**NEXT WORD PREDICTON USING LSTM**

**A MINOR-I PROJECT REPORT**

***Submitted by***

**SHAIK KHAIS MOHAMMED ALI**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**COPUTER SCIENCE ENGINEERING**

***Under the supervision of***

**SYED ALI MEHDI**



**Department of Computer Science& Engineering**

**School of Engineering Sciences & Technology**

**JAMIA HAMDARD**

**New Delhi-110062**

**(2022)**

DECLARATION

I, **Mr.Shaik Khais Mohammed Ali** a student of B.TECH,**CSE (Computer Science Engineering),** (**Enrolment No: 2019-310-147)** hereby declare that the Project/Dissertation entitled **“Next Word Prediction Using LSTM ”** which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **B.TECH**,**CSE (Computer Science Engineering),** is my original work and has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

**(Signature and Name of the Applicant)**

**Date:**

**Place:**

Table Of Content

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Chapter Name** | **Page No.** |
| 01 | Abstract | 4 |
| 02 | Introduction | 5 |
| 03 | Reason for Project | 6 |
| 04 | Problem Statement | 7 |
| 05 | Objective | 7-8 |
| 06 | User Function Specification | 8 |
| 07 | Hardware Requirement | 8 |
| 08 | Technology used in Project | 9-10 |
| 09 | Data Flow Diagram (DFD) | 11-12 |
| 10 | Snapshots | 12-14 |
| 11 | Limitation | 14-15 |
| 12 | Conclusion | 15 |
| 13 | REFERENCES | 16 |

**Abstract**

Word prediction model is an assistive technology tool for writing that suggests as a child types. The Word prediction can be used in smartphones, search-engines and other applications, though more advance tools also exist. There the working of the tool is high-lighten. First the text is upload tokenized and put in sequence. Then the main operation of model is processed. The Long short-term memory (LSTM) is used for training with sets of data can make the performance of model nearer to standard, So the prediction of the next word is more accurate by learning what words you personally use the most often and the program will be robust to the person to utilize. The design of model layer by layer, training and testing, cost function, activation is described and also visually presented. The output is included and the project is in python language.

**Introduction**

Natural language processing has long been a study topic with numerous applications in many fields and path of daily life. We frequently text each other, and anytime we try to compose a text, a suggestion appears, attempting to predict the next word we want to enter in the sentence. The applications of natural language processing is prediction is common thing. We've made significant progress in this area of technology, and we can now employ recurrent neural networks to accomplish our human goal. This project deals with how we can use a neural model better than a basic recurrent neural network and use it to predict the next word. We deal with a model called Long short-term memory (LSTM). We can use the TensorFlow library in python for building and training the deep learning model. Here we can make sure of having longer memory of what words are important with help of those Embedding, LSTM and

Dense layer the model is created using python libraries.

**Reason for Project**

In the project the study of Natural Language Processing with Long short-term memory to pre-process and learning sequential data prediction problems is very interesting and familiar with my recent study. This provides greater accuracy for demand forecasters which results in better decision making for the generation of right word. The implementation and understanding the algorithm and the libraries used in the code are straightforward. Further the capability of memory cells, Artificial cells that used in deep learning for training the artificial neural network, to remember the previous context has a major role.

Predicting the next word has a wide range of applications; it is utilized by everyone in any technology that improves typing speed. It is beneficial in communication, business, research, education, and other areas, it also includes daily life application like dairy writing. Because of the role of the word prediction model, communication via message is easy and quick, business outcomes are improved by records and paper that are produced, and data is anatomized to enhance our technology.

**Problem Statement**

The problem with is model is it LSTM needs lots of resources to train, and to train with these large datasets the model will takes time to get the precision. The problem of vanishing gradient is solved but not eliminated completely, it has to evaluate all the nodes one by one. Thus it is a long, complex and hardware-wise it is inefficient as the real-world application need high memory-bandwidth because of linear layer present in each cell which the system usually fails. LSTM are prone to overfitting and there are difficult to regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation function and weights are updated while training a network.

**Objective**

The objective of this project is to boost typing speed and helps to omit errors by guessing the upcoming word.

-Taking resource, a text file ,tokenize the strings and removing alpha-numeric character and the put in Sequential so to process the input.

-The Training of model is done by Embedding, Long short-term memory and dense function will be explained.

-Then the output and the working of model is illustrated and described in this project.

