NEXT WORD PREDICTION USING DEEP LEARNINGAPPROACH

[1]Himani Dighorikar, [2]Shridhar Ashtikar, [3]Ishika Bajaj, [4]Shivam Gupta, [5]Dilipkumar A. Borikar
[1,2,3,4,5]Department of Computer Science and Engineering,
Shri Ramdeobaba College of Engineering and Management, Nagpur
Maharashtra, INDIA
[1] dighorikarhm@rknec.edu, [2] ashtikarsa@rknec.edu, [3] bajajij@rknec.edu, [4] guptasb_1@rknec.edu,
[5] borikarda@rknec.edu

Abstract

Predicting the next word has been one of the most important subjects of discussion in Natural Language Processing. Nowadays, instead of wasting time on writing everything and then proof reading it, we simply use Auto-complete. We use it every day but don't give it the time to understand the processing. Natural language generation (NLG) focuses on generation of natural, human interpretable language. NLG is a methodology that allows us to predict the next word in a sentence most likely to be used. The concepts of Deep Learning such as Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) models are used for simplifying the process of writing and suggesting. This paper aims to provide the analytical study of text prediction and give a formal representation of how text can be correlated to produce a set of words that can be driven out from the data.

Keywords:Deep Learning, Natural Language Processing, Recurrent neural networkLong Short-Term MemoryNatural language generationBi-LSTM

DOINumber:10.14704/NQ.2022.20.11.NQ66027 **NeuroQuantology 2022; 20(11): 247-253**

1. Introduction

Natural Language processing, a Machine learning approach which is mainly used to generate natural language from data. It is a rapidly growing field of study with an expanding number of applications like Text Summarization, Automatic Image captioning, Machine translation, etc. One of the widely used applications is Predictive Text Generation. People use various applications and they often like texting, and whenever they type something, some suggestion pops up that tries to predict the next word that must be intended to be written. This model predicts a few words which have a high probability of following the preceding words. This can be useful in many ways, reducing the number of keystrokes needed by the user to write, which aims to increase the typing speed of the users. The predictive system can be seen as an intelligent agent that assists people in sentence composition hence reducing the physical efforts in composing the sentences.

This paper aims to study and gather various approaches which have been applied to develop a predictive model. It describes the analysis and implementation of the approaches by comparing with the existing work.

1.2 Literature Review

Whenever research on an existing system starts, it is necessary to review some of the earlier studies of researchers in the field which helps to understand the system and its complexities. We have researched published literature on predictive systems to broaden and diversify our knowledge about the topic. This section includes the predictive text system work of previous researchers.

There exist studies based on how the word prediction model can be built on unigram, bigram and trigram [**1**]**.** Authors have described the prediction model to predict the next word in Assamese language using the higher order n-grams.

The traditional NLP approach for text prediction is the n-grams approach. Each gram is a word in another set of words and training is performed in order to extract information and create a set of features to categorize the data **[2]**. This mainly uses joint probability tables to increase the rate of understanding but also recursively requires lots of information to get trained which makes it an unfeasible approach. However, the Naive Bayes overcomes the issue of recursive training but is less accurate.

In the context of predicting in an applicable time, Latent semantic analysis (LSA) was created, which analyses the relationship between the text and even after being an accurate technique for prediction, its inference highly depends on the frequency of words and not on the sequence of words. The authors developed the word predictive model using the hybrid technology of Naive Bayes and LSA and efforts were made to improve the prediction precision through Gradient Descent **[3][4]**. Later, various statistical approaches for text generation such as bag of words, word2vec are applied which generates a bag of vocabulary for text generation. These approaches could generate text at character level or word level without ensuring the generation of meaningful sentences because they lack understanding of the context of the sentence. In order to ensure grammatical errors, methods like Lemmatization, POS Tagging, etc. are adapted**[5].**

Fazly A, et al. **[7]**describes how Authors developed a Word prediction model using Machine learning and new feature extraction and selection techniques which was adapted from Mutual Information(MI) and Chi Square (X^2). They casted this problem as a word classification task and a bunch of words were classified to determine the most correct word in a given context.

