NEXT WORD PREDICTOR

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology
in

Computer Science and Engineering/Information Technology

By

Sukhpreet Singh 181401
Puru 181294

Under the supervision of

Dr. Pradeep Kumar Gupta

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I therefore refer to the work presented in this report entitled "**Next Word Predictor"** in fulfilling part of the **Bachelor of Technology** degree requirements **in Computer Science and Engineering / Information Technology** submitted to the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is a true record of my work done during the period from August 2021 to December 2021 under the direction of (**Dr. Pradeep Kumar Gupta**) (Associate Professor, Computer science & engineering and Information Technology).
The content of the report has not yet been submitted for the recognition of any other degree or diploma.

Puru 181294
Sukhpreet Singh 181401

This is to confirm that the above statement made by the candidate is true to the best of my knowledge.

Acknowledgement

I feel very happy to present this “Next word Prediction (Predictor)” to all of youl earn this or apply it in any way or continue to create better changes. I am very happy too Thanks to the people who made me eligible to complete / undertake a project like this and I will give this credit to my teachers who teach computer science, so that I can understanding my work. My special thanks and especially to my project manager and mentor to whom the project could not have been completed without him. My next thanks to the Project Partner for helping me complete this project. My thanks are to the Websites / Books and the people who have these Websites, from whom I have received help for their work.

Thanks and Regards

Sukhpreet Singh 181401

Puru 181294

Table of content

List of Abbreviations

1. LSTM : Long Short Term Memory
2. RNN : Recurrent Neural Networks
3. LoW : List of Words
4. NLP : Natural Language processing
5. CNN: Convolutional neural networks

List of Figures

List of Graphs

List of Tables

Abstract

As we all do our work utilising a tool (computer, mobile phone, etc.) to share records with one another, we are all doing so. How great is it that our tool software suggests or anticipates the next phrase through itself? It will help us save time and effort. So that it'll be fantastic. Natural Language Processing, a widely used research topic in special programs, is used to create this software in computer science. If we wish to write a word, an offer word may appear to be the same as the one we need to write. This project is an example of how NLP can be used. We'll look at one variant that can be used to anticipate the next phrase. We'll be dealing with a model known as Long Short Term Memory. As previously stated, we use backpropagation, which may be a problem faced by neural networks. Vanishing gradient is what it's called. The model has grown obsolete and has had a significant impact on the weight replacement process. There are a few inputs that could come from an earlier time or a subsequent stage. We also have an overview gate that regulates the burden so that it can accurately determine what information has to be removed before proceeding to the next gate.

The information that will be provided to the mobile and output gates must be trusted by the input gate. The LSTM structure and artwork are visible. RNN is used to predict the next word in a neural software component. We cross for LSTM since basic RNN has a range of flows. With the help of gates that we've made evident in advance, we may ensure that we have a longer recollection of what words are important. Language Modeling is another name for Next Word Prediction. It is the task of foreseeing what sentence will appear next. It is one of NLP's most essential projects, with several applications. Making a model utilising Nietzsche's default textual content record as a good technique to predict a client's phrase once the client has written forty letters, the model will recognize 40 letters and anticipate coming close to the top 10 words using Nietzsche's default textual content record. Expect to get close to the top 10 terms using RNN neural networks, which will be finished with Tensorflow.

Our goal in developing this model was to produce 10 or more words in the shortest amount of time possible. RNN recognizes beyond text and predicts words that may be useful for the customer to border sentences, and this approach employs letter to letter prediction, which predicts a letter after letter to form a phrase.

Chapter-1 INTRODUCTION

1.1Introduction

The stage will contain precise performance in order to cover the next phrase prediction venture name. Through the model, the next feasible phrase might be predicted. Natural language processing, language modelling, and deep learning approaches could be applied. We will begin by analyzing the facts, which will be followed by the pre-processing of the statistics .After that, tokenization of records is performed, and finally, the deep learning model is created using LSTM .How does your phone's keypad recognise what you want to enter next? Language prediction is a Natural Language Processing - NLP tool that predicts the text in the text before it. Popular methods of language prediction include auto-complete and suggested responses.

The selection of a language model is the initial stage in language prediction. This article demonstrates many ways for developing the Next Word Predictor seen in apps such as Whatsapp or any other messaging app.You can use one of two models to create a Next Word Suggester/Predictor: 2) Long Short Term Memory or 1) N-grams model (LSTM). We'll look over each model and determine which is the best.If you choose the n-grams route, you'll need to concentrate on the 'Markov Chains,' which use the training corpus to estimate the likelihood of each subsequent word or character. This approach's code snippet is shown below. In this method, the length of one sequence is used to anticipate the next word. This indicates we'll use the preceding word to forecast the next word.Long Short Term Memory (LSTM) technique: Using a neural language model, a more sophisticated way is to use Long Short Term Memory (LSTM) (LSTM). The LSTM model combines Deep Learning with a network of artificial "cells" to manage memory, making it more suitable for text prediction than classic neural networks and other models.Synthetic intelligence includes a significant amount of natural processing learning. It is a fantastic field of study that is widely utilised in extraordinary packages. We frequently text one another and have discovered that every time we type a word, the next phrase appears. This is one of the services that NPL provides.

We've made significant progress and will employ the recurrent neural community in this fashion.LST recognises beyond textual content or seek within a dataset and assists us in finding text in the future.This is a crucial LSTM application with numerous packages.Predicting the following word is a difficult method. SVM, decision tree, and other algorithms are examples. Don't promise high results and take longer to deliver the final product. The task is pretty difficult. To make such predictions, the text editor must be extremely intelligent and have a thorough comprehension of the data set. In addition to authoring essays and framing long sentences, the mission next phrase prediction makes

framing the artwork simple.The venture title subsequent word predictor predicts 40-50 percent of the words out of one hundred percent. Recently, software developers constructed an auto-following language predictor for Bengali, which was challenged, and it was determined that finding accurate accuracy using Rnn is difficult.When the distance between the context and theword to be predicted grows, standard RNNs and other language models become less accurate. Because it has memory cells to retain the previous context, LSTM is used to solve the long-term dependency problem.

More information on LSTM networks may be found here.Let's get started writing code and defining our LSTM model. First, an embedding layer is employed, followed by two stacked LSTM layers with 50 units each.Two fully connected or thick layers follow the two LSTM layers. The first dense layer contains 50 units, whereas the second dense layer is our output (softmax) 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.

1.2 Problem Statement

When there was no following sentence, a predictor was used to compose huge paragraphs and books (which had been on line).The typist's burden has now diminished. This application is also utilised in search engines and on YouTube. Previously, a lot of time was spent when sending SMS and emails, but now we can easily anticipate words with the help of the subsequent word predictor.This software is being used for download and pre-processing.Wikipedia is also an example of egalitarianism. Wikipedia's facts package includes sixteen million sentences.Previously, generating vector representation was a difficult task; now, with the help of various word predictors, it is possible to do it without difficulty.The fundamental goal of NLP is to replace the original phrase with a word vector that can be used to locate each word's meaning using phrase prediction software.Previously, a great deal of time was lost at some point in the version structure, but since the invention of this programme, it has become much easier to achieve this.Previously, a great deal of time was wasted during model education and output, but thanks to the advent of next word prediction, a great deal of time was saved.

1.3 Objectives

The project's purpose is to provide predictive tools similar to those created by shift Key, an app that makes it simple for people to type on their phones.

(1) Data is downloaded from blogs, Twitter, and the news,
(2) a quick explanation and beginning explanation of the statement is generated.
(3) The words whose frequency is in unigrams, biograms, and trigrams were created to make the data easier to interpret.
(4) They were used in producing word vector.

(5) They were also used in model architecture, model training, and output.
(6) They were also utilised in Google, emails, and YouTube.

