JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

NEXT WORD PREDICTION USING RECURRENT NEURAL NETWORK

[1]Rutuja Gosavi, [2]Pradnya Konduskar, [3]Pranav Chavan, [4]Jyoti Kundale
[1]Engineering Student, [2] Engineering Student, [3] Engineering Student, [4]Assistant Professor
[1,2,3,4]Department of Information Technology, Ramrao Adik Institute of Technology, Nerul

Abstract: With Next Word Prediction, also known as Language Modeling, is the task of predicting the next word. It is one of the most important tasks in NLP and has a wide range of applications. Attempting to create a model using the Nietzsche default text record that will predict the clients' sentence after they have written n no of letters, the model will comprehend n letters and predict the top words using RNN neural network and Tensor Flow. Our goal in developing this model was to predict 10 or more words in the shortest amount of time possible. Because RNN has a long short-term memory, it can understand previous content and forecast words, which can assist the user to construct phrases. Also, the model is trained in such a way that it can understand Hinglish language and also can predict words based on it. We present a Bi- Directional Long short term memory network (BI-LSTM) which is a special kind of Neural Network (RNN) in which our thing is to prognosticate the coming word for a given set of words in the model.

Index Terms **- RNN, LSTM, NLP, Text Prediction, Bi-directional LSTM**

I. INTRODUCTION

Natural language processing is used extensively in different operations and has been an area of exploration. Natural Language Processing (NLP) is a significant part of artificial Intelligence, which incorporates AI, which contributes to chancing productive approaches to speak with people and gain from the associations with them. We love texting each other and chancing out that whenever we try to class a textbook a suggestion pops up trying to prognosticate the coming word we want to write. NLP has to deal with. Next Word Prediction. It is also called Language Modelling that's the task of predicting what word comes next. As you are reading this paper you understand every word on the basis of the previous work. Your thoughts have an order. Traditional Neural Networks fail to store a large amount of data, due to which RNN comes into the picture. RNNs are networks that have loops that help to store information Sometimes we come across small sentences such as "Birds fly in the sky" where it is very obvious that the last word will be the sky. But the problem happens when there are big sentences and you need to further back where the information is stored. Consider the sentence "I grew up in U, I speak fluent English". It suggests that the next word is probably the name of the language, and if we want to know which language we need to go back as we need the context of the US. To solve the problem of long-term dependencies LSTM comes into the picture. But on more research, we found out that Bi-directional LSTM is more effective rather than using only LSTM. So, we have used Bi-directional LSTM in this project.

1.1 Data and Sources of Data

For this study the data has been collected from Kaggle. The dataset contains data from two YouTube channel comment sections. Both the data have been combined to form a dataset. Data of sentences in English as well as Hindi sentences written in English have been included. A dataset of sentences has been used to understand the use of the model by predicting the next word of the sentence.

1.2 Theoretical framework

The work which is presented has taken into consideration the users' need for words while typing. Also, the main aim of this model is to create smooth functioning for users and make the work hassle-free. The dataset contains data from two YouTube channel comment sections. Both the data have been combined to form a dataset. The methodology to predict the next words for Hinglish dataset is described.

Jupyter Notebook: Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating notebook documents.

A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ". ipynb" extension.

II. RESEARCH METHODOLOGY

The dataset contains data from two YouTube channel comment sections. Both the data have been combined to form a dataset.

2.1 Pre-processing:

The goal of pre-processing is to get rid of non-essential elements that reduce the model's performance and are useless for the next word prediction. This critical stage establishes this work is the foundation and eliminates all extraneous terms. We have 10 fields and 6508 records here, however, we will only use the title field to forecast the next word. We must delete some unnecessary characters and words from it because they are not beneficial for us to predict and may even reduce the accuracy of our model. After that, the Tokenization process begins wherein each word is assigned a unique id which creates a word index. The majority of the papers attempt to develop a model to predict the next text; however, only a few of them are useful, such as a work that uses SVMN-gram and RNN to predict the following code. Although this approach is useful, a new algorithm such as LSTM or Bi-directional LSTM may be able to predict favorable results for this issue statement.

2.2 Data and Sources of Data

For this study the data has been collected from Kaggle. The dataset contains data from two YouTube channel comment sections. Both the data have been combined to form a dataset. Data of sentences in English as well as Hindi sentences written in English have been included. A dataset of sentences has been used to understand the use of the model by predicting the next word of the sentence.

2.3 Theoretical framework

The work which is presented has taken into consideration the users' need for words while typing. Also, the main aim of this model is to create smooth functioning for users and make the work hassle-free. The dataset contains data from two YouTube channel comment sections. Both the data have been combined to form a dataset. The methodology to predict the next words for Hinglish dataset is described.

