International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

Recurrent Neural Network based Models for Word Prediction

S.Ramya, C.S.Kanimozhi Selvi

Abstract— Globally, people are spending a cumulative amount of time on their mobile device, laptop, tab, desktop, etc,. for messaging, sending emails, banking, interaction through social media, and all other activities. It is necessary to cut down the time spend on typing through these devices. It can be achieved when the device can provide the user more options for what the next word might be for the current typed word. It also increases the speed of typing. In this paper, we suggest and presented a comparative study on various models like Recurrent Neural Network, Stacked Recurrent Neural Network, Long Short Term Memory network (LSTM) and Bi-directional LSTM that gives solution for the above said problem. Our primary goal is to suggest the best model among the four models to predict the next word for the given current word in English Language. Our survey says that for predicting next word RNN provide accuracy 60% and loss 40%, Stacked RNN provide accuracy 62% and loss 38%, LSTM provide accuracy 64% and loss 36% and Bidirectional LSTM provide accuracy 72% and loss 28%.

Keywords: Artificial Neural Networks, Recurrent Neural Networks, Long Short Term Memory, Bi-directional LSTM.

I. INTRODUCTION

Amino acids are little biomolecules with a normal atomic Artificial Intelligence is the process of creating machine to think and act like human by taking percept from the given environment and action that is performed on the environment based on the absorbed percept. Neural network is to imitate the human brain. It consists of neuron and links that join various neurons to form a network. Artificial Neural Networks (ANNs) are the combination of both AI and NN that are created by layers of linked units called as artificial neurons. This network consists of Input, output and hidden layers. The number of input and output layer is one. The hidden layer may vary from one to many. As the number of hidden layers increases the complexity of the network will also increases. This huge network may lead to deep learning [1], [12], [13].

In Artificial Neural Networks when the neurons are connected recurrently then that network is named as Recurrent Neural Network (RNN) [8], [14], [22]. In RNN the first output is based on the given input and the later outputs are based on its previous output. This forms a sequence among the data. RNN contains internal state/memory which helps to process sequence of various inputs. This makes RNN popular and used in various applications like speech

recognition, protein structure prediction, handwriting recognition [10], etc.

Recurrent connections improve the performance of neural network with the ability to known the dependencies among the sequence of data. The problem arrives while training the data in RNN. In training phase, the RNN leads to face vanishing-gradient problem. That is unable to store the long-term dependency between the data [8]. To specifically solving this problem a model named Long-Short Term Memory (LSTM) RNNs is designed. It is very much effective in reducing the vanishing-gradient problem [18]. This network structure consists of one “forget gate” along with input and output gate. The input and output gates regulate the flow of data to hidden neural layer and preserve the extracted features from earlier time steps [21], [18]. It is noted that for the continuous data sequence, the internal numerical value of LSTM model may grow extremely. The forget gate along with the memory provides solution for vanishing-gradient problem [16]. The error that is found can be corrected by Backpropagating it to the same network. This Backpropagation makes LSTM to learn tasks [9] that entail memories of discrete time events that take place thousands or more time steps earlier. LSTM even can work with the event with extended delays.

Two LSTM models are stacked on top of each other to form Bidirectional LSTM (BLSTM), which interpret the input in opposite direction. One LSTM will interpret the input from right to left and the other will work in opposite way, from left to right. The output is decided by the hidden state of both the LSTM [17]. This makes BLSTM network model more powerful than unidirectional LSTM model [8][9]. Similarly, two RNN can be stacked on each other to form Stacked RNN to improve the working of unidirectional RNN.

II RELATEDWORKS

Sequence Prediction has been an interesting chapter for the researcher for a long duration [2], [3], [4], [5], [6], [11]. Here we briefly discuss about some of the work which carried out by different author suggesting different solutions.

Manuscript published on November 30, 2019.
* Correspondence Author
Ms.S.Ramya*, Computer Science and Engineering, Kongu Engineering College, Perunduari, Erode, Tamil Nadu, India.(Email: ramya@kongu.ac.in.)
Dr.C.S.KanimozhiSelvi, Computer Science and Engineering, Kongu Engineering College, Perunduari, Erode, Tamil Nadu, India. (Email: kec.kanimozhi@gmail.com.)

