# **Next Words Prediction Using Recurrent NeuralNetworks**

Sourabh Ambulgekar<sup>1,\*</sup>, Sanket Malewadikar<sup>2,\*\*</sup>, Raju Garande<sup>3,\*\*\*</sup>, and Dr. Bharti Joshi<sup>4,\*\*\*\*</sup>

**Abstract.** Next Word Prediction is additionally called Language Modeling. It is the undertaking of predicting what word comes straightaway. It is one of the major assignments of NLP and has numerous applications. Attempting to make model utilizing nietzsche default text record which will foresee clients sentence after the clients composed 40 letters, the model will comprehend 40 letters and anticipate impending top 10 words utilizing RNN neural organization which will be executed utilizing Tensorflow. Our Aim of creating this model to predict 10 or more then 10 word as fast as possible utilizing minimum time. As RNN is Long short time memory it will understand past text and predict the words which may be helpful for the user to frame sentences and this technique uses letter to letter prediction means it predict a letter after letter to create a word

### 1 Introduction

Natural Language Processing (NLP) is a significant part of artificial Intelligence, which incorporates AI, which contributes to finding productive approaches to speak with people and gain from the associations with them. One such commitment is to give portable clients anticipated "next words," as they type along within applications, with an end goal to assist message conveyance by having the client select a proposed word as opposed to composing it. As LSTM is Long short time memory it will understand the past text and predict the words which may be helpful for the user to frame sentences and this technique uses a letter to letter prediction means it predicts a character to create a word. As writing an essay and framing a big paragraph are time-consuming it will help end-users to frame important parts of the paragraph and help users to focus on the topic instead of wasting time on what to type next. We expect to create or mimic auto-complete features using LSTM. Most of the software uses different methods like NLP and normal neural networks to do this task we will be experimenting with this problem using LSTM by using the Default Nietzsche text file also known as our training data to train a model.

Next Word Prediction is also called Language Modeling that is the task of predicting what word comes next. It is one of the fundamental tasks of NLP and has many applications.

#### 2 Literature Review

This section describes the methodology adopted for the literature review. This paper represents an exploration of the contributions that have already been made in the academic field.

[1] multi-window convolution(MRNN) algorithm is implemented, also they have created residual-connected minimal gated unit(MGU) which is short version of LSTM in this cnn try to skip few layers while training result in less training time and they have good accuracy by far using multiple layers of neural networks can cause latency for predicting n numbers of words.

[2] This paper used RNN algorithm and also they have used GRU another form of RNN for code completion problem as RNN help to predict next code syntax for users. Authors claim that their method is more accurate compare to existing methods. They have separated next word prediction in two components: within-vocabulary words and identifier prediction. They have used LSTM neural language model to predict within vocabulary words. A pointer network model is proposed to identifier prediction.

[3] Authors worked on Bangla Language. They have proposed a novel method for word prediction and word completion. They have proposed N-gram based language model which predicts set of words. They have achieved satisfactory results.

[4] Authors have used LSTM for next word prediction for Assamese language. They have stored transcripted language according to International Phonetic Association (IPA) chart and fed to their model. They created a model for physically challenged people. This model uses Unigram, Bigram, Trigram, based Approach for next

<sup>&</sup>lt;sup>1</sup>Ramarao Adik Institute of Technology Mumbai, Maharashtra

<sup>&</sup>lt;sup>2</sup>Ramarao Adik Institute of Technology Mumbai, Maharashtra

<sup>&</sup>lt;sup>3</sup>Ramarao Adik Institute of Technology Mumbai, Maharashtra

<sup>&</sup>lt;sup>4</sup>Ramarao Adik Institute of Technology Mumbai, Maharashtra

<sup>\*</sup>e-mail: ambulgekars@gmail.com

<sup>\*\*</sup>e-mail: malewadikarsanket088@gmail.com

<sup>\*\*\*</sup>e-mail: garande.raju.16ce1029@gmail.com

<sup>\*\*\*\*</sup>e-mail: bharti.joshi@rait.ac.in

word predict tion and was found out average to predict the

word but accuracy was around 30-40 percentage.

