**Text Generation Using LSTM**

**Major Project Report**

**Submitted By**

**(BATCH NO: CSE\_04)**

**SAI CHANDANA PALA (18UK1A0509)**

**KANCHA DINESH (18UK1A0572)**

**BOINAPALLY DEVIKA (18UK1A0508)**

**PUNNAM NAGAVARDHAN REDDY (18UK1A0548)**

**INDEX**

**TITLE PAGE NO**

**CHAPTER 1: INTRODUCTION 3**

* 1. OVERVIEW 3
  2. PURPOSE 3

**CHAPTER 2: LITERATURE SURVEY** **3**

* 1. EXISTING PROBLEM 3
  2. PROPOSED SOLUTION 3

**CHAPTER 3: THEORTICAL ANALYSIS**   **4-5**

3.1 BLOCK DIAGRAM 4

3.2 HARDWARE/SOFTWARE DESIGNING 5

**CHAPTER 4: EXPERIMENTAL INVESTIGATIONS**  **6-7**

**CHAPTER 5: FLOW CHART 8-9**

**CHAPTER 6: RESULTS**  **10-11**

**CHAPTER 7: ADVANTAGES AND DISADVANTAGES**  **12**

**CHAPTER 8: APLLICATIONS**  **12-13**

**CHAPTER 9: CONCLUSION**  **13**

**CHAPTER 10: FUTURE SCOPE**  **14**

**CHAPTER 1**

**1. INTRODUCTION**

**1.1. OVERVIEW**

Text Generation is a type of Language Modeling problem. Language Modeling is the core problem for a number of natural language processing tasks such as speech to text, conversational system, and text summarization. A trained language model learns the likelihood of occurrence of a word based on the previous sequence of words used in the text. Language models can be operated at character level, n-gram level, sentence level or even paragraph level. In this project, we are creating a language model for generating natural language text by implementing and training state-of-the-art Recurrent Neural Network.

**1.2. PURPOSE**

Text Generation is a task in Natural Language Processing (NLP) in which text is generated with some constraints such as initial characters or initial words. We come across this task in our day-to-day applications such as character/word/sentence predictions while typing texts in Gmail, Google Docs, Smartphone keyboard, and chatbot.

**CHAPTER 2**

**2. LITERATURE SURVEY**

**2.1. EXISTING PROBLEM**

Estimation and generation of sequence of text by analysing the text or input text.

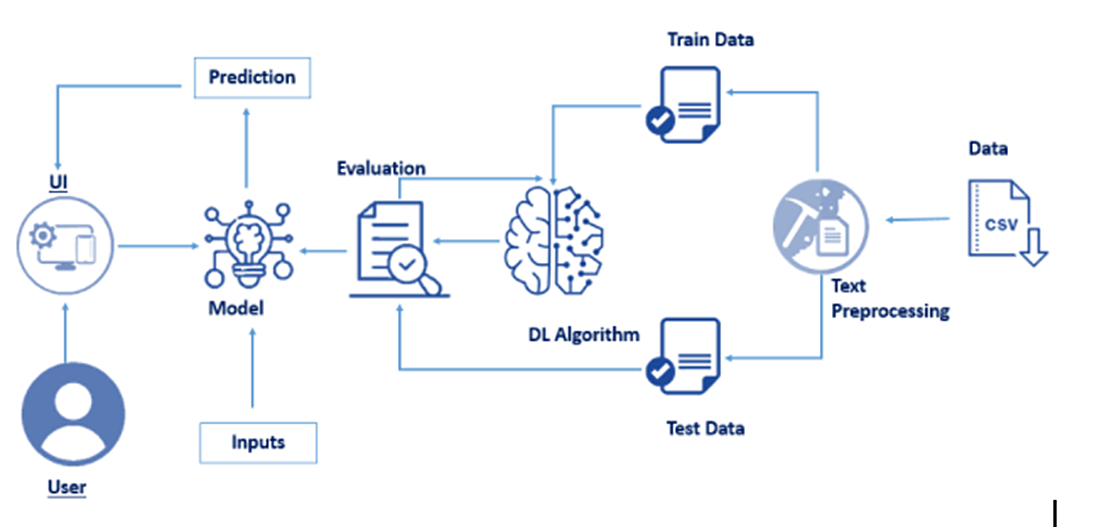
**2.2. PROPOSED SOLUTION**

Building a Web App which automatically generates the sequence of text by analysis the input text. Our model uses sliding window for analyzing large sequence of input text.