Supachai et al., 2012 [10] make use of BOCR-WP system in predicting the word in the bilingual document that consist of both the English and Thai word. The algorithm includes character recognition, word prediction and word verification. Character recognition makes use of feature extraction and also classification.

Using n-gram tree they attempt to relate the complete word with the list contains predictive words. In word verification, positive and negative matching approach is used by template matching to accept similar and reject dissimilarity character. This system gives a better performance in recognizing combined Thai-English document.

Bharat et al., 2013 [14] work is to support the braille users in using the braille keyboard. To reduce the number of keystrokes they include word prediction facility in braille keyboard. The word prediction is implemented using B+ tree. The prefix indexing technique is used to construct B+ tree. The index value is calculated for the given prefix and the related word is taken from both history and also from domain specific database.

Carmelo et al., 2015 [15] proposed a postgram based word prediction model in suggesting the missing word for Italian language. In their word prediction they focus on reducing the cardinality of finite candidate word set. The candidate word set can be reduced through predicting the part of speech of the missing word. This leads to improvement in the accuracy and processing speed.

SitiSyakirah et al., 2016 [11] conducted experiment on text document using bigram and using trigram. From their experiment they point out that trigram worked better than bigram in predicting the next exact word to be used. In trigram it looks up both the previous and after word for prediction. They work with Hadith and AI-Quaran text document in Malay language. In their work they concentrate on removing ambiguity in some Malay words which improve the efficiency of retrieving the exact text.

Dipti et al., 2018 [23] worked with creating a new story based on the input from the previously written stories. First, they have taken the character and storyline as an input. Later they have taken stories from various volumes written by the same author as an input. In their planned work they combined RNN with LSTM for creating new stories and measured the outcome based on metrics grammar, interest level, events linkage and Uniqueness. Their result shows that combining these two models give improved accuracy with human evaluation when compare to using the model separately.

Our work is based on analyzing the text document in predicting the next word. We pass our text document through various models like RNN, Stacked RNN, LSTM and Bi-LSTM. The final result taken from each model and it is compared with one another.

III PROPOSEDSYSTEM

The overall architecture of our proposed work is shown in figure 3.1.

The following is about the detailed description of the proposed system. The headlines from the articles file are taken. Each headline is given as an input for preprocessing. Then, tokenization is done. Each word is assigning a random number and the maximum length among all the headlines is taken as the vector length and using N_gram sequence and left padding all the headlines are arranged in a vector sequence. The input is passed to various models and output is taken for the training data.

The test data is given with various lengths and the expected length of the output is given. The loss and accuracy for various models are analyzed.

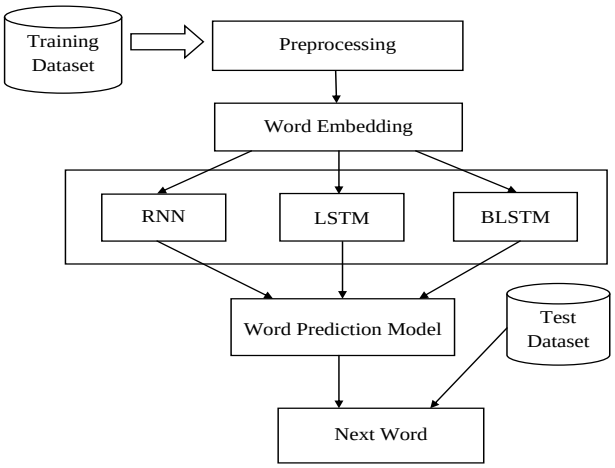

Fig 1 Working of Proposed System Architecture

```
Input: News Articles headline
Output: Next word
Main procedure ()
        Data_preprocessing ()
        Word Embedding ()
        Create model ()
        Predict next word text()
End
Function Data_preprocessing ()
        Remove Unknown data
        Remove punctuation and Converted into lower
        case
End
Function Word_Embedding ()
        Tokenization
    N_gram_sequence
   Generate padding sequences
End
Function Create_model ()
        Add input embedding layer
        Add hidden layer
        Add output layer
End
Function Predict_next_word (text, next_word, model,
max_sequence_len)
      for i in range(next_word)
 Word Embedding ()
        for wordindex in tokenizer.word_index.items()
    if wordindex == predicted
 geerated_word = word
 exit
 display(generated_word)
        end
      end
End
End
```

Fig 2 proposed Algorithm

Retrieval Number: D5313118419/2019©BEIESP
DOI:10.35940/ijrte.D5313.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A Data Preprocessing - Removing Unknown and Punctuation

The articles files are considered and from that file the data about the headline is extracted. The headline may consist of news about some news desk and also "Unknown" data. The first step in preprocessing is to remove the unknown data. In English grammar normally 14 punctuation marks are commonly used. They are question mark, exclamation point, comma, semicolon, colon, dash, hyphen, parentheses, brackets, etc., while predicting the next words punctuations are not necessary. The data cleaning also consist of converting the given sentences into lower case.