Figure 1

1.3 Methodology

The steps in methodology are:-

1)Data pre-processing

2)Comunity of LSTM

DATA PRE PROCESSING

This painting depicts a versatile version that can also help users construct an indisputable model that can assist customers in detecting the next word. When creating the user.We would offer letters to the records set first and foremost, and then the data set would provide letters through which we should expect certain phrases. This sentence might be surpassed by the LSTM neural community and later it would expect the same the rating would then be transmitted via the same LSTM and later it would forecast the next word, as demonstrated in the above parent supplying input to the statistics. The neural network structure and our use of the Tensorflow library are shown below.The output layer with the same node as the input layer. First, introduce our helpful model. After several modules have been imported, Numpy pandas will import Nietzsche's default textual material. It is our information system.Step 2: The LSTM path is cell explicit, including a line that passes through the optimal point of frame.The transit line is

comparable to the cell kingdom.With only a slight direct touch, it turns instantly throughout the chain.It is straightforward for facts to unfold when they are unaltered.LSTM can delete or upload data to the telephone country, which is strictly monitored by entryways.Information can be bypassed via gates.They are the result of sigmoid neural network layers and a point repetition assignment. The sigmoid layer displays numbers somewhere within a clean and single list, indicating that the overall value must pass. The values of 0 way "do not permi anything," while the values of 1 way "permit it's all over!" For safety and to manipulate cell popularity, LSTM features those gates.

Tokenization is the process of breaking down huge text facts, essays, or corpora into smaller chunks. Smaller documents or traces of text can be used to create these smaller chunks.They can also serve as a word dictionary.The Keras Tokenizer allows us to vectorize a textual content corpus by turning each textual content element into a token.content into either a set of integers (each integer being the index of a token in a dictionary) or a list of stringsinto a vector with a binary coefficient for each token, based solely on phrase count,tf-idf is used. To learn more about the Tokenizer's magnificence and content, click here.Go here for information pre-processing using Keras.the texts will subsequently be converted into sequences.This is a method of converting textual content data to numbers so that we may perform more advanced studies on it. The education dataset will then be created. The 'X' will be made up of education statistics plus textual input..The outputs for the education records will be included in the 'y'. As a result, the 'y' includes all of the next phrase predictions for each 'X'.

The vocab size will be calculated using the period taken from the tokenizer.After that, submit 1 to Word index. We're including 1 since zero is allocated for padding, and we want to start counting from one.This will come in handy with our categorical crossentropy loss. The following is the rest of the code for tokenizing records, expanding the dataset, and converting the prediction set into express facts:It should be noted that pre-processing can be improved. You can experiment with various approaches to improve the pre-processing step, which may aid in achieving higher loss and accuracy in fewer epochs.

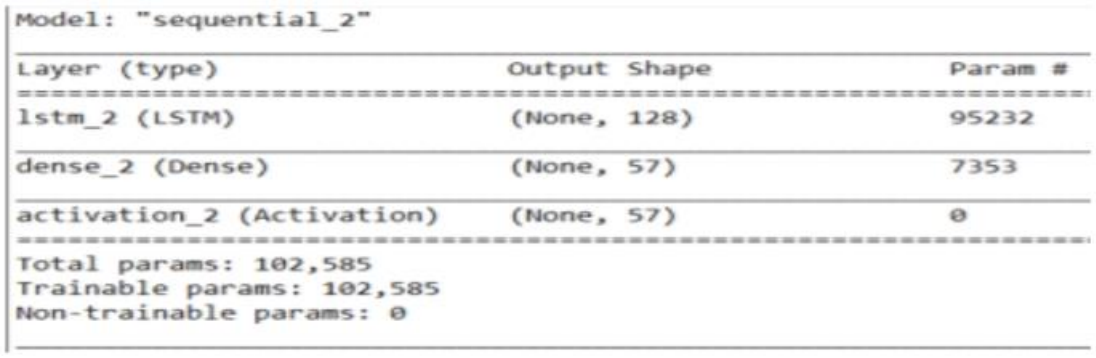


Figure 2

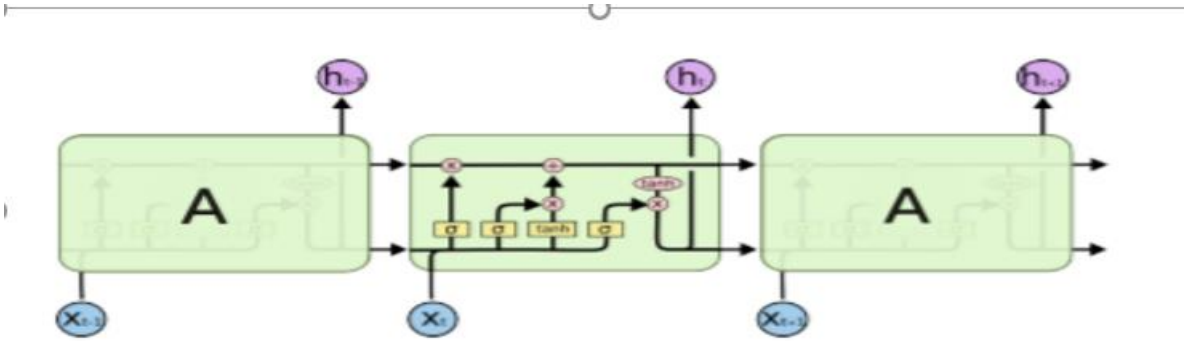


Figure 3

Building the Model: We can be building a sequential model. After that, we'll establish an embedding layer and define the input and output dimensions.The enter length must be set to 1 since the prediction will be made on only one word, and we will receive a response for that single phrase. Our structure will then be enhanced using an LSTM layer. We will provide 1,000 units and ensure that the sequences are returned in right order. This ensures that it can be passed via any other LSTM layer.

We'll also skip the next LSTM layer during every step.We don't need to define the go back sequence because it is far fake by default. This will be passed through a hidden layer with a thousand uses of the dense layer characteristic and relu as the activation. Finally, we bypass it using a softmax activation and an output layer with the appropriate vocab length. The softmax activation ensures that we receive a lot of chances for outputs with the same length as the vocab. The entire code for our model shape is shown here. We'll look at the version code, the version summary, and the version plot once we've looked at the version code.