**User Function Specification**

User can give some words or make a sentence as input that it analyze the input and display the output by clicking the next key.

* Entering the input in the input terminal should be valid and no symbols should be included.
* The user get the predicted word displayed on the same plane which can be use or can give new word that can take as data to analyze for further training and increase precision.
* The language is English in which the input, process and output is shown. So this helps the user to do work or make something is only done in English.

**Hardware Requirement**

RAM :512 MB

Hard disk: 10GB

Processor: 1.0 GHz

Devices could be smartphone, computer, laptop, tablets.

**Technology used in Project and Explaination**

**About Python 3.9:**

Python 3.9 is the last version to provide those Python 2 backward compatibility layers, allowing Python project maintainers more time to plan the removal of Python 2 support and the addition of Python 3.9 support. Aliases for Abstract Base Classes, such as collections, in the collection module.

**LSTM, Embedding, Dense Functions and other important libraries -**

Because it has memory cells to retain the previous context, LSTM is used to solve the long-term dependency problem. Deep learning's LSTMs are a complicated topic. It can be difficult to grasp the concept of LSTMs and how terminology like bidirectional and sequence-to-sequence apply to the field.Google had designed and distributed TensorFlow, a Python library for fast numerical computing. It's a foundation library for building Deep Learning models, either directly or through wrapper libraries built on top of TensorFlow to make the process easier.

Keras has an embedding layer for text data that may be utilized with neural networks. The Embedding layer learns embeddings for all of the words in the training dataset using random weights. It necessitates the representation of input data in integers. This data preparation phase can be completed with Keras Tokenizer API. More information on the Embedding layer may be found here.

Following the two LSTM layers are two fully connected or thick layers.

The first dense layer contains 50 units, whereas the second dense layer is our output (activation) layer, which contains the same amount of units as the vocabulary size. Based on the probability, the model will predict the next word from our vocabulary for each input. As a loss function, categorical cross-entropy is used.

Our Sequential model now has five layers: an Embedding layer, two LSTM layers, and two Dense layers. The input length of our model's Embedding layer is set to the length of a sequence, which in this case is three.

Note: We'll have to submit a three-length vector as input to our model for prediction because we split the data for training inputs and training targets 3 to 1.

## **Predicting words:**

After our model has been trained, we may provide encoded input and the ReLU function will provide the three most likely words, as seen below in the snapshorts.

We utilise padding in the code because we trained our model on 3-word sequences, thus when we input 5 words, code ensures that the last three words are taken into account. What happens if we enter fewer than three words? We are not going to achieve the best results! When we enter an unknown word, the error will have printed in the output on that treminal. For optimum results, we can add sequences of length 3(inputs) to 1(target label). Below is the final output of our model predicting the next 3 words based on the previous words.

The Output of the code is shown in the Snapshot section.

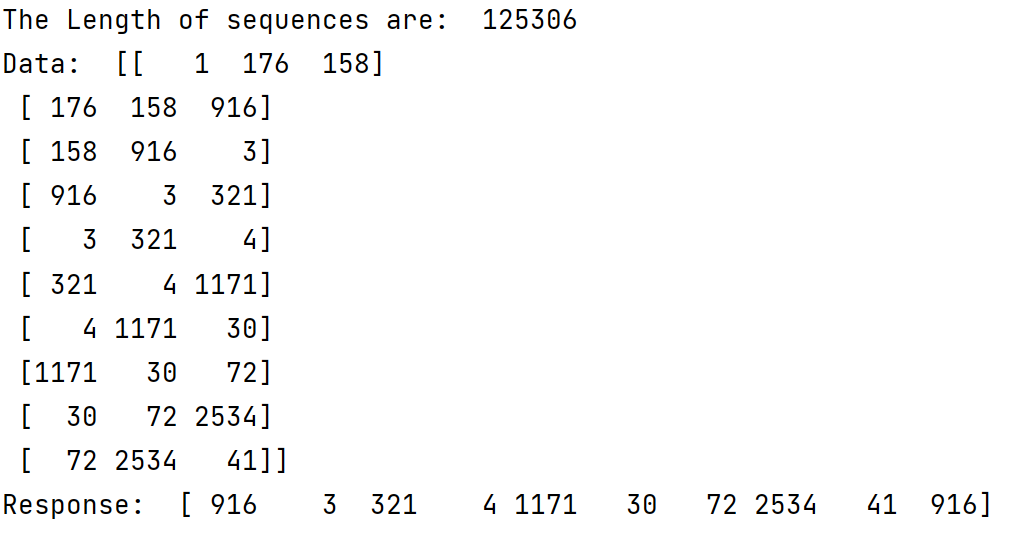
**Data Flow Diagram (DFD)**

A Data Flow Diagram (DFD) is a graphical representation of data that displays data flow and transformations as it goes from input to output.