**CHAPTER 3**

**3. THEORTICAL ANALYSIS**

**3.1. BLOCK DIAGRAM**



**3.2. HARDWARE/SOFTWARE DESIGNING**

|  |  |
| --- | --- |
| **REQUIREMENT** | **SPECIFICATION** |
| Anaconda Navigator | You must have anaconda installed in your device prior to begin. |
| Spyder, Jupyter Notebook, Flask  Frame work | 1. One should have Spyder and Jupyter notebook. 2. One should install flask framework through anaconda prompt for running their web application 3. We need to build the model using jupyter notebook with all the imported packages. |
| Web browser | For all Web browsers, the following must be enabled:   1. cookies 2. JavaScript |

**Hardware Specifications:**

|  |  |
| --- | --- |
| **REQUIREMENT** | **SPECIFICATIONS** |
| Operating system | Microsoft Windows  UNIX  Linux® |
| Processing | Minimum: 4 CPU cores for one user. For each deployment, a sizing exercise is highly recommended. |
| RAM | Minimum 8 GB. |
| Operating system specifications | File descriptor limit set to 8192 on UNIX and Linux |
| Disk space | A minimum of 7 GB of free space is required to install the software. |

**CHAPTER 4**

**EXPERIMENTAL INVESTIGATIONS**

**Dataset preprocessing**:

Text Pre-processing includes the following main tasks

* Import the Libraries.
* importing the dataset
* Clean the data
* Create a sliding window

**Data set collection**:

1. kaggle.com
2. data.gov
3. UCI machine learning repository.

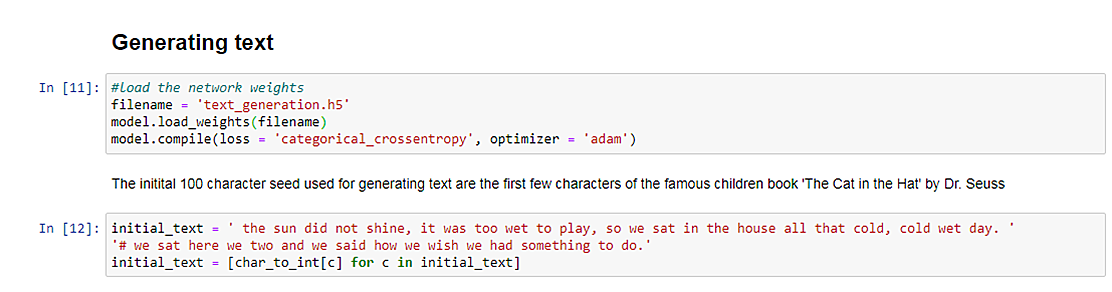
**Model Building:**

Model Building Includes:

* Import the model building Libraries
* Initializing the model
* Adding LSTM Layers
* Adding Output Layer
* Configure the Learning Process
* Training the model
* Model Evaluation
* Save the Model
* Test the Model