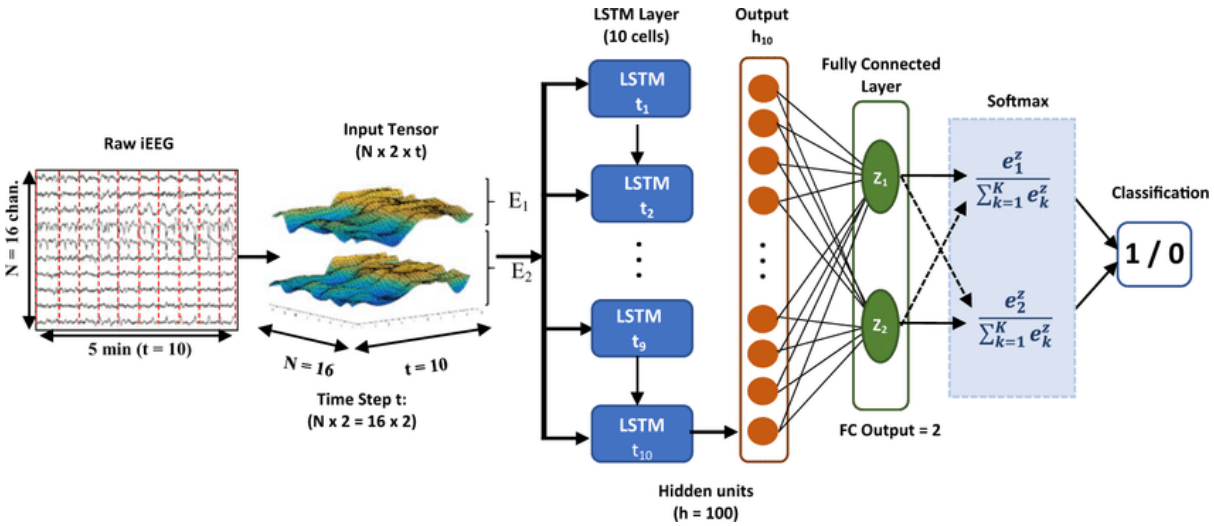


Figure 4

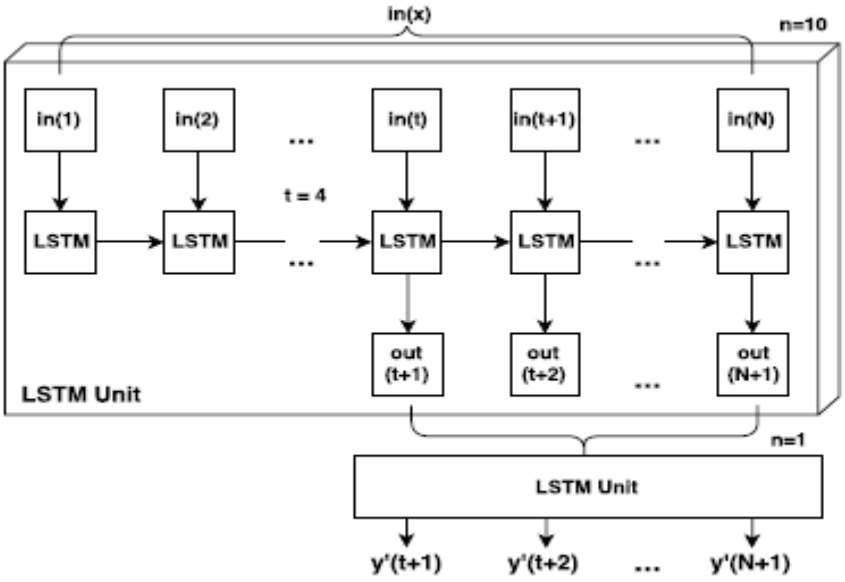


Figure 5

1.4 Organization

User: The user's goal is to enter forty words for The LSTM Neural Network, just like in the video. LSTM: LSTM is an expert on the default next phrase prediction textual content file; it will compute its weight and look for a top phrase letter by letter. 1. "It's tough enough, hey, I'm Sanket, what's up?" 2. "That which no longer harms us strengthens us.". "I'm no longer sad that you lied to me; I'm disillusioned because I can no longer have faith in you." " four. "People who were observed vibing were thought to be nuts by people who couldn't hear the tune.It's difficult enough to remember my judgments without having to recall my reasons for them. "It's not a lack of love, but a lack of friendship that's the problem."Input test examples • Phrase list: This will keep everything together.If the buyer wants a certain number of pinnacle works, it will be counted as number N.Phrases and return the term list to the individual. • Above define five.1 experimental end resultas seen in the n-grams approach is inferior than the LSTM method because LSTMshave the ability to inspect the putting from a distance, similar to the substance corpus.

When embarking on any other venture, remember to incorporate one of the most recent pre-organized designs by searching the internet for open-supply executions. As a result, you may no longer wish to start with no planning and no longer be concerned about the association cycle or hyperparameters. Attempting to build a model utilising the Nietzsche default text report in order to predict clients' sentences after they wrote forty letters; the model will recognise forty letters and anticipate future letter/terms using an LSTM neural network, which can be done with Tensorflow.

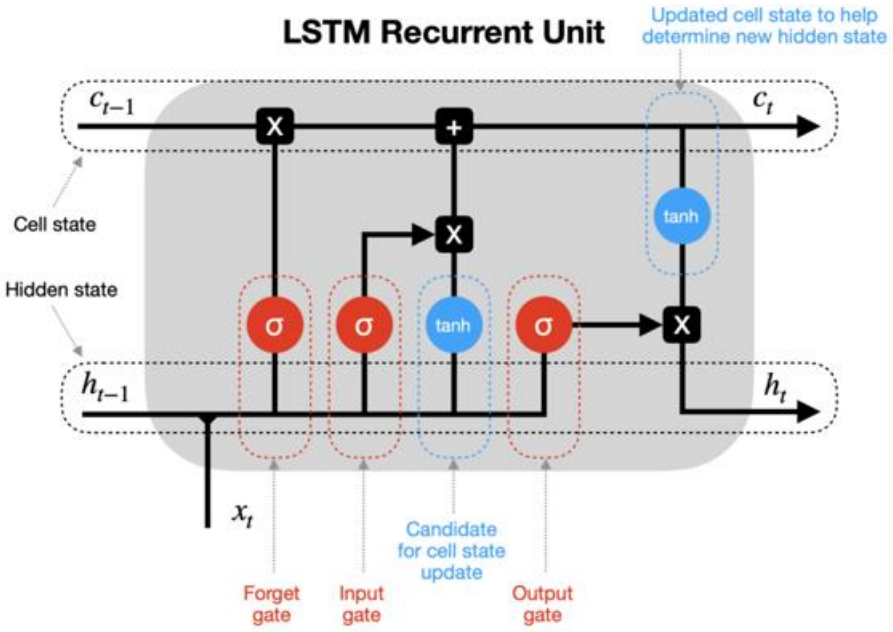
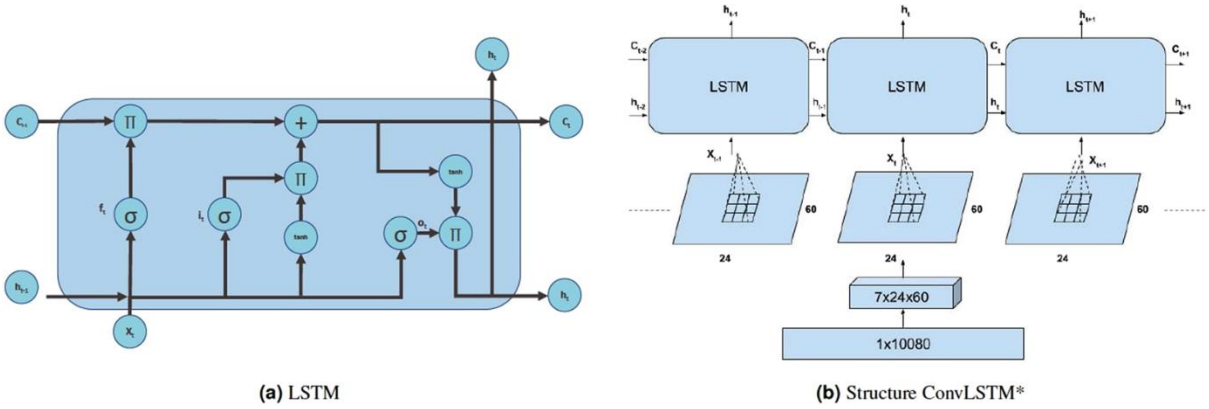


Figure 6

This product has more potential on social media for syntax and semantic evaluation in artificial intelligence herbal language processing. • We try to tweak several entry layers for exceptional enter layers, and we can find that regardless of the input layer size, the output prediction has an accuracy of 54% to 55%. • The schooling accuracy for 10 entry nodes is approximately 56%, whereas the sorting out accuracy is approximately 54%. • The education accuracy for 20 input nodes is around 56%, but the testing accuracy is around 80%.five-fifths of a percent.The training accuracy for 30 entry nodes is around 56%, whereas the testing accuracy is around 55%. • The schooling accuracy is roughly 56% for forty entrance nodes. However, the accuracy of checking out is at 54%. 9% The Neural network did an excellent job in forecasting, as you can see in the second case in " Fig 6 " by inputting six check cases and passing five in our params (i.E stronger, energy) According to "Fig 7," our solution predicted these input test periods with a spherical 56% accuracy in real-world scenarios.



(a) LSTM **(b)** Structure ConvLSTM*

Figure 7

Because of its ability to accomplish jobs quickly and easily, neural networks have exploded in popularity in the modern age of data science. There are several types of neural networks that we utilise to complete various jobs. In this post, we'll look at the LSTM model, which is one of the neural network's versions. We already addressed how LSTM networks perform better with sequential data, such as time series, in one of our previous posts. We'll use text data as the sequential data in this case and try to fit an LSTM model to it.

Chapter-2 LITERATURE SURVEY

This section explains how to conduct a standard literature review. This paper is for seeking to break free from the schooling challenge distribution.The multi-window convolution algorithm (MRNN) is released, along with a connected remnant minimum gated unit (MGU), which is a condensed version of the LSTM on this cnn tries to skip some layers while training, resulting in significantly less training time and higher accuracy. The use of multiple layers of neural networks can purpose prediction delays n range names.RNN allows code consumers to rely on the following syntax to finish the code hassle. In comparison to existing methodologies, the authors claim that their methodology could be highly correct.Inner-vocabulary names and identifier prediction are examples of word prediction in elements. To predict terms within words, they used the LSTM neural language model. At the objective forecast, the community network model is proposed. The Bangla language was worked on by the authors. They recommend a unique method of anticipating the loss of words and terms. They advise using an N-gram-based whole language model, which predicts a set of phrases. They have reaped remarkable benefits. The authors utilised LSTM to predict the following word in Assamese. Maintain note language in accordance with the International Phonetic Association (IPA) chart and provide an example.

