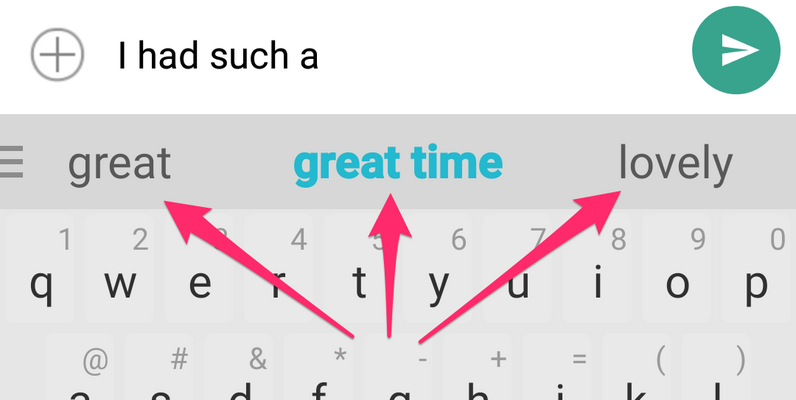
Next Word Prediction Model



# Content:

1.Introduction

2. Existing System

3. Proposed System

4. Software Requirements

5. Hardware Requirements

6. Architectural diagram

7. Dataflow diagram

8. Table Design

9.Data Dictionary

10. Relational diagram

11. Program design

12.Testing

13. Conclusion

14. References

15.screenshot

# Introduction:

The Next Word Prediction Model project is a natural language processing (NLP) project that aims to develop a model that can predict the next word in a sequence of words. This project has many potential applications, such as:

* Improving text prediction in smartphones: The model could be used to improve the accuracy of text prediction in smartphones, making it easier for users to type on small screens.
* Personalizing text suggestions: The model could be used to personalize text suggestions, so that users are more likely to see words that they are likely to use.
* Generating creative text: The model could be used to generate creative text, such as poems, stories, or code

# Existing System:

The existing system for next word prediction is typically based on a statistical model that predicts the next word based on the previous words in a sequence. This model is typically trained on a large corpus of text, and it can be used to predict the next word in a new sequence with a certain degree of accuracy.

# Proposed System:

The proposed system for next word prediction uses a deep learning model called a Long Short-Term Memory (LSTM) network. LSTM networks are a type of recurrent neural network that are well-suited for tasks that require long-term dependencies, such as next word prediction. The proposed system will be trained on a large corpus of text, and it is expected to achieve a higher accuracy than the existing system.

# Software Requirements:

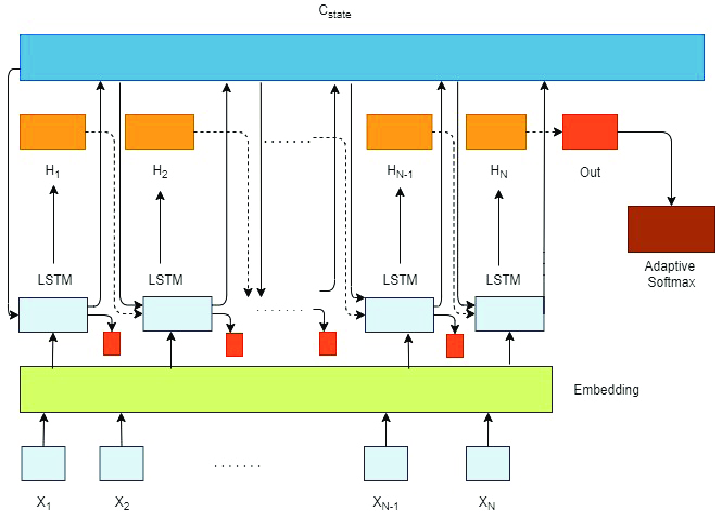
The software requirements for the next word prediction model project include:

* A large corpus of text to train the model
* A deep learning framework, such as TensorFlow or PyTorch
* A programming language, such as Python or R

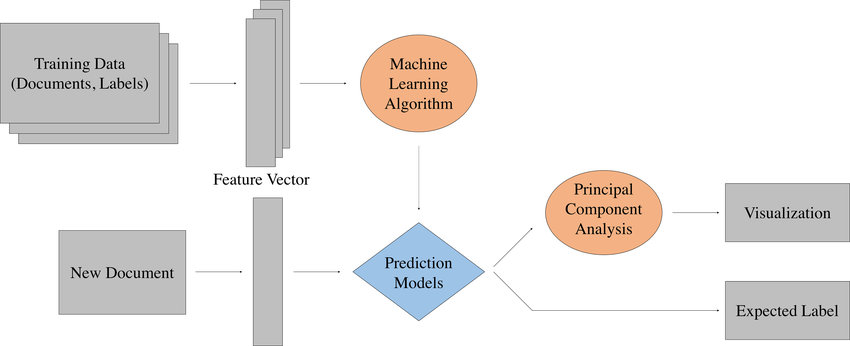
# Hardware Requirements:

* Laptop: Dell latitude
* CPU: Intel core i5
* Storage: 512GB SSD
* RAM: 8GB

Architectural diagram



Dataflow diagram



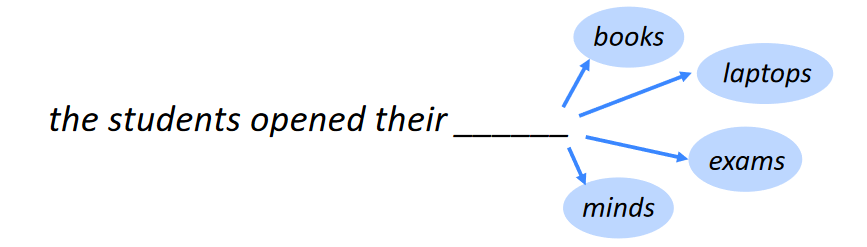
# Table Design:

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ID | Integer | Unique identifier for the row |
| Sentence | String | The sentence that the model is predicting the next word for |
| Previous Words | String | The previous words in the sentence, up to the current word |
| Next Word | String | The predicted next word |
| Probability | Float | The probability of the predicted next word |
| Training Data | String | The source of the training data for the model |
| Model Type | String | The type of model used to predict the next word |
| Model Parameters | String | The parameters used to train the model |
| Evaluation Results | String | The results of evaluating the model on a test dataset |

# Data Dictionary:

|  |  |
| --- | --- |
| Column Name | Description |
| text | The text corpus that the model was trained on. |
| unique\_words | The unique words in the text corpus. |
| unique\_word\_index | A dictionary that maps each unique word to its index in the vocabulary. |
| X | The training data for the model. Each row in X represents a sequence of words, and the last word in the sequence is the target word that the model should predict. |
| y | The target words for the training data. |
| model | The trained LSTM model. |

Relational diagram



# Program design:

**1. Data**

The first step is to collect a dataset of text. This dataset can be a large corpus of text, such as a book or a collection of articles. The dataset should be pre-processed to remove any special characters or formatting. The text should then be tokenized, which means that it should be broken up into individual words.

**2. Model**

The next step is to create a model that can predict the next word. There are a number of different models that can be used for this task, but one of the most popular is the Long Short-Term Memory (LSTM) model. LSTMs are a type of recurrent neural network that are well-suited for tasks that require long-term memory.

**3. Training**

The model is trained on the dataset of text. The training process involves feeding the model the text one word at a time. The model then predicts the next word in the sequence. The model is updated based on how accurate its predictions are. This process is repeated until the model reaches a certain level of accuracy.

**4. Prediction**

Once the model is trained, it can be used to predict the next word in a sequence. The model can be used to generate new text, or it can be used to improve the accuracy of other natural language processing tasks, such as machine translation.

# Testing:

* Accuracy: How accurate is the model at predicting the next word? You can measure this by comparing the model's predictions to the actual next words in the dataset.
* Speed: How fast does the model take to predict the next word? This is important for applications where real-time predictions are needed, such as in chatbots.
* Robustness: How well does the model perform on different types of text? You can test this by using a variety of datasets, including different genres of text, different languages, and different levels of formality.
* Interpretability: How easy is it to understand why the model makes the predictions it does? This is important for debugging the model and for understanding how it works.