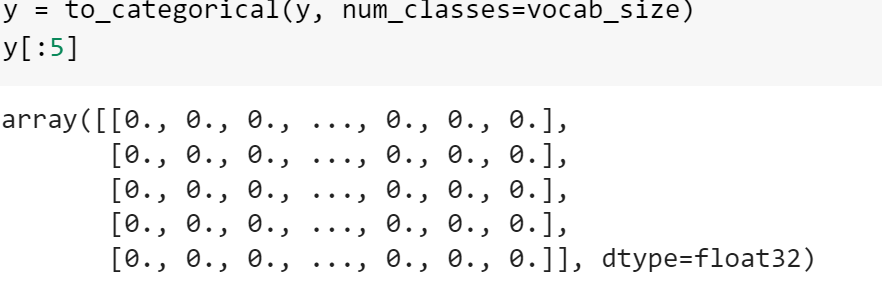
*Fig: DFD of Next word prediction*

*Fig: Model function layers*

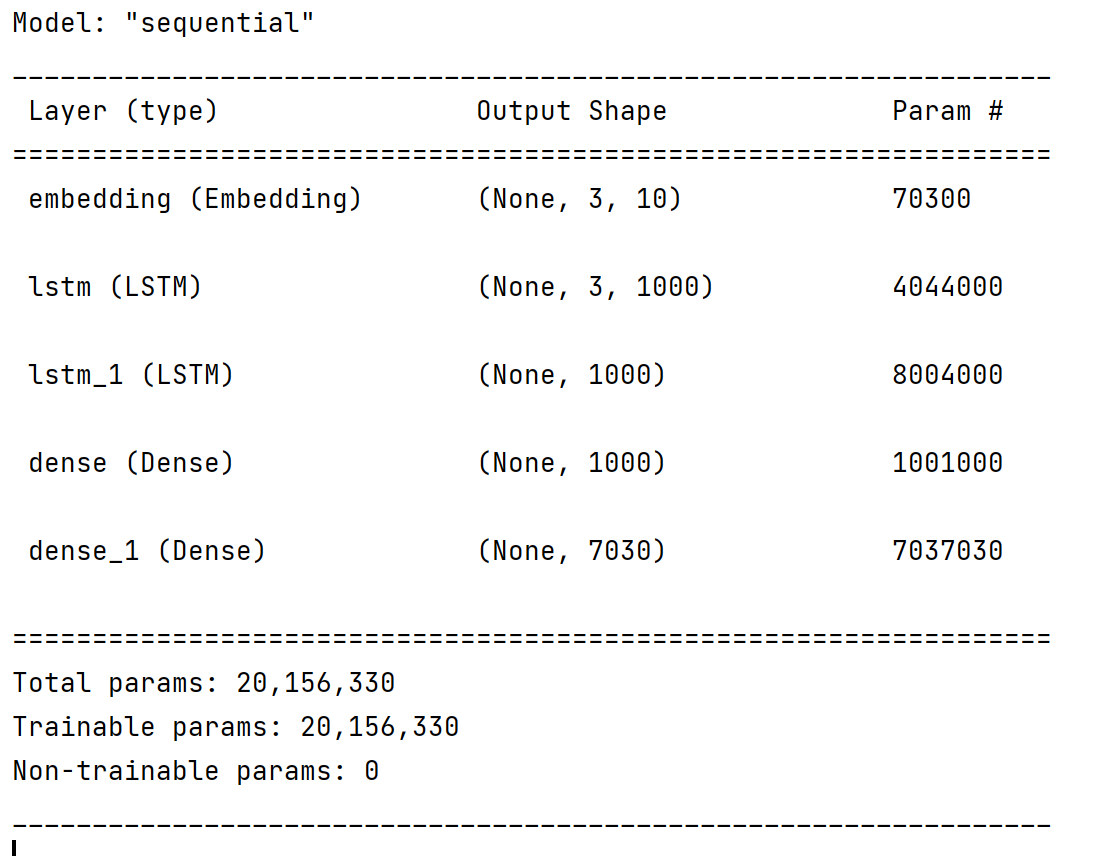
**Snapshots**



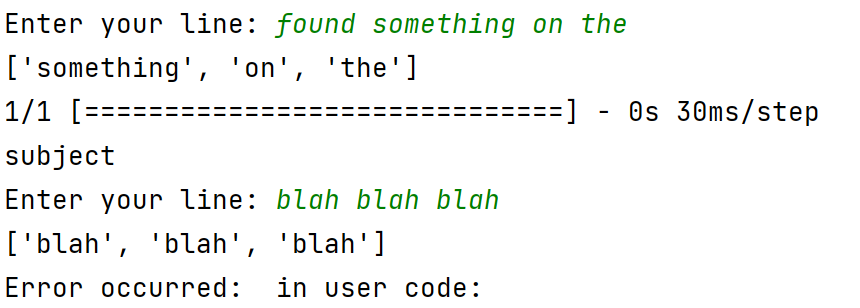
*Fig: Data in Sequence*

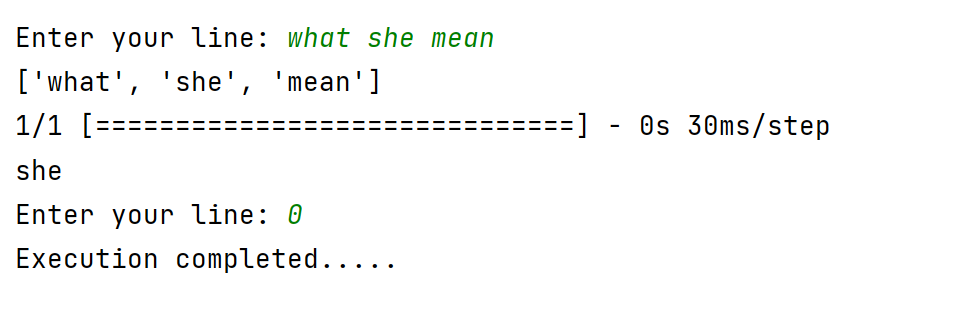


*Fig: Categorical value in matrix for loss calculation*



*Fig: Model parameters and params*





*Fig: The above are output of the LSTM model*

**Limitation**

* The difficulty with this model is that it requires a lot of resources to train, and training with these enormous datasets will take a long time.
* The vanishing gradient problem is solved, but not totally solved; each node must be evaluated separately.
* Require high memory-bandwidth due to the linear layer present un each cell, which is where the system typically fails.
* Input and recurrent connections to LSTM units are probabilistically omitted from activation and weight updates during training a network, making LSTM prone to overfitting and challenging to regularize.

**Conclusion**

Language modelling is the process of predicting the following word, is one of the fundamental tasks of Natural Language Processing (NLP), and it has a wide range of applications. We found that the LSTMs have the memory to remember context from earlier in the text corpus. If you're starting a new project, you might wish to check for open-source implementations