Humans with physical limitations in Baka form. This model employs Unigram, Bigram, and Trigram approaches to forecast the following phrase, and the prediction score is converted into the received phrase, however the accuracy drops to 30-40%.Because of the absence of gradients and complexity, it was discovered that obtaining acceptable accuracy via RNN method is difficult due to the disappearance of gradients and complexity Continuous NN takes more time to teach and evaluate in this paper.They used to be anticipating the subsequent letter of the Indian alphabet (PNCH), which became more to correct the textual content and a little bit regarding forecasting the subsequent phrase, however it turned out to be absolutely correct to recognise.A method known as hit and miss is used, but the accuracy is low, and the model has stopped operating on this type of problem statement. This has been the most popular method of dealing with this type of problem; the paper explains LM and the confusion algorithm, which is the foundation of natural language processing; this allows us to generate 3-D input data for our model.The 1-degree sample characteristic function set of rules is utilised for extinction problem selection, but with as little accuracy as possible because the fundamental sentence became employed in teaching and comparing comparable statistics .

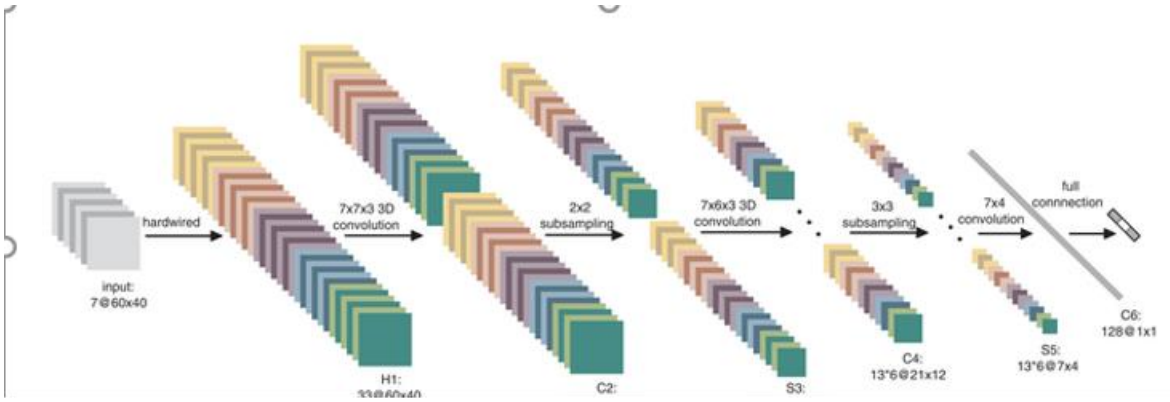


Figure 8

GPT will best do it as a waste of helpful resources for clean artwork because it is the largest and most expensive model of this type of characteristic as word prediction is a smooth project.As this set of rules does, the majority of the paper tries to version to expect the following text, the multiple papers are beneficial, the paper predicts the following code using SVM and RNN.When compared to other ML venture algorithms such as SVM, Decision Tree, and so on, the accuracy of phrases the client thinks predicts user mind. Because the work may be difficult, don't deliver excellent impacts and wait longer for outcomes.

Textual content prediction requires as much language knowledge as possible to produce accurate predictions. As a result, we must continue to educate the neuronal community with new languages and information.

3. Chapter-3 SYSTEM DEVELOPMENT

Statistical (at least one methodology from the aforementioned methodologies should be used for version improvement) It is necessary to embody some mathematical remedy or associated facts.This will cover the section where the next phrase prediction is developed and performed precisely.This version will remember the final word of a certain sentence and forecast the next possible word.We will remember natural language processing and language modelling techniques, as well as deep learning.We'll start by looking at the statistics that have been noticed utilising statistical preprocessing. We'll tokenize this data and, sooner or later, we'll be able to construct a deep learning model. LSTMs will be used to build the deep learning model.The text information records are/are quite easy to get, and we can recall Project Gutenberg, which is a volunteer effort to scan and archive cultural works in order to "promote the introduction .

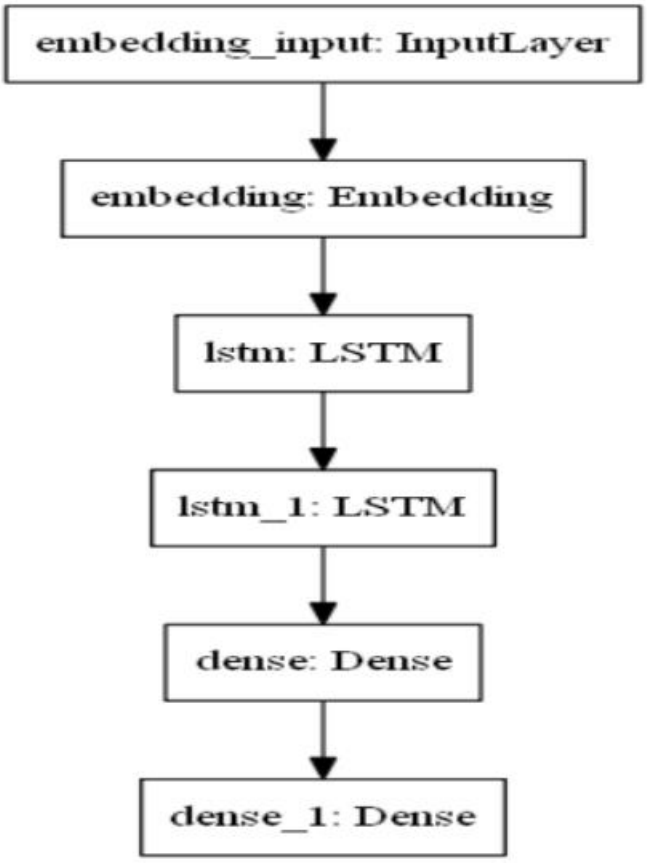


Figure 9

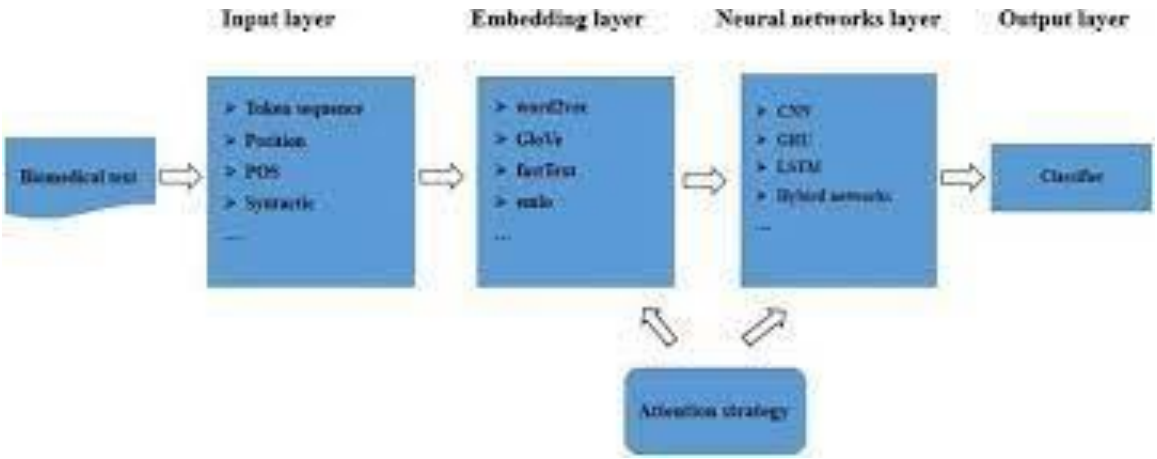


Figure 10

The first step is to remove all redundant information from the Metamorphosis database. We'll get rid of the database's start and stop points. This is a set of records that we don't need. The first line should be something like this.When Gregor Samsa awoke one morning from a nightmare, he discovered it. He starts offevolved to rise and straighten his small physique, which must be the database's last line.Keep when this phase is finished. We'll retrieve Metamorphosis clean.Txt item using the utf-eight encoding. The next stage in our cleanup procedure is to replace all new unnecessary strains, cart returns, and Unicode characters. Finally, we can be positive that we have just unique names.