In preprocessing there are another two methods used to clean the data they are Stop Word Removal and Stemming. Stop word removal is used to remove unwanted words like was, is, are, etc., from the sentence and Stemming is to form a root word from the given words. In word prediction these two preprocessing methods can't be used as the user might expect the stop word and other words in proper tense.

B Recurrent Neural Network (RNN)

RNN takes the current input and previous output in order to predict the next output [25]. It can handle sequential problem.

$$S_t = F_w(S_{t-1}, X_t) \quad (1)$$

$$S_t = tanh(W_n.S_{t-1} + W_x.X_t) \quad (2)$$

$$Y_t = \mathrm{Wy.St} \quad (3)$$

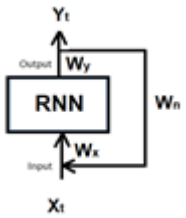

Fig. 3. RNN

Here S_t is the state at time t, F_w is the recursive function, S_{t-1} is the previous state and X_t is the input at time t.

In the above equation Y_t is the output of the current state and W_y is the weight assigned for output state which is passed as the next input. The weight W_y is then replaced as W_x.

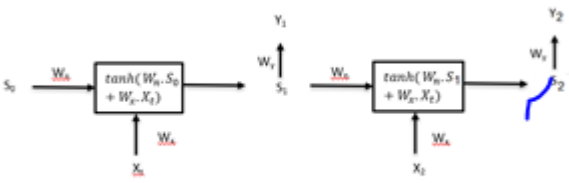

Fig 4. Sequence in RNN

C Long Short-Term Memory

The Long Short-Term Memory (LSTM) is the extension of RNN. It has memory blocks in the hidden layer which is a recurrent layer contains memory cells. It has recurrent connections to retain the temporal state of the network. Also, the network has special gates which are multiplicative units for controlling the information flow across the network. The input and output activations are controlled by the input and output gates respectively. An additional forgets gate is included in the LSTM network to enable or disable its own state. LSTM uses additional connections to learn the timing

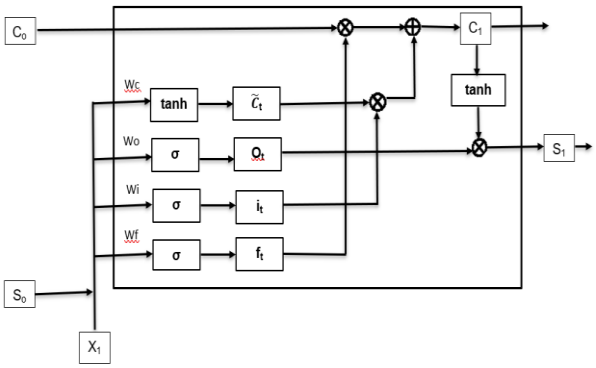

Fig 5. Single LSTM cell

of the outputs [19], [20].

Consider X = (X_1,..., X_t) is an input sequence and h = (h_1, ..., h_t) is an output sequence. The following are different activation equations [7] iteratively for't' time steps.

Forget Gate

$$f_t = \sigma(W_f.S_{t-1} + W_f.X_t) + b_f \quad (4)$$

Input Gate

$$i_t = \sigma(W_i.S_{t-1} + W_i.X_t) + b_i \quad (5)$$

Output Gate

$$O_t = \sigma(W_o.S_{t-1} + W_o.X_t) + b_o \quad (6)$$

Intermediate Cell State

$$\tilde{C}_t = tanh(W_c.S_{t-1} + W_c.X_t) + b_c$$

Cell State

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (8)$$

New State

$$S_t = O_t * tanh(C_t) (9)$$

Here, W_i, W_o and W_c denotes weight matrices used in input, output and intermediate cell state activations respectively. C_t is the cell state and b_i, b_o and b_c is the input, output and cell state bias vectors respectively. And i, f, o is the input, forget and the output gate layer. The tanh activation function is used as cell out activation. To predict the next word, the output layer uses the softmax function in the output layer.