Dipti Pawade, et, al. **[8]** have worked in the field of text generation in which they generated a new story by combining two different stories. They have given multiple input files and then generated an output story aligned to the stories of input files to some extent.J. Yang, et al. **[9]** have proposed a MCNN-ReMGU model based on Multi-Window Convolution and Residual Network for natural language word prediction which solves the problem of redundant network layers which lead to poor network performance by adopting residual connection to the original MGU and also the activation function is modified to ReLU to train deeper networks.

O. F. Rakib, et, al. **[10]** have proposed next word prediction for Bangla Language using GRU (Gated Recurrent Unit) based RNN (Recurrent Neural Network) on n-gram dataset to develop language models that can predict the word(s) from the input sequence supplied is the recommended technique. Daniel C. Cavalieri, et al. **[11]** presented an interpolation model to develop a word prediction model by merging the part of speech based model and n-gram language model using three different languages namely Spanish, Portuguese and English. For the word prediction task, a partial derivative function is defined to derive the interpolation model which was used to improve the prediction.

Habib, et al. **[13]** proposed a word prediction model in Bangla language using stochastic model i.e, n-gram language model approach which predicts a set of words after a given sentence.BaekCheol Jang, et al. **[14]** proposed the sequence classification problems in the Word prediction System using CNN and Bi-LSTM model and tried to solve the problem of extracting meaningful information from big data, classifying it into different categories, and predicting end-user behaviour or emotions.

RNN-based models view text as a sequence of words, and are intended to capture word dependencies and text structures but due to limitation of vanishing gradient, more featured models are used such as LSTM and Bi-LSTM **[12]**. Bi-LSTM accuracy as compared to LSTM is examined more accurate since input flows in forward and backward direction and it learns quickly as compared to LSTM **[6].**HarshaVardhana, et, al. **[15]** developed a word level predictive model using LSTM and RNN that simply generates text based on the history of sentences.

2.Methodology

2.1 Modelling Approach

Text prediction has numerous implementations and in our work we have proposed Bi-directional LSTM which overcomes the drawback of simple RNN and LSTM.

Long Short term Memory (LSTM)

The LSTM recurrent unit tries to “remember” all the past knowledge that the network has seen so far and to “forget” irrelevant data.LSTM has three layers as a forget gate, input gate and output gate which makes it capable of learning long term dependencies.

Forget Gate: Ex: Aman is a nice person. Vishal is evil.

In the above example, Aman and Vishal are the subjects. As the system works on sentences, It must understand that in the second sentence there is no context of Aman being used so no need to store it and the forget gate will remove it from the memory cell.

Input Gate:Ex: Vishal knows swimming and he is a gold medallist in it.

Now in both sentences we have a common subject which is Vishal, so now the system should realize the context is matching and store the previous one in memory for further processing.

Output Gate:Ex: Vishal is very brave and he wants to work in ______.

During this process, we have to complete the second sentence. Now, when the system sees “work in” they found it related to Vishal and based on the current expectation we have to give relevant words to fill in the blank. This is the function of Output gate.

The vital part when it comes to predicting the next words is the context and which involves understanding sentence from both the directions. And to solve that Bi-directional model is used.

Bi-Directional LSTM (Bi-LSTM)

Bidirectional LSTM is a recurrent neural network used primarily on natural language processing. Unlike standard LSTM, the input flows in both directions, and it’s capable of processing data from both aspects backward and forward. It’s also an effective tool for modelling the sequential dependencies between words and phrases in both directions of the sequence.

Bi-LSTM adds one more LSTM layer, which reverses the direction of information flow. It means that the input sequence flows backward in the additional LSTM layer and then we combine the outputs from both LSTM layers. This functionality of Bi-LSTMs adds the advantage of complete and faster learning on the input sequence.

For example:

Sentence: Apple is something that …

It might be about an apple as a fruit or an Apple, a company. Thus, LSTM doesn’t know what “Apple” means, since it doesn’t know the context from the future.

It can be interpreted as “Apple is something that competitors simply cannot reproduce”. Or it can be, “Apple is something that I like to eat”.

In both sentences the context of Apple is different and it depends on previous data and it only happens with backward propagation.

As LSTM only recognizes the relationship between values at the beginning and end of a sequence. LSTM has no concern with current word and previous sequence. Hence LSTM is not able to generate more meaningful output. However Bi-LSTM will have a different output for every word in a sequence.