)f text data.

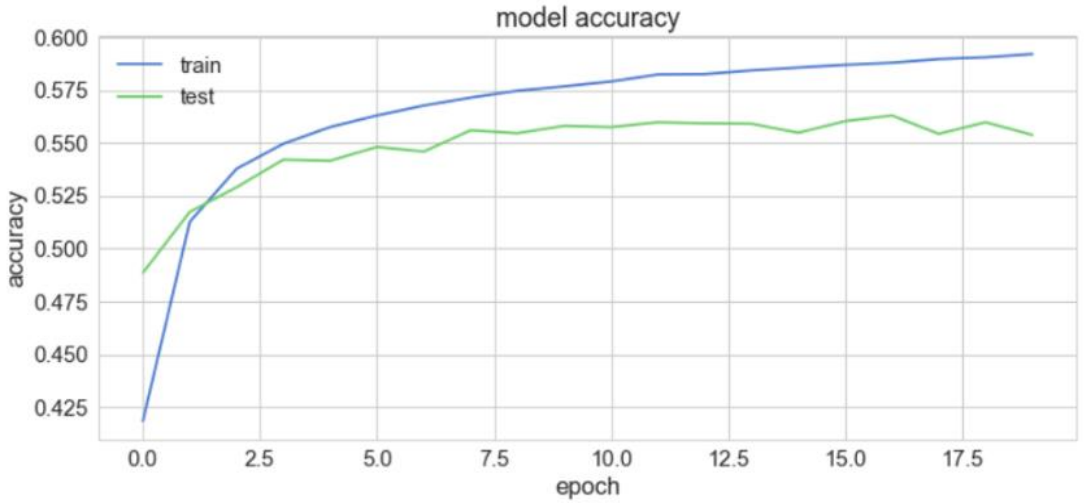


Figure 11

We will only consider each phrase once and will delete any additional duplicates. This will help the model teach more effectively and avoid similar confusion caused by phrase recurrence. The overall code for pre-processing text data is shown below.Tokenization is the process of splitting large text material, essays, or corpora into smaller chunks. These subdivisions can take the shape of tiny files or text data strains. It can also function as a word dictionary.Tokenizer

allows us to vectorize a chorus of text by transforming each text into a vector of consecutive numbers (each variation is a token index in the dictionary) or a vector in which the coefficient of each token can be binary, based on a large number of words (tf-idf). Visit here to learn more about the Tokenizer style and how Keras may be used to pre-process text records.The textual content will subsequently be turned into a collection. This is a method of converting text facts into numbers so that we can analyse them more thoroughly. We will develop a training programmer.

The 'X' will include educational records as well as textual content entry data. The 'y' will include the schooling's output data. As a result, for each incorporates all of the following phrase predictions.We'll use the length taken from tokenizer to determine vocab size.1 is added to it. Because 0 is about to end and we need to start counting from 1, add 1. Finally, we can turn our 'y' prediction knowledge into font size statistics. This function transforms an integer category vector into a binary elegance matrix. This could be beneficial for our losses, allowing you to be categorical crossentropy.The following is the rest of the code for creating information tokens, developing databases, and changing the guesses in category records:Note that pre-processing should be improved. You can try unique approaches to expand a pre-processing stage, which can help you achieve superior losses and accuracy in smaller epoch.

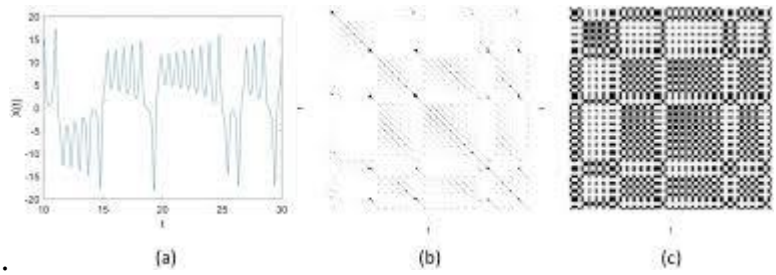


Figure 12

Model Creation:

We could be building a sequential model. After that, we'll make an embedded layer and specify the input and output sizes. It's crucial to set the duration of the enter to 1, because the prediction could be made with just one phrase, and we'll be able to acquire the answer for that word.Any other layer of the LSTM We can also move it to every other 1000 units in the succeeding layer of LSTM,but there is no need to specify the return collection because it is automatically faked. We'll get around this by using an encrypted 1,000-node layer with a dense layer feature and a com set as activation.

Finally, we send it to the output layer using the specified voice length and softmax activation. When softmax is enabled, we have a lot of options in outputs that are proportional to the size of the voice. Our version structure's whole code is demonstrated here.The three essential callbacks for our model's education will be uploaded. ModelCheckpoint, ReduceLROnPlateau, and Tensorboard are the three most important callbacks. Let's have a look at the functions of each of these character callbacks.ModelCheckpoint — This callback stores the weights of our version once it has been educated.By setting save best only=True, we only save the nice weights of our model. The loss metric will be used to track our education. ReduceLROnPlateau — This callback is used to reduce the optimizer's studying price after a specific number of epochs. The endurance has been set to three in this case. If the accuracy does not increase after three epochs, we will cut our mastering cost by employing an aspect of zero. 2. Loss is also employed as a monitoring metric in this case.Tensorboard — The tensorboard callback is used to visualise graphs, specifically graph plots for accuracy and loss. We'll start by searching on the internet.The following phrase prediction loss graph.We'll save the best styles to the report nextword1.H5 based entirely on the metric loss. This record will be necessary for gaining access to the anticipate function and looking for our next sentence. For the loss to improve, we'll look ahead three epochs. If it does not improve, we will cut the cost of studying. Finally, if desired, we will use the tensorboard function to visualise the graphs and histograms.

Chapter-4 PERFORMANCE ANALYSIS

System evaluation progressed in at least methods depending on the intensity of the usual These often used Analytical / Computational / Statistical / Experimental / or mathematical methods.Outcomes in distinct classes can be compared to extraordinary views Output in distinct categories.N-grams method is inferior to the LSTM method because LSTMs have the memory to examine the setting from the beginning while also paying tribute to the substance corpus. When starting any other business, you should pick one of the current pre-organized designs by searching the internet for open-source software.

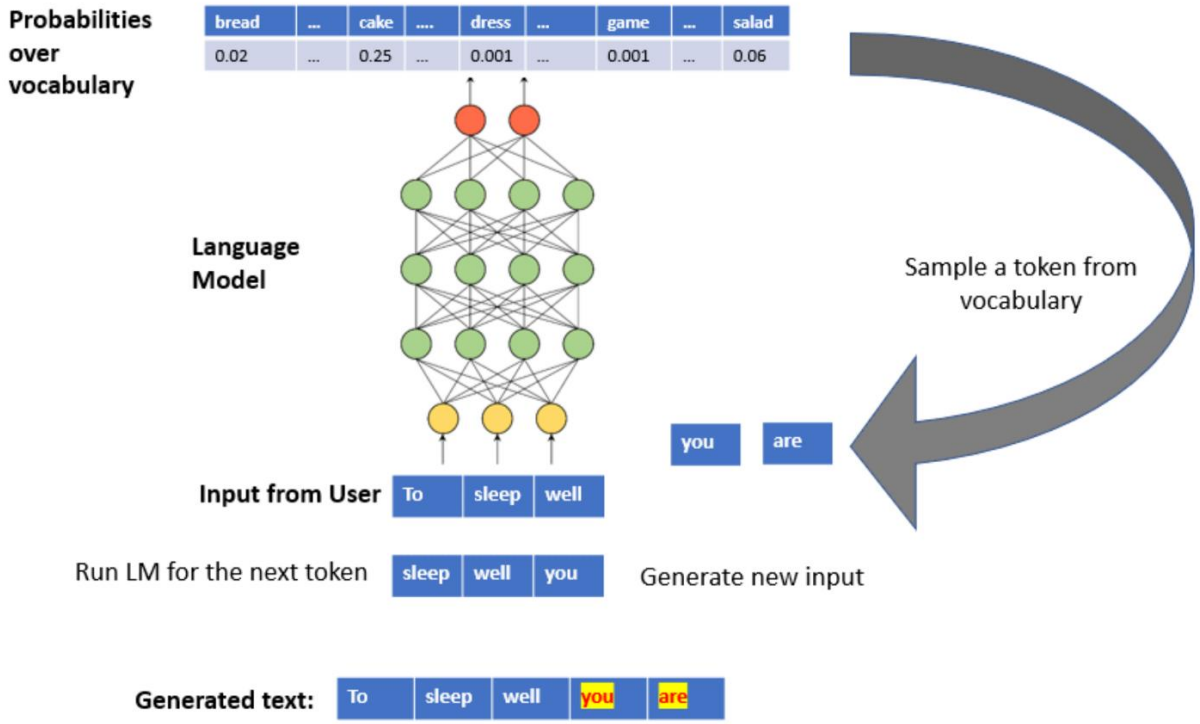


Figure 13

of text data.