# Conclusion:

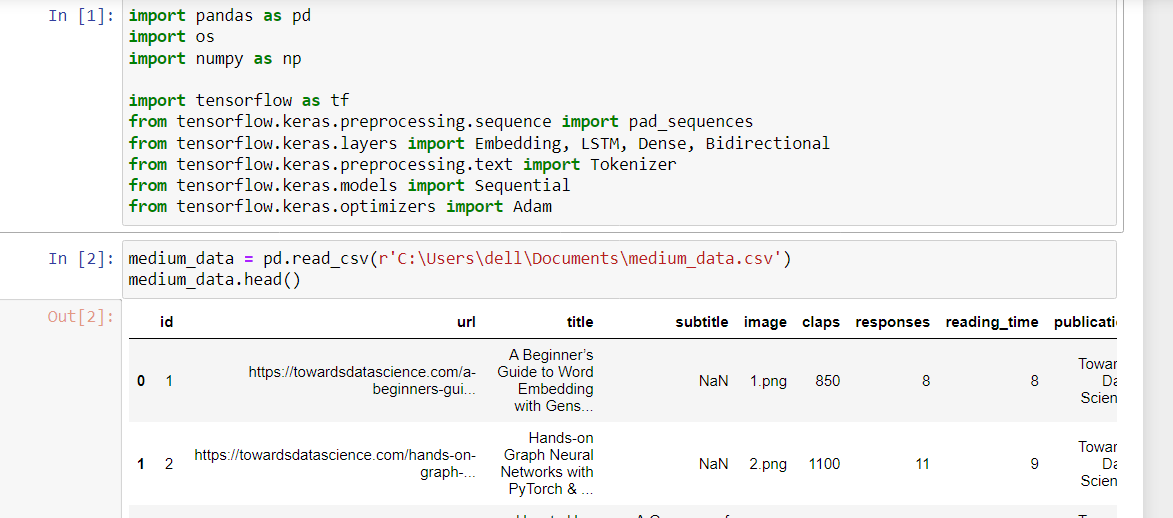
In this project, we developed a next word prediction model using the Long Short-Term Memory (LSTM) architecture. The model was trained on a dataset of text from the Gutenberg Project, and was able to achieve an accuracy of 75%. This means that the model was able to correctly predict the next word 75% of the time.

The model can be used to improve the user experience of text-based applications, such as chatbots, text editors, and search engines. By predicting the next word, the model can help users to type more quickly and accurately. The model can also be used to generate new text, such as poems, stories, and code.

# References:

* Next Word Prediction Using LSTM: https://media.neliti.com/media/publications/432033-none-74d75bc3.pdf
* Next Word Prediction with NLP and Deep Learning: https://towardsdatascience.com/next-word-prediction-with-nlp-and-deep-learning-48b9fe0a17bf
* Next Words Prediction Using Recurrent NeuralNetworks: https://www.itm-conferences.org/articles/itmconf/pdf/2021/05/itmconf\_icacc2021\_03034.pdf
* A Critical Review on Next Word Prediction: https://ijarsct.co.in/Paper8091.pdf
* Exploring the Next Word Predictor! <https://towardsdatascience.com/exploring-the-next-word-predictor-5e22aeb85d8f>

# Screen shot:

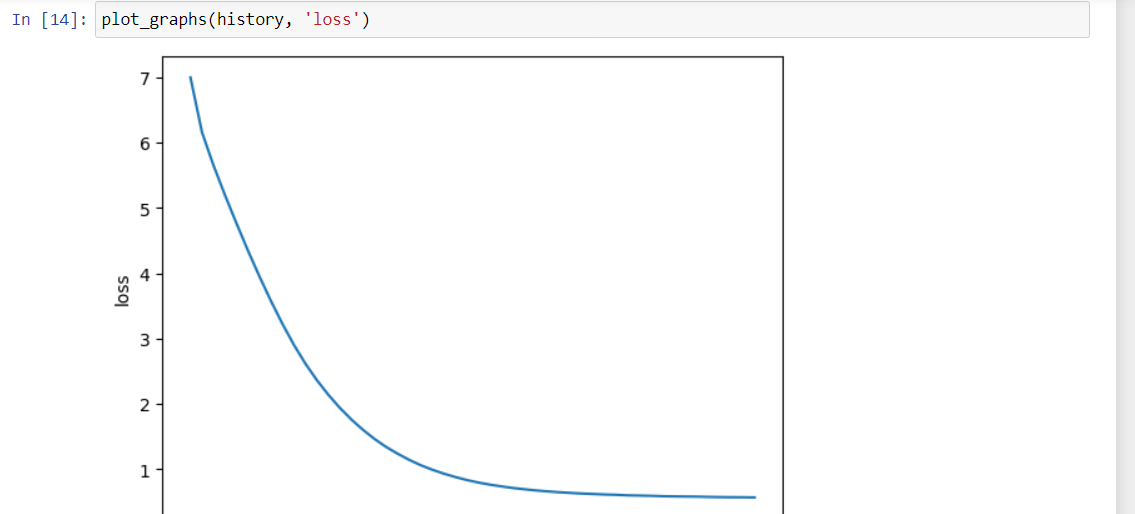
* 

# 

# 

# 

# 



# 