**Prediction of Output:**

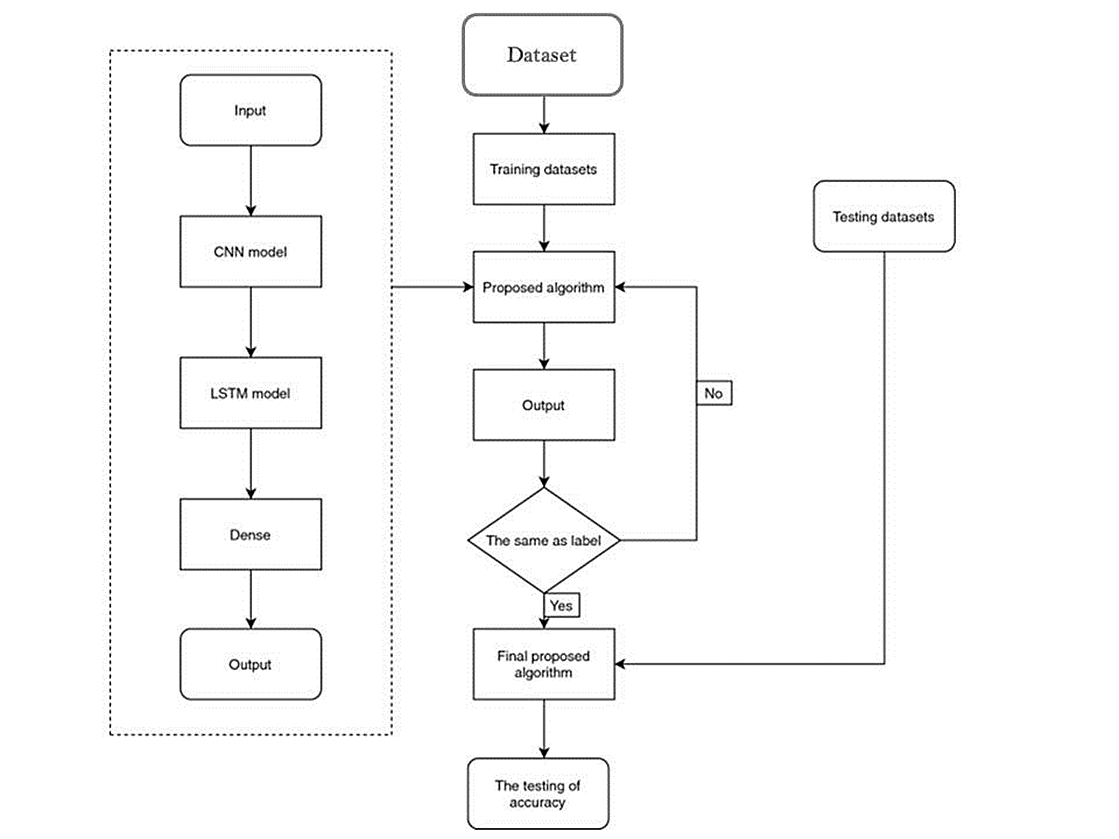
Following code is used for prediction of Next sequence of an given input Text.





