrained Embeddings The results confirmed the hypothesis that pretrained embeddings improve model performance, even for specialized text like Shakespeare. The significant reduction in perplexity (roughly 27%) demonstrates that the semantic information captured in GloVe embeddings helps the model make more confident and accurate predictions.

Limitations of Vanilla RNN While the models showed clear learning, the absolute accuracy values remained modest (around 5-6%). This is expected due to:

- The inherent limitations of vanilla RNN architectures, which struggle with capturing long-range dependencies
- The complexity and uniqueness of Shakespearean language
- The large vocabulary size (21,949 words)
- The challenging nature of next-word prediction, where multiple continuations may be equally valid

The increasing validation loss indicated that the model was memorizing training patterns rather than generalizing perfectly, a common challenge with simpler RNN architectures.

4 Fourth Section

4.1 Conclusion

The comparison between randomly initialized and pretrained embeddings demonstrated that leveraging pretrained word representations significantly improves model performance, even for domain-specific text.

Key findings include:

- Pretrained GloVe embeddings reduced perplexity by roughly 27% compared to random embeddings
- Word-level accuracy showed modest but consistent improvement with pretrained embeddings