Word Completion Using LSTM: Predicting the Next Word in Shakespeare's Plays

Name : Manahil Sarwar
Roll Number : 21I-0293
Section : AI-K
National University of Computing and Emerging Sciences

***Abstract*—This report presents the development of a word-level Long Short-Term Memory (LSTM) model for sentence completion, trained on lines from Shakespeare's plays. The model predicts the next word in a sequence and provides real-time word suggestions through a Flask-based web interface. The report covers the preprocessing steps, model architecture, and the coherence of generated sentences. Challenges and model performance are analyzed, especially in terms of sentence fluency and coherence.**

I. Introduction

The objective of this assignment is to create a word-level LSTM model that can complete sentences by predicting the next word in a sequence. The model is trained on a dataset consisting of Shakespeare's plays, aiming to capture the style and language structure. A web-based interface was developed using Flask, HTML, and CSS, allowing users to input partial sentences and receive real-time word suggestions. This report details the methodology, results, and challenges encountered during the development process.

II. Methodology

A. Dataset

The dataset consists of lines from Shakespeare's plays, which were downloaded from Kaggle[1]. The text was pre-processed by tokenizing the words, and sequences of a fixed length were generated for training the model.

B. Preprocessing Steps

The preprocessing of the dataset involved the following steps:

- Loading the text data from a file.
- Tokenizing the text using the Keras `Tokenizer` class.
- Converting the text into sequences of words.
- Defining a sequence length of 5 words.
- Creating a data generator to provide input and output sequences in batches for model training.

C. Model Architecture

The LSTM model was designed using Keras with the following layers:

- An embedding layer with an input dimension equal to the vocabulary size and an output dimension of 100.
- A single LSTM layer with 150 units.
- A dense layer with a softmax activation function to predict the next word.

The model was compiled with the categorical cross-entropy loss function and the Adam optimizer, and was trained over 10 epochs using a batch size of 64.

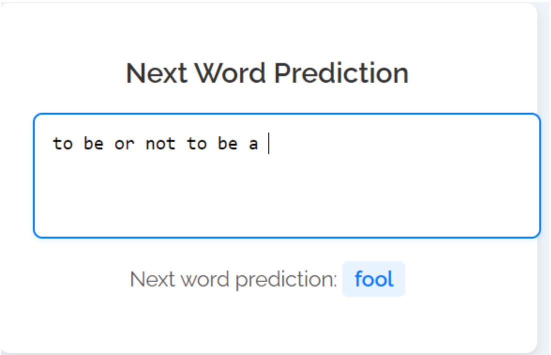
D. User Interface

The frontend was built using Flask, HTML, and CSS. The user interface allows users to input a partial sentence, and the model provides real-time word suggestions. As the user types, the interface dynamically updates with the next word prediction.

III. Results

A. Examples of Predictions

The model performed well on several sentences, as shown in the examples below. These examples showcase how the LSTM model successfully completed sentences based on its training on Shakespeare's language style after just been given the starting words.

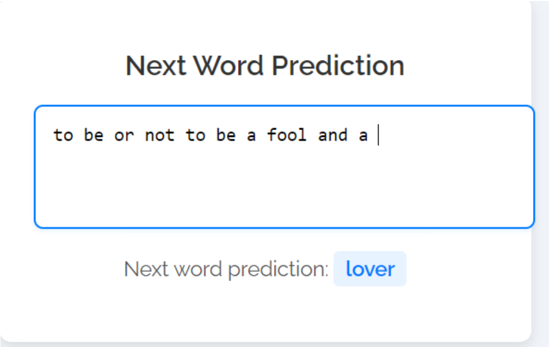


to be or not to be a |

Next word prediction: fool

Fig. 1. Example of successful sentence completion.