**CHAPTER 5**

**Flow chart**



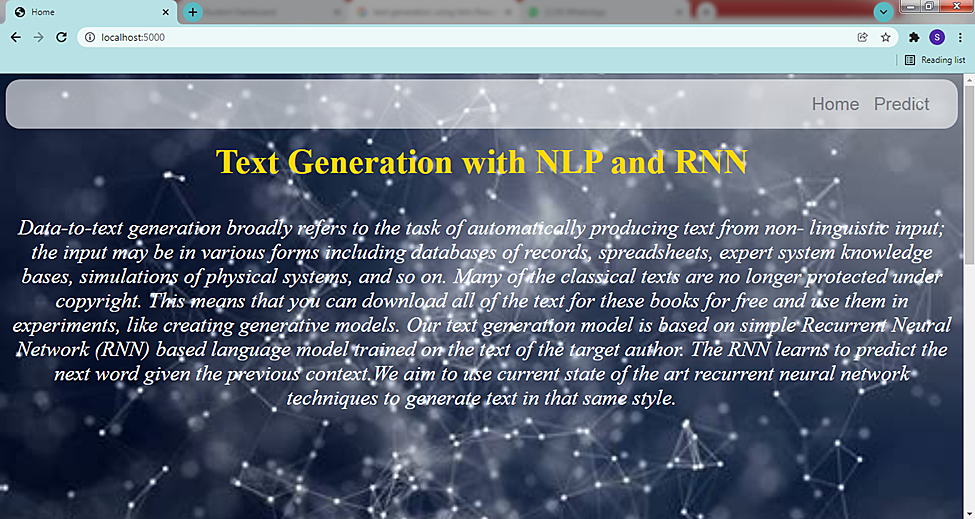
`

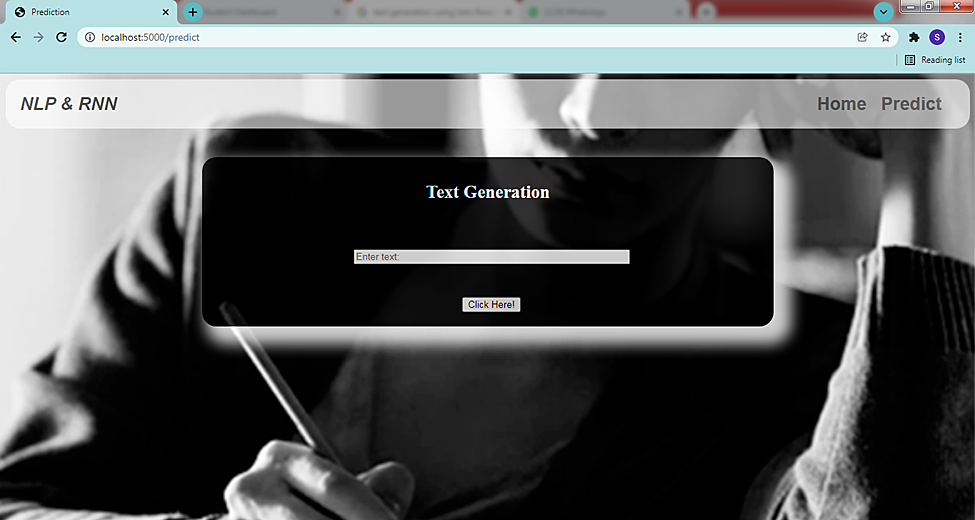
**CHAPTER 6**

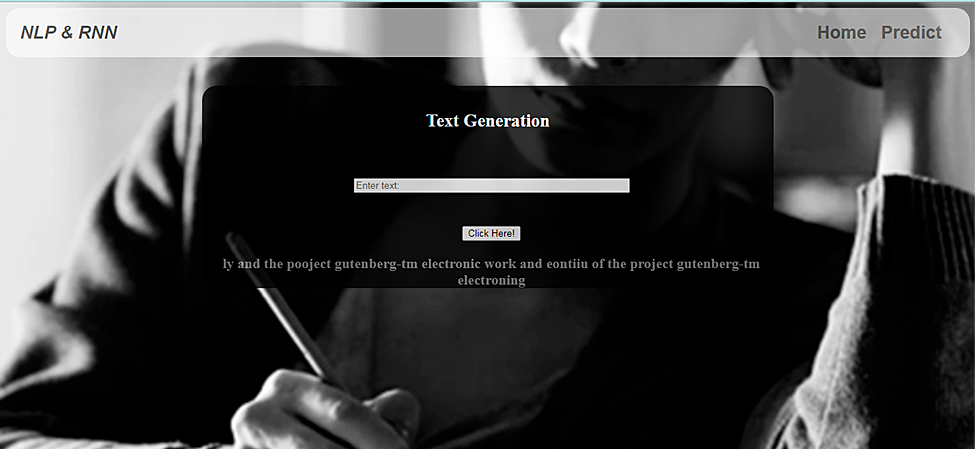
**RESULTS**

**Final output of the project:**

The home page is be shown as:







**CHAPTER 7**

**ADVANTAGES AND DISADVANTAGES**

LSTMs provide us with a large range of parameters such as learning rates, and input and output biases. Hence, no need for fine adjustments. The complexity to update each weight is reduced to O(1) with LSTMs, similar to that of Back Propagation Through Time (BPTT), which is an advantage.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behaviour required in complex problem domains like machine translation, speech recognition, and more.

**You are right that LSTMs work very well for some problems, but some of the drawbacks are:**

1. LSTMs take longer to train.
2. LSTMs require more memory to train.
3. LSTMs are easy to overfit.
4. Dropout is much harder to implement in LSTMs.
5. LSTMs are sensitive to different random weight initializations.