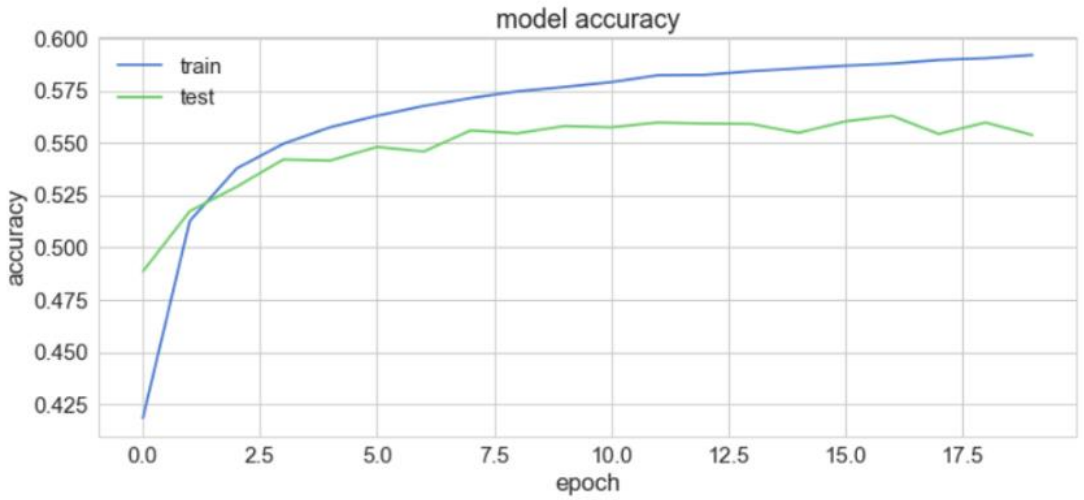


Figure 14

Table 1

Table 1. Structure of different models with it's highest average accuracy with its epoch for Assamese language.

Test number	Hidden Layers	Neurons in each hidden layer	Learning Rate	Epoch	Average Accuracy
Test 1	2	128	0.001	98000	88.20%
Test 2	2	256	0.001	82000	77.20%
Test 3	2	512	0.001	86000	70.40%
Test 4	3	128	0.001	100000	81.10%
Test 5	3	256	0.001	58000	74.30%
Test 6	3	512	0.001	50000	71.90%
Test 7	4	512	0.001	98000	70.30%
Test 8	4	512	0.003	74000	22.20%

As a result, you won't have to start from scratch and won't have to be concerned about the association cycle or hyperparameters. Attempting to construct a version using Nietzsche's default textual content file with the goal of anticipating users' sentences after they input forty letters; the version will recognise forty letters and anticipate future letter/words using an LSTM neural network in an effort to apply Tensorflow. exceptional input layers We can observe that regardless of the size of the input layer, the output prediction is accurate to the tune of 54% to 55%. The training accuracy for 10 input nodes is around 56 percent, but the checking out accuracy is around 54 percent.The training accuracy for a 20 input node is approximately 56%, but the testing accuracy is around 55%; for a 30 input node, the training accuracy is around 56%,but the testing accuracy is around 55.5 percent, which is identical to 20; and for a 40 input node, the schooling accuracy is around 56%. However, the testing accuracy is around fifty-four percent. 9% of the population .The Neural network did a fantastic job anticipating the ultimate outcome as you can see in the second case where the model observed the string that comes after "str" by entering 6 check cases and passing 5 in our params (i.E more potent, energy) With a precision of around 56%, we're confident in our predictions.

The very final step involves compiling and fitting our version. Here, we're training the model and saving the exceptional weights to nextword1.H5 so that we don't have to retrain it every time we need it and can use our saved model whenever we need it.I have excelled at the schooling records in this area.You can, however, use both educate and validation statistics to train. We used categorical crossentropy, which calculates the pass-entropy loss between labels and predictions. With a mastery rate of zero.001, Adam is the optimizer we can utilise, and we'll gather our version at the metric loss. The end result is as follows:We may load the tokenizer file that we saved in the pickle format into the prediction notebook.Then we'll load the next word model that we've saved in our directory. This equal tokenizer will be used to tokenize each of the input sentences for which we need to make predictions. Following this, anything could be done.While walking through the predictions, we'll use the try and except statements. We use this statement because we don't want the programme to leave the loop if there is an error in locating the enter sentence. We must execute the script for as long as the customer wants it to run. When the user wants to exit the script, he or she must manually select this option. The software will run as long as the user's objectives are met.If you wish to be offered in the next portion of the item, you can examine this by using the predictions script. Within the next segment, I may provide a link to the GitHub repository. As we can see, the prediction model can best forecast maximum lines.

The stop the script line will exit the application and end the model. The entire software may be terminated if we enter the path "prevent the script." The last phrase of the entered line is used to anticipate the rest of the sentences. We may be considering the very last phrase of each line and attempting to match it with the next word that has the best chance.

Table 2

Table 1. Structure of different models with it's highest average accuracy with its epoch for Assamese language.

Test number	Hidden Layers	Neurons in each hidden layer	Learning Rate	Epoch	Average Accuracy
Test 1	2	128	0.001	98000	88.20%
Test 2	2	256	0.001	82000	77.20%
Test 3	2	512	0.001	86000	70.40%
Test 4	3	128	0.001	100000	81.10%
Test 5	3	256	0.001	58000	74.30%
Test 6	3	512	0.001	50000	71.90%
Test 7	4	512	0.001	98000	70.30%
Test 8	4	512	0.003	74000	22.20%