A Next word prediction using the N-gram model has made the model more niche by only focusing on the Kurdish language. They have trained the model on the Kurdish text corpus .They had to face more difficulties because the Kurdish text corpus is very limited. To save time while typing the Kurdish language, the N-Gram model is utilized to predict the following word. When a user inputs a word, the system prompts them to type the next five words. That is based on the preceding written word or words, the suggested system will recommend the next five words. This model has an accuracy of 96.3 %(Hochreiter S.et al. 1997)

A Vietnamese Language model used a recurrent neural network. Traditional Neural Networks can only understand words that they have seen before. The N-gram model is not suited for long-term dependencies. The model was trained on 24M syllables constructed from 1500 movie subtitles. In this paper, RNN are explored for a Vietnamese language model. The following is a summary of the contributions: Building a Vietnamese syllable-level language model based on RNNs. Building a Vietnamese character-level language model based on RNNs. Extensive testing on a 24 million syllable dataset derived from 1,500movie subtitles. Also, this model concludes that RNN based language model yields better results. The perplexity of 83.8% is thought to be reasonable as this model outstands the N-gram model in terms of results (Bengio Y et al,1994)

A paper based on the Ukrainian Language analyzed the next word Prediction model but it concentrates more on the Ukrainian Language. One main reason for working with a specific Ukrainian language is because of limited support for Ukrainian language tools. Their sequential character aids in completing the next-word guessing test successfully. The Markov chains produced the most accurate and timely results. The hybrid model produces adequate outcomes, but it is slow to implement the goal of this paper is to examine existing next-word prediction methods based on entered text and put them to the test in Ukrainian language material (Asma Rashid et al, 2021)

In this research for Assamese Phonetic Transcription described a LSTM model for instant messaging, which is a type of RNN with the purpose of predicting the user's future words given a set of current. With an accuracy of 88.20 percent for Assamese text and 72.10 percent for phonetically Tran scripted the Assamese language, this model employs LSTM to predict the next word from a data set of Tran scripted Assamese words (A. F. Ganai et al, 2019)

Next word Prediction using RNN tried to create a model using the Nietzsche default text record that will predict the client's sentence after they have written 40 letters, the model will comprehend 40 letters and predict the top 10 words using RNN neural organization and Tensor flow. Our goal in developing this model was to predict 10 or more words in the shortest amount of time possible. Because RNN has a long short-term memory, it can understand previous material and anticipate words, which can help user structure phrases. Letter-to-letter prediction is used in this technique, which means it predicts a letter after another to build a word (S. Siami-Namini et al,2019)

2.3.1 LSTM VS BI-DIRECTIONAL LSTM

LSTM For a long time, there have been issues with sequence prediction. They are considered one of the most challenging challenges to solve in the data science industry. Long Short-Term Memory networks, often known as LSTMs, have been discovered to be the most effective solution for practically all of these sequence prediction challenges thanks to recent developments in data science. We prioritize our appointments when we plan our day's schedule, right? We know which meeting could be canceled to accommodate a possible meeting if we need to make some space for anything vital. It turns out that an RNN is incapable of doing so. LSTMs, on the other hand, perform tiny modifications to the data using multiplications and additions. Cell states are a system that transports information and preserve information from previously processed inputs. When you use BI-LSTM, your inputs will be processed in two directions: one from the past to the future, and the other from the future to the past. The difference between this strategy and the LSTM that goes backward is that the LSTM that runs backward preserves information from the future, whereas the two hidden states combined maintain information from the past and future at any point in time. Thus, BI-DIRECTIONAL LSTM gave good and correct results.

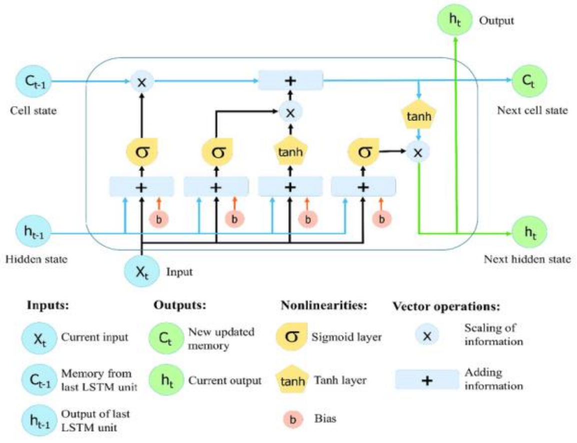

Fig.1. LSTM Architecture

Fig.1 shows it has four interacting layers with a unique method of communication. LSTM networks are a type of RNN architecture that "recalls" recently read values for a random time frame. There are specifically three gates in LSTMs control that gives how the information flow to and from their memories. The new data is fed to the memory using "in- put gate". The "forget gate" has control over how long particular values are held in memory. Activation of the block is affected by the "output gate" that manages the amount of the value contained in memory. These functionalities are shown in figure2. The method of making any neural network have sequence information in both ways backward (future to past) or forwards (ahead to future) is known as bi-directional long-short term memory (BI-LSTM). The blank area in the line "boys go to..." cannot be filled. Still, when we have a future sentence like "boys come out of school," we can easily anticipate the previously blank space and have our model do the same thing, and BI-DIRECTIONAL LSTM allows the neural network to do so.