of existing pre-trained frameworks on the internet. Nodes within the tree correspond to different document parts like titles, paragraphs and sections. At every node within the document tree, there's a well-defined language model. The language model for a leaf node is predictable directly from the text within the document part related to the node. Inner nodes within the tree are predictable employing a linear interpolation among the various youngster nodes.

**REFERENCES**

<https://ieeexplore.ieee.org/abstract/document/9734151/>

## [*https://towardsdatascience.com/exploring-the-next-word-predictor-5e22aeb85d8f*](https://towardsdatascience.com/exploring-the-next-word-predictor-5e22aeb85d8f)

<https://ieeexplore.ieee.org/abstract/document/8985885>

<https://keras.io/api/layers/core_layers/embedding/#embedding>

<https://www.gutenberg.org/browse/scores/top#authors-last30>

<https://www.analyticsvidhya.com/blog/2021/08/predict-the-next-word-of-your-text-using-long-short-term-memory-lstm/>

<https://androidkt.com/predict-next-word-using-tensorflow-keras/>

<https://www.understood.org/en/articles/word-prediction-technology-what-it-is-and-how-it-works>

<https://towardsdatascience.com/using-deep-learning-for-text-generation-of-wikipedia-articles-c9f1897f54a0>

<http://www.viva-technology.org/New/IJRI/2019/5.pdf>

Output

Input

Has

Uses

Predicts

Takes

Gives

Application

user