D Bidirectional LSTM Network

The Bidirectional LSTM (Bi-LSTM) extends the LSTM by splitting the neurons of a regular LSTM into two positive time directions and negative time directions. The positive time direction refers to the forward state and the other direction denotes the backward state. This two-time direction takes input from the previous and next state. Our model here consisted of same Deep LSTM structure and difference is that

the LSTM cell is bidirectional so as to keep the information flow in both directions. A constant dropout value of 0.1 is used in all layers. Three dense layers are added at the end of LSTM layers which uses 'relu', 'tanh' and 'sigmoid' activation functions respectively.

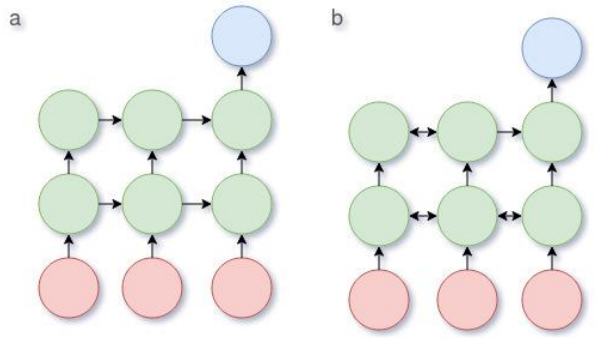

Figure6. (a) LSTM Network (b) Bidirectional LSTM

IV EXPERIMENTAL SETUP

The following section describes the dataset used in our experiments and the results obtained.

A Language and Framework

Python programming language is used to implement our models. We installed anaconda navigator and added packages: jupter, python libraries like: keras, tensorflow as backend, numpy, pandas and scikit-learn.

B Dataset Description

The proposed system uses the "news" data set from Kaggle. It consists of both articles and comments. The articles dataset consists of article ID, article word count, byline, document type, headlines, keywords, web URL, etc., and the comments dataset consists of approve date, comment body, comment id, status, etc,. Headlines from the articles are used in this system. The headlines include various news desk like insider, editorial, sports, games, culture, travel, business, etc,. News from "The New York Times" is taken as a source for this system.

C Parameter for Evaluation

The performance of the system is evaluated based on loss functions Cross Entropy and Kullback-Liebler Divergence. For each loss function three activation function 'sigmoid', 'tanh' and 'relu' has been applied and result is analyzed for various models.

V ANALYSIS AND EXPERIMENTAL RESULTS

The model discussed in this paper consists of input, hidden and an output layer. In the output layer 'softmax' activation function and 'adam' optimizer are used.

Table 1. Loss Analysis

Model	Loss Functions					
	Cross Entropy			KL Divergence		
	σ	tanh	ReLU	σ	tanh	ReLU
RNN	3.683	1.148	1.148	3.619	1.144	0.679
LSTM	3.745	3.146	0.957	3.812	3.198	1.29
Bi-LSTM	3.346	1.947	0.734	3.538	1.981	0.694

Table 2. Accuracy Analysis

Model	Accuracy					
	Cross Entropy			KL Divergence		
	σ	tanh	ReLU	σ	tanh	ReLU
RNN	3.683	1.148	1.148	3.619	1.144	0.679
LSTM	3.745	3.146	0.957	3.812	3.198	1.29
Bi-LSTM	3.346	1.947	0.734	3.538	1.981	0.694

VI CONCLUSION AND FUTUREWORK

In this study we compared various sequential models with various loss functions and activation functions. Through this study the Bidirectional LSTM shows minimum loss compared to that of other models for the training dataset. Combination of Bidirectional LSTM with Cross Entropy: Kullback-Liebler Divergencework better than Bidirectional LSTM with Cross Entropy: RelU.Among the three activation functions sigmoid, tanh and relu, the activation function relu work better. The future work is based on analysis of other models and optimizer.