2.2 Proposed Methodology

The figure 2.2.1 describes an overview of the proposed methodology. The process starts from the data extraction which follows pre-processing on the data and data transformation to get the tokens (words) from the text. The model is built using Bi-LSTM and LSTM layers and finally the model is trained which is discussed in detail in section 2.5. Each iteration evaluates the model results of training in terms of its accuracy and the model is saved. The saved model is then used to predict words based on user input to evaluate the model's performance.

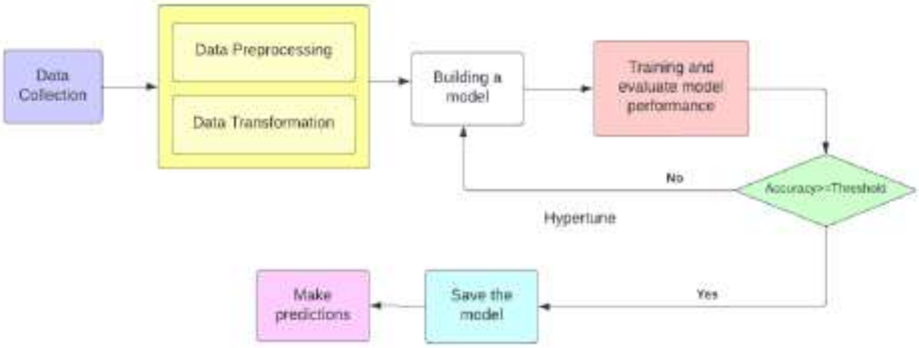
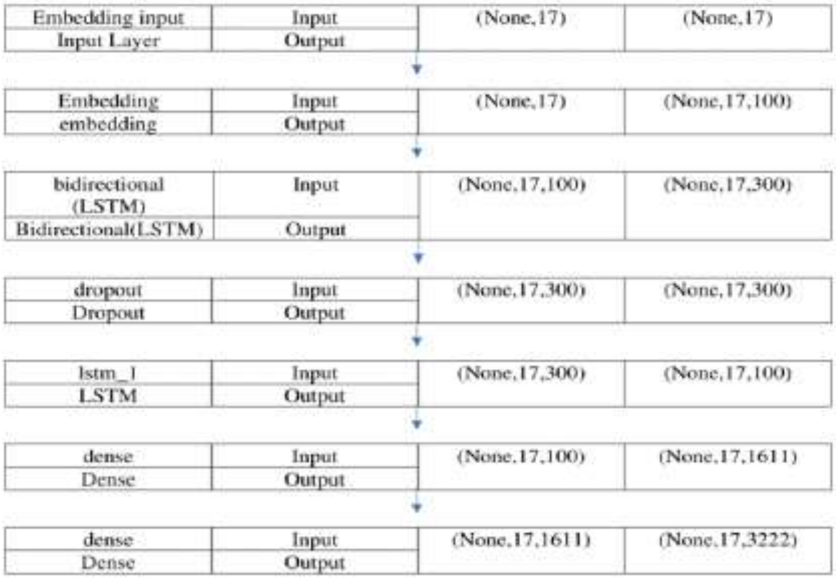


Figure 2.2.1: System Overview

Figure 2.2.2 describes the architecture of the deep learning model utilized for the task of predictive system generation. It consists of an Embedding layer followed by Bi-LSTM of batch size 150 with a dropout layer. The returned sequence by the Bi-LSTM model is again passed to the second layer which is a LSTM layer. Dense layer with "ReLU" and "Softmax" activation function is added.

This will scan the sequences of text from left to right and right to left and will understand the text with its context. Finally, at each iteration it will save the results and that saved model will be used to predict the words based on the input given to the system.



Embedding input Input Layer	Input Output	(None,17)	(None,17)
Embedding embedding	Input Output	(None,17)	(None,17,100)
bidirectional (LSTM) Bidirectional(LSTM)	Input Output	(None,17,100)	(None,17,300)
dropout Dropout	Input Output	(None,17,300)	(None,17,300)
lstm_1 LSTM	Input Output	(None,17,300)	(None,17,100)
dense Dense	Input Output	(None,17,100)	(None,17,1611)
dense Dense	Input Output	(None,17,1611)	(None,17,3222)

Figure 2.2.2: Architecture of predictive model

2.3 Data Pre-processing

The Dataset used in our project is a story named Metamorphosis by Franz Kafka which provides the data of a specific domain for text generation. The dataset was in text format, so all the punctuations were removed and the dataset was cleaned. The sentences of fixed word length were selected and then split into words, to achieve uniformity in the data.