[5] In this paper they created a auto-next-keyword for Bengali language which was challenging and It was found out that it is hard to get good accuracy by using RNN algorithm as due to it's vanishing gradients and heavy recurrent NN take more time to train and test.

[6] They useded predicting next character high-lighter(PNCH) for Indian language it was more of text correction and less about next word prediction but was quite good to understand. The Method called hit and miss but accuracy is less and the model was not efficient for this kind of problem statement.

[7]This was first approach to tackle this kind of problem, the paper discusses about LM and perplexity algorithm which is base of natural language processing, This help us to make a 3D input data for our model.

[8] 1-degree feature patterns algorithm is useded to Solve the problem of Vanishing but provide less accuracy as basic sentence was useded in training and testing data like 'A man cries', 'A baby cries', 'A dog barks', and 'A cat mews'.

[9] As GPT is quite huge and costly model for this kind of task as word prediction is a simple project, GPT will only act like wasting useful resources for simple task.

Most of the paper is trying to create a model to predict next text, few of the paper was helpful, paper to predict next code using SVM and RNN, As this algorithm are helpful but new algorithm like LSTM may predict good results for this problem statement below are some limitations of the existing system.

- This process of predicting the next Word is quite complex because we have to predict words which user think so it is predicting the thoughts of the user so the accuracy is quite low compare to other ML project
- algorithms like SVM, Decision Tree, etc are not providing good results and take more time to predict the results as the task is quite complex.
- To be able to make useful predictions, a text predictor needs as much knowledge about language as possible so we have to keep training the neural network continuously on new languages and new data.

### 3 Proposed methodolgy

### 3.1 Data pre-processing

This proposed work in "Fig. 1" is an illustration to create a flexible model that can help users to detect next word while understanding user vocable in a fast and effective manner so user need to provide 40 letters then it passes this letter to LSTM NN and predicts N number of letters

- As shown in the above diagram, like in "Fig. 1" providing input data up to 40 letters later this sentence will pass through LSTM Neural Network
- Letter LSTM understand and learn every letter, letter by letter and create a score for the next letter.

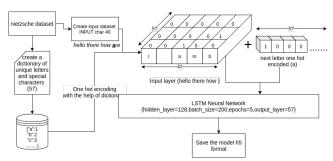


Figure 1. Our Proposed Work flow

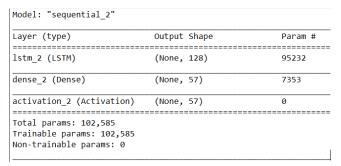


Figure 2. Neural network Architecture

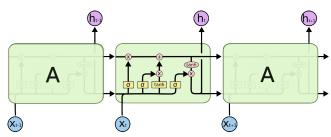


Figure 3. Recurrent Neural network

- This score then again will pass through the same LSTM and later it will predict a word letter by letter.
- Below is our neural network architecture plus our implementation methodology using the Tensorflow library.
  - Letter to bits, As computers, don't understand words so converting words to bits or array of bits using NumPy software.
  - Now creating a 3D array of all words it's like one-hot encoding for all letter and unique characters (200285, 40, 57) this was our training data
  - Later passing this X features to our model with input Neural node 40 and hidden node 128 then this will have an output layer with node equal to the input node.

#### 3.2 LSTM Networks

Step 1: first import our helpful model Numpy pandas and other modules later importing Nietzsche default txt which is our dataset.

Step 2: The way to LSTMs is the cell express, the even line going through the highest point of the outline.

# **Sequence Diagram**

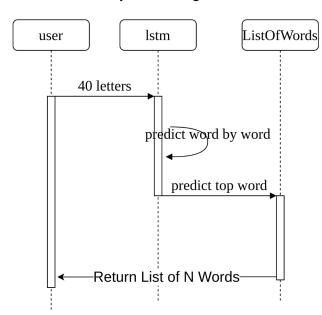


Figure 4. sequence diagram of application

The cell state is similar to a transport line. It runs straight down the whole chain, with just some minor direct communications. It's simple for data to simply stream alongside it unaltered.

