**TEXT AUTOCOMPLETE**

Machine Learning Project

### Predicting the next word from the current context using machine learning LSTM model and comparing it with RNN model.

**Report**

### Course: IC272(Data Science-3)

### Group - 47

Team Members:

* Pratibha Mathur
* Prem Shankar
* Shubham Kumar
* Sadiq Ali
* Pritesh Kumar Gupta
* Sonam Shrivastava

Project Examiner – Sushovan

Project Mentor – Aayush Sharma , Yuvraj Mishra

**Contents**

1. [Introduction](#_bookmark0) 1
   1. About project. . . . . . . . . . . . . . . . . . . . . . . 1
2. [Data](#_bookmark3) 1

[2.1 Data preparation](#_bookmark4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1. [Approach](#_bookmark5) 1
2. [Results](#_bookmark6) 2
3. [Discussion](#_bookmark7) 2

[5.1 References and Challenges](#_bookmark8) . . . . . . . . . . . . . . . . . . . . . . . . 2

1. [Conclusion](#_bookmark9) 2
2. [Reflections on own work](#_bookmark10) 2

# Introduction

Natural language processing has been an area of research and used widely in different applications. We often love texting each other and find that whenever we try to type a text a suggestion poops up trying to predict the next word we want to write. This process of prediction is one of the applications NLP deals with. We have made huge progress here and we can use Recurrent neural networks for such a process. There have been difficulties in basic RNN so we have used LSTM also to show the difference between both the models and proving how LSTM is better than RNN.

We can use the TensorFlow library in python for building and training the deep learning model.

**Why use LSTM?**

Vanishing gradient descend is a problem faced by neural networks when we go for backpropagation . It has a huge effect and the weight update process is widely affected and the model became useless. So, we used LSTM which has a hidden state and a memory cell with three gates that are forgotten, read, and input gate.

## About Project

Predicting the next word is a neural application that uses Recurrent neural networks. Since basic recurrent neural networks have a lot of flows we go for LSTM. Here we can make sure of having longer memory of what words are important.

# Data

The dataset we have used is taken from internet.

Title: Pride and prejudice

Author: Jane Austen

Release Date: November 12, 2022

Language: English

## Data preparation

* Converting string to sequence.
* Removing all the unnecessary characters and spaces.
* Converting in list and array of all the words and then again to the string for prediction.

# Approach

Main Steps followed in this project:

### Import the required libraries and Read the dataset –

### We use TensorFlow with Keras for our model building. We can import the LSTM model from Keras and use it. We can check the length of the corpus by using the len function on text after reading

### Using tokenizers and Getting unique words –

### The tokenizers are required so that we can split into each word and store them. We get all the unique words and we require a dictionary with each word in the data within the list of unique.

### Feature Engineering—

### Feature engineering will make the words into numerical representation so that it is easy to process them.

### Storing features and labels—

### X will be used to get the features and Y to get the labels associated with them.

### Building our model—

### We can see that we have built an LSTM model and used a softmax activation at the end to get few specific words predicted by the model.

### Model training—

### The model training uses Adam as the optimizer and uses categorical cross-entropy for loss function

### Saving model—

### The model is saved using the save function and loaded.

### Testing next word—

### These functions will help us to predict the next few words when we provide a sentence.

### Predict the next word

# Results

Describe your final data analysis results and everything that you discovered on the way the put you forward and you think is worth describing here. Elaborate on the significance of your findings/results and convey your thoughts.

# Discussion

## References and Challenges

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

<https://www.geeksforgeeks.org/intuition-of-adam-optimizer/>

challenges: main challenge we faced in training the data as the file is quite large so it takes so much time to run so to solve this problem we have used runtime as GPU.

Also the data we got is in text format so it contains lots of special characters and spaces so we have to remove them from our data.

# Conclusion

So as we have seen that on giving the input of 3 words we have predicted the 4th word and doing the same with both the models we found that the results of LSTM model is more precise than Simple RNN.

**Why LSTM is better than RNN?**

LSTM allows the user to train models using sequences with several hundreds of time steps(when back propagating from cell to cell), where as RNN struggles to do.

LSTM is capable of dealing with more complex problems than the RNN by keeping a constant flow of error throughout the back propagation from cell to cell.

It difficult to train RNN that requires long-term memorization.

Similarity between RNN and LSTM: It mantains the input and output configurations of one-to-one, many-to-one, one-to-many and many-to many.