A standard set of pre-processing is performed using the Tokenizer library from Keras. The tokenizer finds all the unique words from the sentences. After this, it will generate a dictionary where each token is assigned a particular place. Once the tokens are created, using those tokens input sequences are generated.

For example,

Suppose we have a sentence: "**friendly laugh that made her unable to speak straight away**"

dictionary = {"friendly" : 112, "her" :14, "laugh" : 89, "made" : 90, "speak" : 55, "that" : 12, "to" : 10, "unable" : 15…}

So using the dictionary, the input sequence for the sentence would be: [112, 89], [112, 89, 12],....... [112, 89, 12, 90, 14, 15, 10, 55]

Based on the previous inputs, the next word will be predicted. Then we have used padding sequences which would normalize the input sequences and ensure all sequences have the same length.

2.4 Model Construction

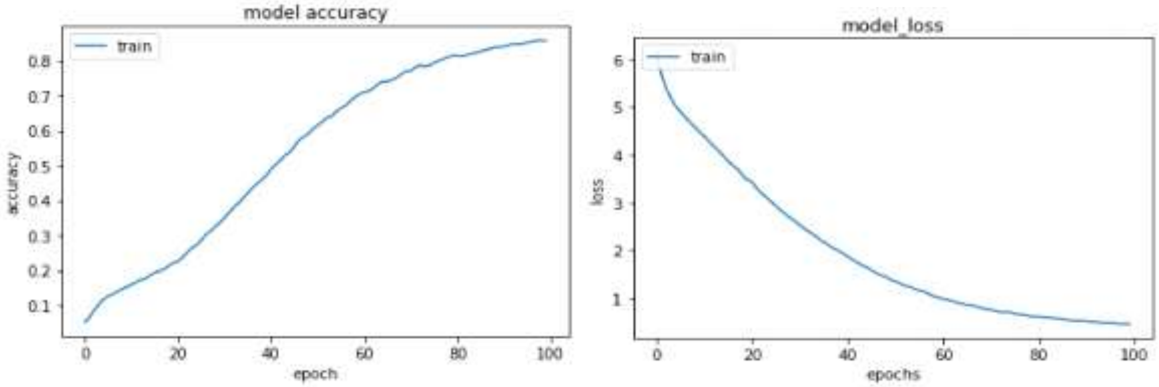
The class labels are converted to a one-hot encoded vector using Numpy arrays which converts the categorical data into a better format. A sequential model is built using an embedding layer which enables it to convert the word into a fixed length vector. The next layer is a Bi-LSTM layer followed by a LSTM layer with activation functions like ReLu and Softmax. The model's structure and layers are shown in figure 2.2.2.

2.5 The Prediction process

After a model is trained an input sequence of seed words is given as an input. These seed words help the network to initiate the process of prediction. A random line is considered from the input text and the model returns the words with the highest probability and this predicted word is then appended back to the input seed sentence and is passed on to the next layer. This will occur iteratively until the desired length is reached.

3.Result and Discussion

The model is trained and evaluated where the accuracy was seen to be 88% with 150 hidden layers added in the Bi-LSTM model and 100 hidden layers added in the LSTM layer. These are followed by a dense layer with ReLU and Softmax as activation functions. It is difficult to understand how the language model has evolved over time before analyzing the outcomes. The following figures 3.1 show the training and loss curves which illustrates how the model has progressed and adapted during the training phase.



(a) (b)

Figure 3.1: Model Performance a) Training accuracy b) Training loss

The model's accuracy can be increased by giving the model as much as data to learn which will familiarize it with different words so that it can understand the meaning and utilization of words together to form different sentences. The model's training is observed by tuning it with different hyper parameters as mentioned in the methodology. Table I shows results given by model on some seed text.