The LSTM can eliminate or add data to the phone state, painstakingly managed by structures called entryways.

Gates are an approach to alternatively let data through. They are made out of a sigmoid neural net layer and a pointwise duplication activity.

The sigmoid layer yields numbers somewhere in the range of nothing and one, depicting the amount of every segment ought to be let through. A worth of zero signifies "let nothing through," while a worth of one signifies "let everything through!"

A LSTM has three of these gate, to secure and control the cell state.

# 4 Data Description and Data cleaning

The dataset contains 25,107 words from ebook author Franz Kafka. The Datasets for text data are easy to find and we can consider Project Gutenberg which is a volunteer effort to digitize and archive cultural works, to "encourage the creation and distribution of eBooks". From here we can get many stories, documentations, and text data which are necessary for our problem statement.

### 5 Experimental design

- User: Users job is to submit 40 input words for The LSTM Neural Network like in the "Fig. 4"
- LSTM: LSTM is train on default next word prediction text file it will calculate it's weight and predict a letter by letter for a top word.

- 1. "It is hard enough hi i am sanket whats y"
- 2. "which does not hurt us makes us stronger."
- 3. "i am not sad that you lied to me, i am upset that from now on I cannot have trust on you."
- 4. "those who were seen vibing were thought to be insane by those who could not hear the tune."
- 5. "is though enough to remember my opinions, without also remembering saurab my reasons for them!"
- 6. "not a lack of effection, but a lack of friendship that raju makes unhappy marriages."

Figure 5. Input test cases

• List Of word: This will holdall the words and if user want N number of top works it will count N word and return list of words to user.

#### 5.1 Experimental result

- Above outline like in "Fig. 7", that the n-grams approach is inferior contrasted with the LSTM approach as LSTMs have the memory to review the setting from further, harking back to the substance corpus. While starting another endeavor, you ought to consider one of the current pre-arranged designs by looking on the web for open-source executions. Thusly, you will not have to start without any planning and don't need to worry about the arrangement cycle or hyperparameters.
- Trying to create model using Nietzsche default text file which will predict users sentence after the users typed 40 letters, the model will understand 40 letters and predict upcoming letter/words using LSTM neural network which will be implemented using Tensorflow.
- This product has more scope on social media for syntax analysis and semantic analysis in natural language processing in Artificial intelligence.
- We try to tune some input layers for different input layers we can see that whatever the input layer size may be the output prediction has accuracy of 54% to 55%
- for 10 input node the training accuracy is around 56% but the testing accuracy is around 54%
- for 20 input node the training accuracy is around 56% but the testing accuracy is around 55%
- for 30 input node the training accuracy is around 56% but the testing accuracy is around 55.5% same as 20
- for 40 input node the training accuracy is around 56.3% but the testing accuracy is around 54.9%

From the above example "Fig 5" inputting 6 test cases and passing 5 in our params, the Neural network did a very great job in predicting, result as you can see in the second case in "Fig 6" the model found out the string which comes after "str" (i.e stronger, strength)

From "Fig 7" With the accuracy of around 56% our model did a very good job in real test case scenarios predicting these input test cases.

it is hard enough hi i am sanket whats y
['a ', 'upiai ', 'e ', 'temtcid ', 'ic ']
 it is which does not hurt us makes us str
['ength ', 'onger, ', 'iver ', 'ange ', 'ungarity ']
 i am not sad that you lied to me, i am u
['pon ', 'nder ', 's ', 'tility ', 'ltimate ']
 those who were seen vibing were tho
['ught ', 'se ', 're ', 'igh ', 'm ']
 is though enough to remember my opinion
[' and ', ', ', 's ', '. ']
 not a lack of effection , but a lack of
['the ', 'a ', 'man ', 'something ', 'his ']