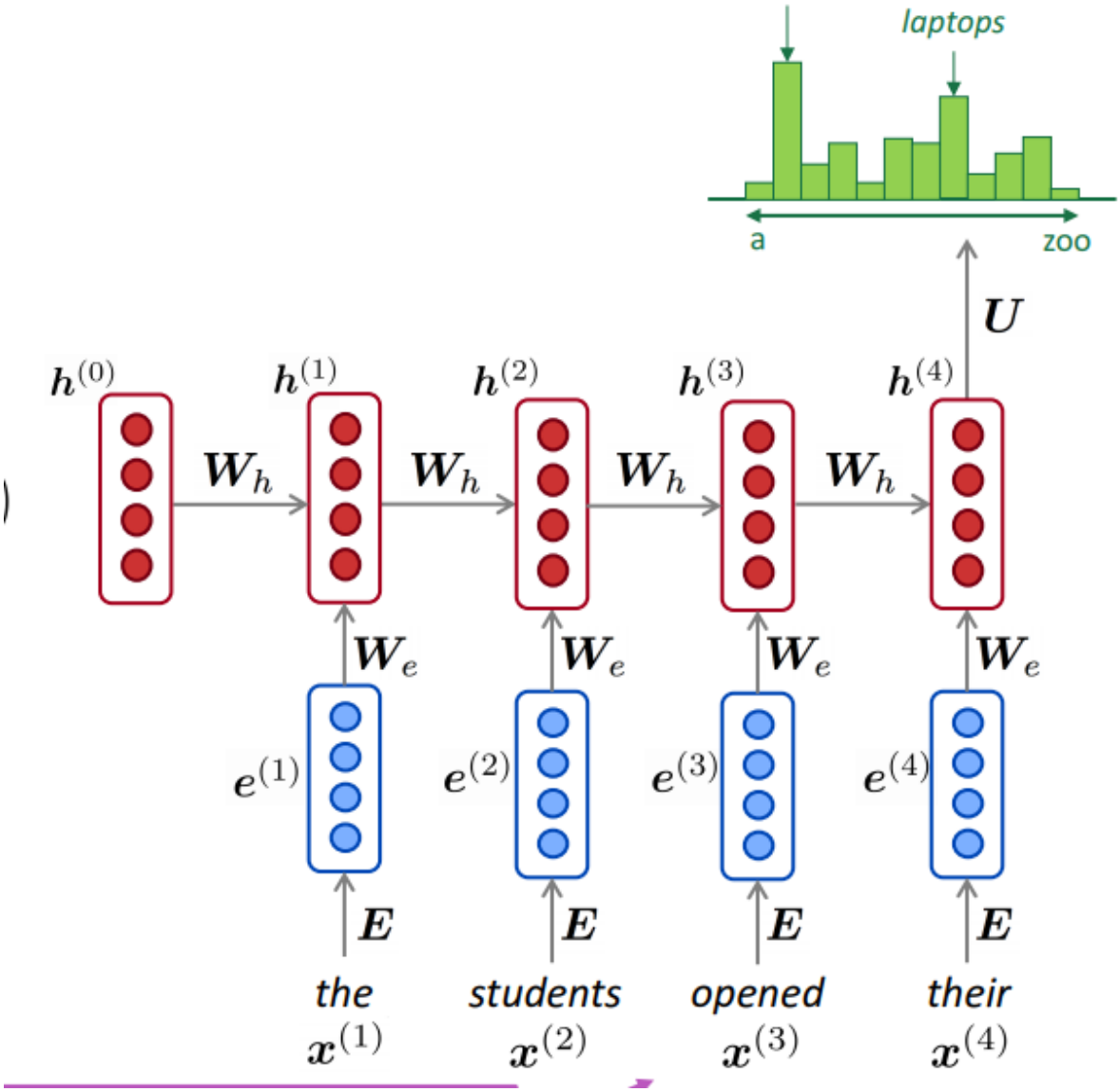


Figure 15

Chapter-5 CONCLUSIONS

5.1 Conclusions

Using device learning algorithms such as RNN to comprehend and body texts and stories will help you understand them quickly. Creating lyrics and songs is a crucial field wherein our algorithm can assist stop-users in anticipating the next phrase in songs because the model is taught on a music lyrics data set. As a bonus, we may train the version to review the weights and recognise the centre aspects of paragraphs/sentences so that we can expect favourable results.Paraphrasing is a technique for expressing someone else's ideas in your own words. To paraphrase a source, recreate a section without changing the sense of the original text, so that our algorithms can predict a wider range of phrases from a single sentence and assist users in bordering n sentences. When the gap between the particular situation and the sentence to be

expected grows, standard RNNs and other language models become less precise. This is where the LSTM comes in to help with the long-term dependence issue, as it includes memory cells to recall the previous setting.LSTM Neural Net can be examined. Our project for this mission is to teach and strive for a set of rules that are finely tuned for this task, and we are particularly interested in implementing an LSTM to achieve appropriate accuracy, as this task is quite difficult because we must predict the user's future text.We are currently manipulating to understand the problem statement as this is a precise problem. We created a 3d vector layer for input and a second vector layer for output and fed them through to the LSTM layer with 128 hidden layers and controlled to get accuracy to around 56% at some point in five epochs. This study explains how the technology predicts and corrects errors.For the metamorphosis dataset, we can expand a massive next phrase prediction. In around 150 epochs, we can significantly reduce the loss. On the available dataset, the next phrase prediction model we constructed is reasonably accurate. The prediction's overall quality is good.

Positive pre-processing steps and version tweaks, on the other hand, can be used to improve the model's prediction.Since March 2006, I've been using the text8 dataset, which is the English Wikipedia sale off. The facts set might be quite extensive, with 16 million words in all. I picked a random subset of statistics with a total of 0.5MM phrases, of which 26k were remarkable phrases, for the purpose of checking out and building a word prediction model. As I will explain later, the answer is no. The intricacy of your version will substantially rise due to the use of unique vocabulary.One of the primary roles in NLP is to replace every word with a phrase vector, which generates a more accurate representation of the phrase's meaning.Please check this blog for additional information on phrase vectors and how they capture semantic meaning. As a result of this design, the RNN can "idea" utilise archaeological statistics to predict the future. However, because pure vanilla RNN has issues with decay and gradient explosion, it is rarely employed. In this example, I employed the LSTM, which employs gates to drift the gradients back in time and solve the extinct gradient problem.In Tensorflow, I created a multi-layer LSTM with 512 devices per layer and two LSTM. The ultimate 5 phrases are used as input to LSTM, and the target phrase is the following phrase.I used sequence loss as a loss characteristic. For a total of 120 seasons, the model was prepared. To assess educational continuity, I study each teach loss and confusion.

A typical metaphor for grading a language model's overall performance is confusion. The alternative to the standard set check effects for the large range of phrases is what makes things hard. The version number rises to reduce confusion. The model had 35 perplexity after a hundred and twenty epoch instruction.On a few sample tips, I tested the model. The version makes available the pinnacle three words from which the user can choose. See the image below. The model performs effectively because it was trained with a limited vocabulary of 26k words.For the metamorphosis dataset, we can expand a massive next phrase prediction. In around 150 epochs, we can significantly reduce the loss. On the available dataset, the next phrase prediction model we constructed is reasonably accurate. The prediction's overall quality is good.Positive pre-processing procedures and version tweaks can, however, be applied to improve the model's prediction.

5.2 Scope of the future

In this work, I explored the problems with insert predictive textual content and attempted to solve them using machine learning algorithms. Machine learning strategies must be optimal in the context of strategies that give device analysis and case evaluation based on user input or demand, which is a major issue for capturing predictive textual material without it. With the use of these tactics, the device becomes bendy in response to user input. However, before using system studying methodologies, developers must first determine what records they should save, how to store them, how to select a case for later retrieval from similar problems, and how to merge a new instance into a memory structure.We will eliminate core t9 flaws (loss of adaptability and learning ability) and improve memory and accuracy by utilising existing systems and machine learning generation.

REFERENCES

[1] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio,“Maxout Networks, ”30th Int. Conf. March. Read. ICML 2013, no. PART 3, pages2356-2364, Feb. 2013, Achieved: May 16, 2021. [Online]. Available:http://arxiv.org/abs/1302.4389.

[2] A. Coates and A. Y. Ng, “A reading feature with K-means presentations,”Lect. Computer Notes. Science. (including Subser. Lect. Notes Artif. Intell.Lect.NotesBioinformatics), vol. 7700 STUDY NUMBER, pages 561–580, 2012, doi: 10.1007 / 978-3-642-35289-8_30.

[3] P. V. Ca, L. T. Edu, I. Lajoie, Y. B. Ca, and P.-A. M. Ca, “Stacked ConflictAutoencoders: Learning Practical Representation in the Deep Network by locationDenoising Criterion Pascal Vincent Hugo Larochelle Joshua Bengio Pierre-AntoineManzagol, ”2010.

[4] J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun, “Stacked What-WhereAuto encoders,”Jun. 2015, Accessed: May 16, 2021. [Online]. Available:http://arxiv.org/abs/1506.02351.

[5] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, “Semi-supervised learning about Ladder networks, ”in Advances in Neural InformationProcessingSystems, Jul. 2015, vol. 2015-January, pp. 3546–3554, Accessed: May 16, 2021.[Online]. Available at: https://arxiv.org/abs/1507.02672v2.

[6] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep-rooted networksin order to study the unsupervised unplanned presentation of program presentations, ”2009, pp. 1–8,doi: 10.1145 / 1553374.1553453.

[7] A. A. Efros and T. K. Leung, "Texture Synthesis by Non-parametric Sampling1999.

[8] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Super-Resolution Based on theModel,” no. April, pages 56-65, 2002.

[9] J. Hays and A. A. Efros, “The completion of the scene with millions ofphotographs,” The community. ACM, vol. 51, nxa. 10, pages 87–94, October 2008, doi:10.1145 / 1400181.1400202.

[10] J. Portilla and E. P. Simoncelli, "Parametric texture model based on integrated calculations ofcomplex wavelet coefficients," Int. J. Computer. Vis., Vol. 40, no. pages 49-71,2000, doi: 10.1023 / A: 1026553619983.