[1] https://www.kaggle.com/datasets/kingburrito666/shakespeare-plays

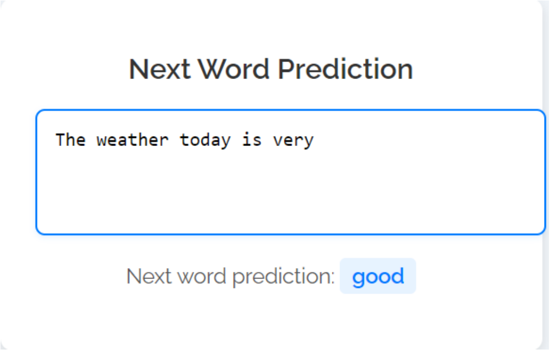


Next Word Prediction

to be or not to be a fool and a

Next word prediction: lover

Fig. 2. Example of successful sentence completion.

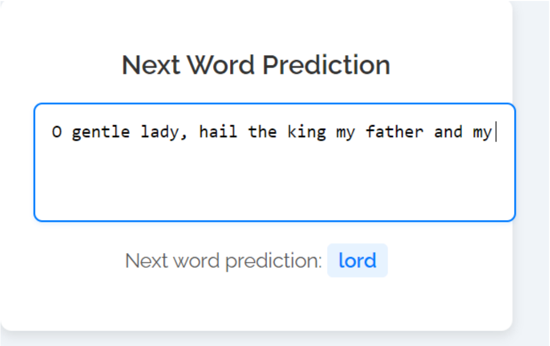


Next Word Prediction

The weather today is very

Next word prediction: good

Fig. 3. Example of successful sentence completion.



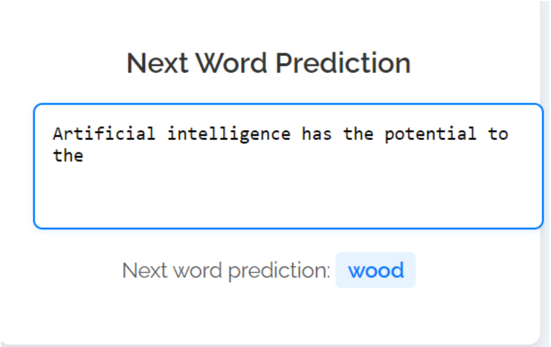
Next Word Prediction

O gentle lady, hail the king my father and my

Next word prediction: lord

Fig. 4. Example of successful sentence completion.

However, the model also encountered some difficulties in predicting the next word in sentences whose vocabulary probably was not present in the Shakespeare's plays like:



Next Word Prediction

Artificial intelligence has the potential to the

Next word prediction: wood

Fig. 5. Example of a sentence where the model performed poorly.

IV. Discussion

The model's performance improved over time as training progressed. Initially, the predictions were often incoherent, but by the later epochs, the model was able to generate more meaningful and contextually appropriate words. The primary challenge was the limited training dataset, which restricted the model's ability to generalize beyond Shakespeare's unique style of writing.

In some cases, the model struggled with sentences that had less frequent words or complex structures, which resulted in nonsensical word predictions. This is likely due to the relatively small size of the dataset and the constraints of the LSTM architecture.

V. Conclusion

In this project, we successfully developed a word-level LSTM model for sentence completion using Shakespeare's plays. The model was integrated into a web interface, providing real-time word predictions as users typed. While the model showed promising results, further improvements can be made by using a larger dataset, fine-tuning hyperparameters, or using more advanced architectures like transformers. This experiment demonstrates the power of LSTM networks in natural language processing tasks like sentence completion.

VI. Prompts

Here are the prompts given during the evaluation of the model:

- ”To be or not to be a _____”
- ”To be or not to be a fool and a _____”
- ”The weather today is very _____”
- ”O gentle lady, hail the king my father and my _____”
- ”Artificial Intelligence has the potential to the _____”

VII. References

References

[1] Dataset: Shakespeare Plays, Available: https://www.kaggle.com/datasets/kingburrito666/shakespeare-plays

[2] Keras Documentation: https://keras.io/