Table I - Predicted Text from the proposed model

Seed Text	Predicted Text	Sentence with Predicted text
If he succeeded in falling out of	bed in this way	If he succeeded in falling out of bed in this way
After a while he had already moved so far	far across that it would have been more than enough	After a while he had already moved so far across that it would
The first response to his situation	situation had been new but that he had heard the door	The first response to his situation had been new but that he had heard the door
But I understand that	was a doorway	But I understand that was a doorway
He seemed disappointed	when the	He seemed disappointed when the
I had to rush him	to her but	I had to rush him to her but
I was amazed	and in	I was amazed and in

Conclusion

This paper aims at providing a better alternate Text Prediction System based on existing models like LSTM, Bi-LSTM. The proposed model uses an embedding layer to convert the word into a fixed length vector, then a Bi-LSTM layer followed by a LSTM layer with activation functions like ReLU&Softmax, to learn the relationship between current word and previous sequence to generate a meaningful sentence.

This model predicts multiple words for the given input sequence and lets the user choose which word is appropriate for them. It has an accuracy of 88% for the dataset mentioned above. The prediction is dependent on the dataset as the model only knows the words used in the dataset. This dependency can be reduced by training the model with various datasets and making it learn new words and sequences.

On comparing results with the pre-existing models, our model has an accuracy of 88% where the other models gave an accuracy of 72% (LSTM) for the given dataset. Thus, we can say that this model gives better predictions.

References

[1] M.P. Bhuyan, S.K. Sarma - “A Higher-Order N-gram Model to enhance automatic Word Prediction for Assamese sentences containing ambiguous Words”, Volume-8, Issue-6, August, 2019

[2] W.B.Cavnar, J.M.Trenkleetal., “N-gram-based text categorization,” Ann Arbor MI, vol. 48113, no. 2, pp. 161–175, 1994.

[3] Peter Foltz, “Latent semantic analysis for text-based research”, 1996

[4] Henrique X. Goulart, Mauro D. L. Tosi, Daniel Soares-Gonc̦alves, Rodrigo F. Maia and Guilherme Wachs-Lopes: “Hybrid Model For Word Prediction Using Naive Bayes and Latent Information”, 2018.

[5] C. Aliprandi, N. Carmignani, N. Deha, P. Mancarella, and M. Rubino, "Advances in nip applied to word prediction," J. Mol. Biol., vol. 147, pp. 195- 197,2008.

[6] Radhika Sharma, Nishtha Gael, Nishita Aggarwal, Prajyot Kaur and Chandra Prakash,- “Next word prediction in Hindi using Deep Learning techniques”,2019.

[7] Fazly A., “The Use of Syntax in Word Completion Utilities” Master Thesis, University of Toronto, Canada, 2002.

[8] Dipti Pawade, AvaniSakhapara, Mansi Jain, Neha Jain, KrushiGada, “Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM”, International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.6, pp.44-53, 2018.

[9] 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.

[10] O. F. Rakib, S. Akter, M. A. Khan, A. K. Das and K. M. Habibullah, “Bangla Word Prediction and Sentence Completion Using GRU: An Extended Version of RNN on N-gram Language Model,” *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9068063.

[11] Daniel C. Cavalieri, Sira E. Palazuelos-Cagigas, Teodiano F. Bastos-Filho, and Mario Sarcinelli-Filho, “Combination of Language Models for Word Prediction: An Exponential Approach”, IEEE Transactions on audio, speech, and language processing, VOL. 0, February 2015.

[12] Sourabh Ambulgekar1,Sanket Malewadikar2, Raju Garande, and Dr. Bharti Joshi, “Next Words Prediction Using Recurrent NeuralNetworks”, 2021.

[13] Habib, Md AL-Mamun, Adbullah Rahman, Md Siddiquee, Shah Ahmed, Farruk, “An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction”, International Journal of Intelligence Systems and Applications, 2018.

[14] BaekCheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang and Jong Wook Kim- “Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism”, 2020

[15] Harsha Vardhana Krishna Sai Buddana, PVS. Manogna, Surampudi Sai Kaushik, Shijin Kumar P.S, “Word Level LSTM and Recurrent Neural Network for Automatic Text Generation” International Conference on Communication and Informatics, 2021.