[11] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” Dec.2014, Accessed: May 16, 2021. [Online]. Available:https://arxiv.org/abs/1312.6114v10.

[12] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli,“Deep Uncontrolled Reading using Nonequilibrium Thermodynamics, ”32nd Int.Conf.March. Read. ICML 2015, vol. 3, pages 2246–2255, March 2015, Accessed: May16, 2021. [Online]. Available at: http://arxiv.org/abs/1503.03585.

[13] I. J. Goodfellow et al., "Generative Adversarial Networks." Accessed: May 16,2021. [Online]. Available at: http://www.github.com/goodfeli/adversarial.

[14] E. Denton, S. Chintala, A. Szlam, and R. Fergus, “Deep Model Modelsusing the Laplacian Pyramid of Adversarial Networks, ”Adv. Neural Inf. The process. Syst., Vol. 2015-January, pp. 1486–1494, Jun. 2015, Achieved: May 16,2021. [Online]. Available at: http://arxiv.org/abs/1506.05751.

[15] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, “DRAW: A a repetitive neural network of imaging, ”at the 32nd International Conference in Machine Learning, ICML 2015, Feb. 2015, vol. 2, pages 1462–1471, Accessed: May 16, 2021. [Online]. Available: https://www.youtube.

[16] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox,“Discriminated Feature Disclosure Feature by Exemplar ConvolutionalNeural Networks, ”IEEE Trans. An analog pattern. March. Intelli., Vol. 38, no. 9, pages1734- 1747, Jun. 2014, Accessed: May 16, 2021. [Online]. Available:http://arxiv.org/abs/1406.6909.

[17] M. D. Zeiler and R. Fergus, “LNCS 8689 - Visualization and Understanding Convolutional Networks, ”2014.

[18] AI Blog: Inceptionism: Deep in Neural Networks https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html(accessed May 17, 2021).

[19] D. J. Heeger and J. R. Bergen, “Pyramid-based texture analysis / synthesis,” in Procedures ACM SIGGRAPH Conference on Computer Graphics, 1995, pp. 229–238, doi: 10.1145 / 218380.218446.

[20] L. A. Gatys, A. S. Ecker, and M. Bethge, A Neural Algorithm of Artistic ,J. Vis., Vol. 16, no. 12, p. 326, Aug. 2015, Accessed: May 17, 2021. [Online].Available at: http://arxiv.org/abs/1508.06576.

[21] K. Simonyan and A. Zisserman, “Social networks are very deep on a large scale. image recognition, ”Sep. 2015, Accessed: May 17, 2021. [Online].Available:http://www.robots.ox.ac.uk/.

[22] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, “Photo-Single Take the SVBRDF through the Rendering-Aware Deep Network, ”ACM Trans.Graph., volume. 37, no. 4, October 2018, doi: 10.1145 / 3197517.3201378\

[23] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, “Texture Networks:Feed forward Synthesis of Textures and Stylized Images, ”33rd Int. Conf. March.Read. ICML 2016, vol. 3, pages 2027–2041, March 2016, Accessed: May 17, 2021.[Online]. Available: http://arxiv.org/abs/1603.03417.

[24] J. Johnson, A. Alahi, and L. Fei-Fei, “Loss of vision with real-time transfer of style and advanced refinement, ”in Lecture Notes in Computer Science (including sub-paragraphs Lesson Notes on Artificial Intelligence and Lesson Notes on Bioinformatics), Mar. 2016, vol. 9906 LNCS, pages 694-711, doi: 10.1007 / 978-3-319-46475-6_43.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: ASimple How to Prevent Neural Networks from Full Installation, ”2014. Accessed: May18, 2021. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[26] J. Deng et al., "Imagenet: A high-quality photographic website," CVPR, 2009, Accessed: May 16, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.155.1729.

[27] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox, “Reading Manufacture Seats, Tables and Cars With Convolutional Networks, ”IEEE Trans.An analog pattern. March. Intelli., Vol. 39, when. 4, pages 692–705, April 2017,doi: 10.1109 / TPAMI.2016.2567384.

APPENDIX

CODE:-

```
Import tensorflow as tf

from tensorflow.Keras.Preprocessing.Token import Tokenizer

from tensorflow.Keras.Layers import Embedding, LSTM, Dense

tensorflow.Keras.Models import Sequences

from tensorflow.Keras.Utils import to category_of

from tensorflow.Keras.Optimizers import Adam
```

```
import pickle

import numpy as np

input os

from google.Colab import documents

uploaded = documents.Add ()

record = open (&amp;quot;textfile.Txt&amp;quot;, &amp;quot;r&amp;quot;, encoding = &amp;quot;utf8&amp;quot;)

# store document within the listing

strains = []

because I file:

rows.Close (i)

# Convert the list into a man or woman unit

records = &amp;quot;&amp;quot;

for I inside the strains:

records = &amp;#39;&amp;#39;. Be part of (strains)

#Return pointless gadgets with area

records = records.Return (&amp;#39; n&amp;#39;, &amp;#39;&amp;#39;) .Update (&amp;#39; r&amp;#39;, &amp;#39;&amp;#39;) .Replace (&amp;#39; ufeff&amp;#39;, &amp;#39;&amp;#39;) .Replace (&amp;#39;&amp;quot;&amp;#39;, &amp;#39;&amp;#39;). Update (&amp;#39;). ”&amp;#39;,&amp;#39; &amp;#39;) #New

line, cart return, unicode character -&amp;gt; repair area

#delete pointless spaces

facts = records.Break up ()

records = &amp;#39;&amp;#39;. Be a part of (statistics)

records [: 500]

tokenizer = tokenizer ()

tokenizer.Fit_on_texts ([data])

# saves token for predictive characteristic

pickle.Unload (tokenizer, open (&amp;#39;token.Pkl&amp;#39;, &amp;#39;wb&amp;#39;))

sequence_data = tokenizer.Texts_to_sequences ([data]) [0]

sequence_data [: 15]

collection = []

in i in grade (three, len (data_ collection)):
```

phrases = facts collection [3: +1]

collection. Combine (phrases)

print ("Length of sequence is:", len (series))

series = np.Array (sequence)

series [: 10]

X = []

y = []because I am respectively:

X.Append (i [0: 3])

y. Integrate (i [3])

X = np.Array (X)

y = np.Array (y)

Print ("Data:", X [: 10])

print ("Answer:", y [: 10])

y = to_section (y, num_classes = vocab_size)

y [: 5]

model = Sequence ()

model.Add (Embedding (vocab_size, 10, input_length = 3))

version.Upload (LSTM (1000, return_sequences = True))

imodeli.Upload (LSTM (a thousand))

version.Upload (Dense (a thousand, activation = "relax"))

version.Upload (Dense (vocab_size, activation = "softmax")

version.Dutty ()

from tensorflow import cameras

from keras.Utils.Vis_utils import plot_model

keras.Utils.Plot_model (version, to_file = 'plot.Png', show_layer_names = True) from t

tensorflow.Keras.Callbacks input ModelCheckpoint

take a look at region = ModelCheckpoint ("next_words.H5", alert = 'loss', verbose = 1, save_best_only = True)

version.Bring together (loss = "categorical_crossentropy", optimizer = Adam (learning_rate = 0.001)) model.In

shape (X, y, epochs = 70, batch_size = sixty four, callbacks = [checkpoint])

from tensorflow.Keras.Fashions import load_model

import numpy as np

import pickle

Upload version and tokenizer

model = load_model ('next_words.H5')

tokenizer = pickle.Load (open ('token.Pkl', 'rb'))

def Predict_Next_Words (model, token, textual content):

collection = tokenizer.Texts_to_sequences ([text])

sequence = np.Array (series)

preds = np.Argmax (model.Are expecting (series))

predicted_name = ""

by way of key, fee in tokenizer.Word_index.Objects (): if fee == preds:

predicted_word = key

smash

print (predicted_word)

update a_ expected word

at the same time as (True):

textual content = input ("Enter your line:")

if text == "0":

Print ("Application Completed .....") break

different:

attempt:

text = text.Split ("")

textual content = text [-3:]

print (textual content)

Predict_Next_Words (version, token, textual content)

except Different as in:

print ("An errors happened:", e)

go on