NEXT WORD PREDICTION

Computer Science and Engineering Department (CSED)
National Institute of Technology, Calicut
Kerala, India

Course: ARTIFICIAL INTELLIGENCE (CS6172D)

Team members:

Durgesh Singh Munda (M220285CS) durgesh_m220285@nitc.ac.in
Himanshu Dewangan (M220263CS) himanshu_m220263cs@nitc.ac.in
Meegada Bhavan Kumar Reddy (M220266CS) meegada_m220266cs@nitc.ac.in

Abstract

Next-word prediction, also called language modeling, is one field of natural language processing as it can improve the efficiency and accuracy of language input systems by predicting the next word. One approach to next-word prediction is to use Long Short-Term Memory (LSTM) networks, a recurrent neural network (RNN) type that can model long-term dependencies in sequential data. In this word prediction project, we are using Bi-LSTM.

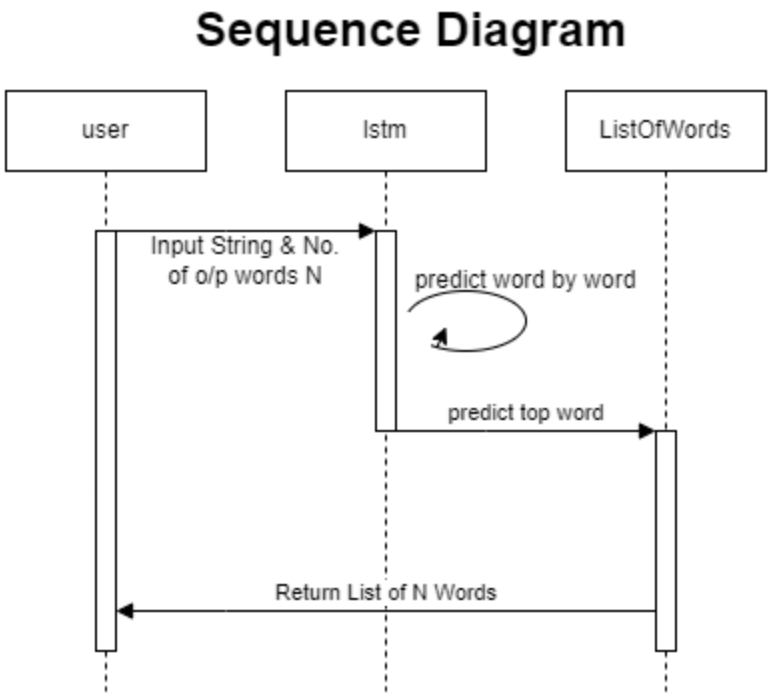
Summary of code:

- Import necessary libraries/modules including Pandas, Numpy, and TensorFlow.
- Read a CSV file containing medium data and pre-processes the 'title' column.
- Tokenize the pre-processed text data using the Tokenizer class from Keras.
- Create input sequences by generating n-grams from the tokenized text data.
- Pad the input sequences to ensure that all sequences have the same length.
- Split the input sequences into features (xs) and labels (ys) and convert the labels to categorical format.
- Build a Sequential model in Keras with an Embedding layer, Bidirectional LSTM layer, and Dense layer.
- Compile the model with an optimizer, loss function, and metrics.
- Fit the model on the input sequences and labels for 25 epochs and plot the accuracy and loss graphs.
- Generate new text data based on a seed text using the trained model.

Technologies and libraries used:

Pandas, Numpy, TensorFlow, Keras, Tokenizer, LSTM (Long-Short-Term-Memory), Bidirectional, Adam

Diagrams:



Output

```
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 33ms/step
Once upon a time there was a boy that cause information leakage transforming the government decisions – data collection
```

Input given: “Once upon a time there was a boy that”

Output by the model “Once upon a time there was a boy that cause information leakage transforming the government decisions - data collection”

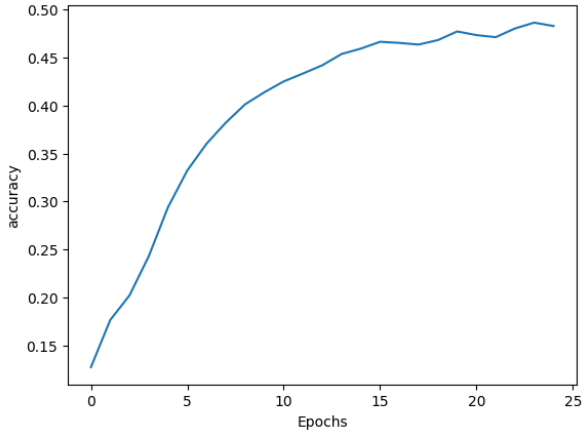


Figure Left: Plot between Accuracy & Epochs

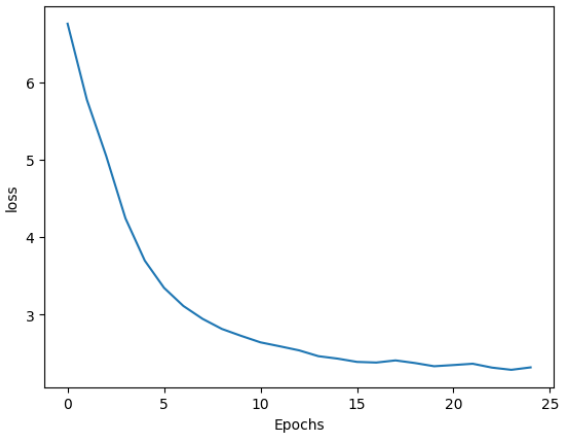


Figure Right: Plot between Loss & Epochs