REFERENCES

1. Partha PratimBarman, AbhijitBoruah,"A RNN based Approach for next word prediction in Assamese Phonetic Transcription" Partha Pratim Barman et al. / Procedia Computer Science 143 (2018) 117–123
2. Even-Zohar, Yair, and Dan Roth. "A classification approach to word prediction." Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. Association for Computational Linguistics, 2000.
3. Prasad R," Next Word Prediction and Correction System Using Tensorflow" E-ISSN No : 2454-9916 Volume : 3 Issue : 5 May 2017
4. Hisham Al-Mubaid"A Learning-Classification Based Approach for Word Prediction",The International Arab Journal of Information Technology, Vol. 4, No. 3, July 2007
5. Seunghak Yu, Nilesh Kulkarni, Haejun Lee, Jihie Kim," On-Device Neural Language Model based Word Prediction, Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 128–131
6. Nicolas Loeff, Ali Farhadi, Ian Endres and David A. Forsyth, Unlabeled Data Improves Word Prediction, 2009 IEEE 12th International Conference on Computer Vision (ICCV)
7. Disha Shree Gupta, "Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks", 2017. Available:https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/
8. HojjatSalehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and ShahrokhValaee, "Recent Advances in Recurrent Neural Networks,". arXiv:1801.01078v3 [cs.NE], 2018.
9. https://machinelearningmastery.com/stacked-long-short-term-memory-networks/
10. Supachai Tangwongsan, BuntidaSuvacharakulton, "OCR with Word Prediction Technique for Bilingual Documents",11th International Conference on Computer and Information Science, IEEE Computer Society,2012.
11. Siti SyakirahSazali, Zainab Abu Bakar and Jafreezal Jaafar, "Word Prediction Algorithm in Resolving Ambiguity in Malay Text", International Conference on Computing for Sustainable Global Development (INDIACom),2016.
12. Bayer, Justin, DaanWierstra, Julian Togelius, and Jürgen Schmidhuber. "Evolving memory cell structures for sequence learning." In International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2009, pp. 755-764.
13. Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," arXiv preprint arXiv:1504.00941, 2015.
14. Kapse, Bharat, and Urmila Shrawankar. "Word prediction using B+ tree for braille users." 2013 Students Conference on Engineering and Systems (SCES). IEEE, 2013.

15. Carmelo Spiccia, Agnese Augello and Giovanni Pilato, “Posgram Driven Word Prediction”, ACM, IC3K 2015 Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, ISBN: 978-989-758-158-8, 2015,Pages: 589-596.
16. Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.
17. Dr. S Lovelyn Rose, Dr. L Ashok Kumar and Dr. D Karthika Renuka, “Deep Learning Using Python”, Wiley India Pvt. Ltd., First Edition, New Delhi, 2019, ISBN: 978-81-265-7991-4
18. Han, Jun, and Claudio Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning." In International Workshop on Artificial Neural Networks, Springer, Berlin, Heidelberg, 1995, pp. 195-201.
19. The Semicolon, “What are Recurrent Neural Networks (RNN) and Long ShortTerm Memory Networks (LSTM)”,2018. Available: https://www.youtube.com/watch?v=S0XFd0VMFss
20. Disha Shree Gupta, “Fundamentals of Deep Learning – Introduction to Recurrent Neural Networks”, 2017. Available:https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/
21. Prasad R," Next Word Prediction and Correction System Using Tensorflow" E-ISSN No : 2454-9916 Volume : 3 Issue : 5 May 2017
22. Hisham Al-Mubaid"A Learning-Classification Based Approach for Word Prediction",The International Arab Journal of Information Technology, Vol. 4, No. 3, July 2007
23. Dipti Pawade, AvaniSakhapara”Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM”, I.J. Information Technology and Computer Science, 2018, 6, 44-53

AUTHORS PROFILE

Dr. C. S. Kanimozhi Selvi is a professor in the Department of Computer Science and Engineering of Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India. She holds a Ph.D. in Computer Science (2011) from Anna University, Chennai. She is at the teaching profession for more than 20 years and more than 10 years research experience. Her areas of academic interest include data mining, Machine Learning and Deep Learning. She has published more than 40 articles in international journals and more than 30 papers in international and national conferences.

S. Ramya is an Assistant Professor in the Department of Computer Science and Engineering of Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India. She received her Bachelor’s degree in Computer Science from Dr.MCET at 2008 and a Master’s degree in Software Engineering from College of Engineering, Guindy at 2010. She is at the teaching profession for 9.4 years. Her areas of academic interest include database management systems and Deep Learning.