**CHAPTER 8**

**APPLICATIONS**

## Applications of RNN’s

* Text Generation
* Machine Translation
* Visual Search, Face detection, OCR
* Speech recognition
* Semantic Search
* Sentiment Analysis
* Anomaly Detection
* Stock Price Forecasting

**Applications of LSTM include:**

* Robot control
* Time series prediction
* Speech recognition
* Rhythm learning
* Music composition
* Grammar learning
* Handwriting recognition

**CHAPTER 9**

**CONCLUSION**

**From these entire findings we know fundamental concepts and can work on IBM Watson and machine learning.**

LSTM networks have proved to be the best type of model existing till date to per-form prediction and classiﬁcation over text-based data. LSTM is successfully able to resolve the problem faced by the standard recurrent neural networks, i.e., the problem of vanishing gradient.

**CHAPTER 10**

**FUTURE SCOPE**

**Enhancements that can be made in the future:**

LSTM is an efﬁcient model but it is overall computationally expensive and requires high processing power, i.e., use of GPUs to ﬁt and train the model. They are currently used in various applications like voice assistants, smart virtual keyboards and automated chatbots, sentiment analysis, etc. As future research, the model accuracy of current LSTM models can be surpassed by appending more layers and nodes to the network and applying the notion of transfer learning on the same problem domain.

**CHAPTER 11**

**BIBILOGRAPHY**

<https://analyticsindiamag.com/beginners-guide-to-text-generation-with-rnns/>

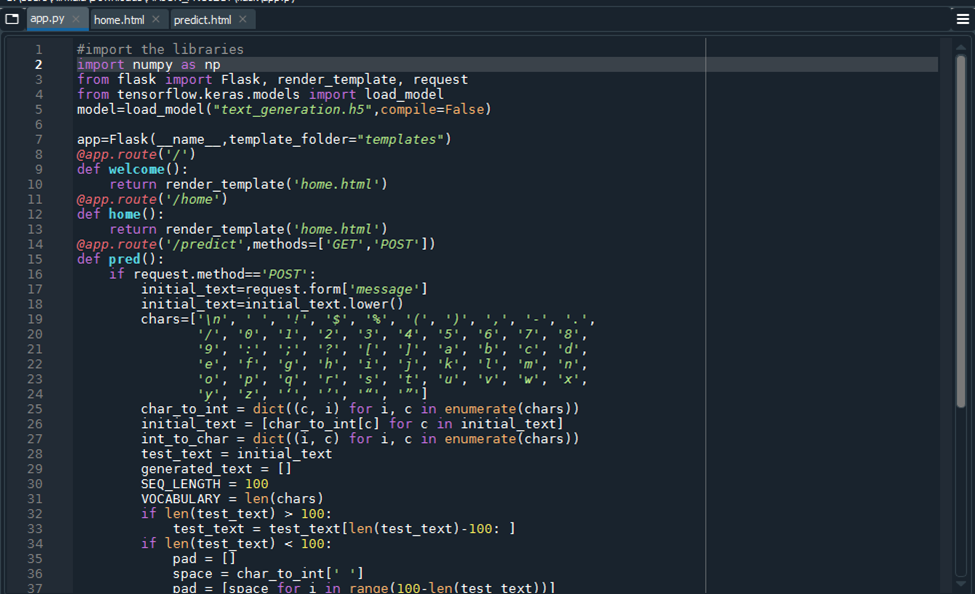
<https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/>

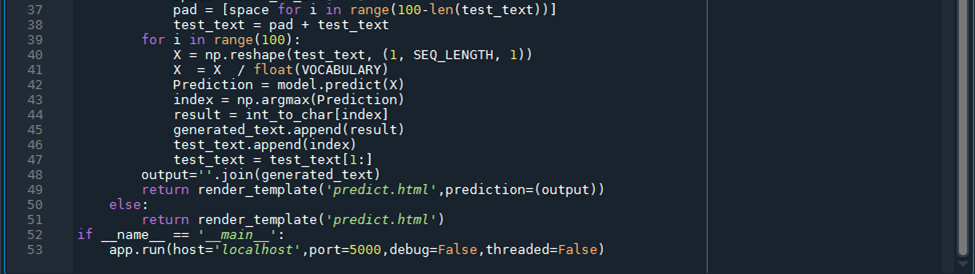
<https://iq.opengenus.org/text-generation-lstm/>