Figure 6. N(5) number of outputs

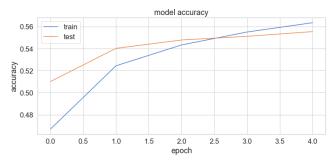


Figure 7. Accuracy on training and testing data

### **Future Work and Conclusion**

- Understanding the paragraph using machine learning algorithms like RNN can help soon to understand and frame paragraphs and stories on their own.
- Creating lyrics and songs can be a major field in which this algorithm can help the end-users to predict the next phrase in songs considering the model is train on a music lyrics data set.
- As more data we can train the model which will reevaluate the weights to understand the core features of paragraphs/sentences to predict good results.
- Paraphrasing means formulating someone else's ideas in your own words. To paraphrase a source, you have to rewrite a passage without changing the meaning of the original text, so our algorithms can predict more number words considering a single sentence and help users to frame n number of sentences.

Standard RNNs and other language models become less exact when the hole between the specific circumstance and the word to be anticipated increments. Here's when LSTM comes being used to handle the drawn-out reliance issue since it has memory cells to recall the past setting. You can study LSTM Neural Net.

Our task in this project is to train and try an algorithm that best fit this task and mostly we are looking forward to implementing an LSTM to get good accuracy as this task is quite complex because we have to predict the user's future text which he will be thinking

At present we manage to understand the problem statement as this problem is unique, we created a 3d vector

layer of input and a 2d vector layer for output and feed through to the LSTM layer having 128 hidden layers and manage to get accuracy to around 56% during 5 epochs

This paper presents how the system is predicting and correcting the next/target words using some mechanisms and using TensorFlow closed-loop system, the scalability of a trained system can be increased and using the perplexity concept the the system will decide that the sentence is having more misspelled and the performance of the system can be increased.

## References

- [1] J. Yang, H. Wang and K. Guo, "Natural Language Word Prediction Model Based on Multi-Window Convolution and Residual Network," in IEEE Access, vol. 8, pp. 188036-188043, 2020, doi: 10.1109/ACCESS.2020.3031200.
- [2] K. Terada and Y. Watanobe, "Code Completion for Programming Education based on Recurrent Neural Network," 2019 IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA), Hiroshima, Japan, 2019, pp. 109-114, doi: 10.1109/IW-CIA47330.2019.8955090.
- [3] Habib, Md AL-Mamun, Abdullah Rahman, Md Siddiquee, Shah Ahmed, Farruk. (2018). An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction. International Journal of Intelligent Systems and Applications. 2. 47-54. 10.5815/ijisa.2018.02.05.
- [4] Partha Pratim Barman, Abhijit Boruah, ARNN based Approach for next word prediction in Assamese Phonetic Transcription, Procedia Computer Science, Volume 143,2018, Pages 117-123, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.10.359.
- [5] S. M. Sarwar and Abdullah-Al-Mamun, "Next word prediction for phonetic typing by grouping language models," 2016 2nd International Conference on Information Management (ICIM), London, 2016, pp. 73-76, doi: 10.1109/INFOMAN.2016.7477536.
- [6] M. K. Sharma, S. Sarcar, P. K. Saha and D. Samanta, "Visual clue: An approach to predict and highlight next character," 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), Kharagpur, 2012, pp. 1-7, doi: 10.1109/IHCI.2012.6481820.
- [7] Minghui Wang, Wenquan Liu and Yixion Zhong, "Simple recurrent network for Chinese word prediction," Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), Nagoya, Japan, 1993, pp. 263-266 vol.1, doi: 10.1109/IJCNN.1993.713907.
- [8] Y. Ajioka and Y. Anzai, "Prediction of next alphabets and words of four sentences by adaptive junction," IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, WA, USA, 1991, pp. 897 vol.2-, doi: 10.1109/IJCNN.1991.155477.
- [9] Joel Stremmel, Arjun Singh. (2020). Pretraining Federated Text Models for Next Word Prediction using GPT2