<https://en.wikipedia.org/wiki/Long_short-term_memory>

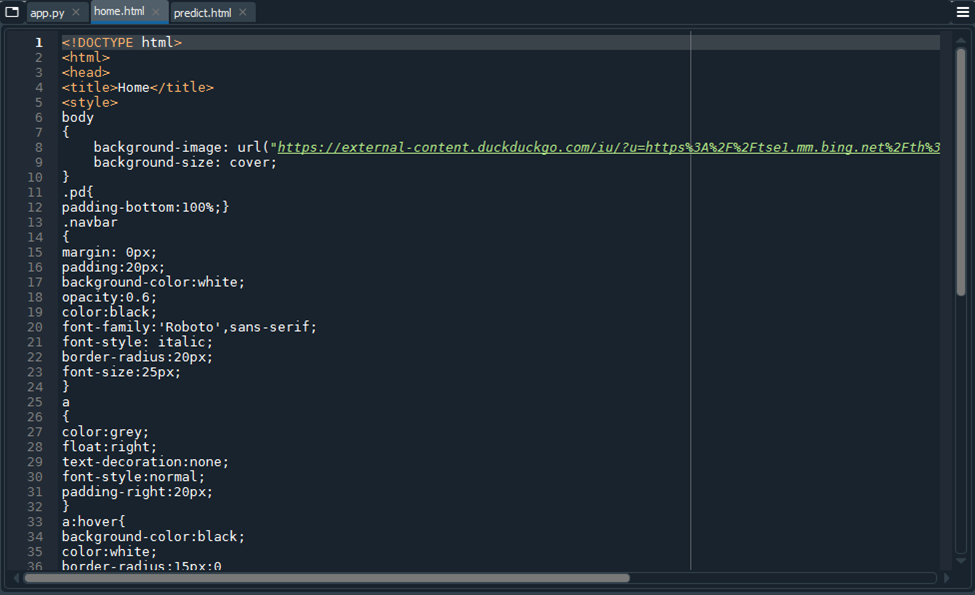
**APPENDIX**

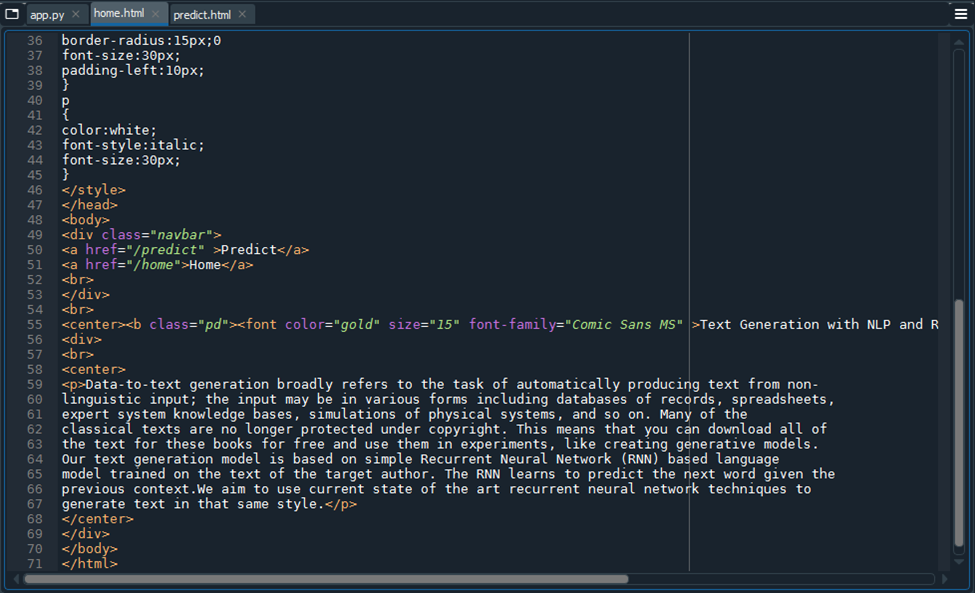
**App.py:**





**home.html:**





**predict.html:**