2.4 System Requirements

Hardware Requirements:

- 4GB Ram
- 256 GB HDD
- Intel 2.8 Ghz i3 Processor

Software Requirements

- Windows
- Jupyter Notebook
- Python Libraries
- YouTube Comments Dataset

III. DESIGN OF SYSTEM

3.1 System Design

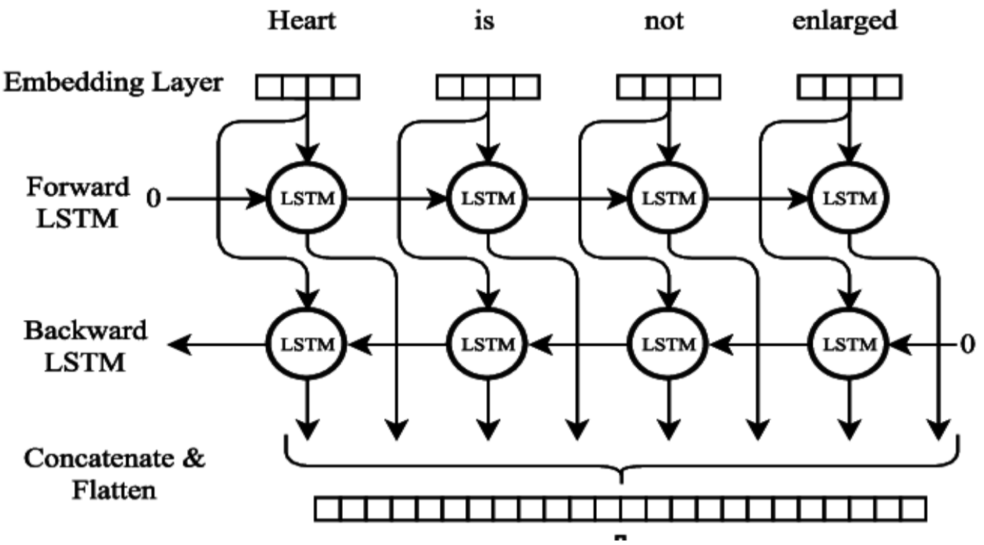

Fig.2. LSTM vs BI-LSTM

LSTM For a long time, there have been issues with sequence prediction. They are considered one of the most challenging challenges to solve in the data science industry. Long Short-Term Memory networks, often known as LSTMs, have been discovered to be the most effective solution for practically all of these sequence prediction challenges thanks to recent developments in data science. We prioritize our appointments when we plan our day's schedule, right? We know which meeting could be canceled to accommodate a possible meeting if we need to make some space for anything vital. It turns out that an RNN is incapable of doing so. LSTMs, on the other hand, perform tiny modifications to the data using multiplications and additions. Cell states are a system that transports information.

3.2 Dataflow Diagram

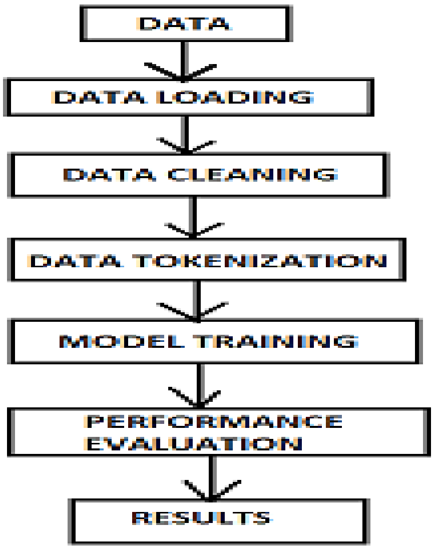

Fig.3 Dataflow Diagram

Figure 3 shows the flow of the overall system.

1. **Data loading:** Data loading is the process of copying and loading data or data sets from a source file, folder or application to a database or similar application.
2. **Data Cleaning:** Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.
3. **Data Tokenization:** The token is a reference (i.e., identifier) that maps back to the sensitive data through a tokenization system
4. **Model Training:** Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.
5. **Performance Evaluation:** Performance evaluation is an important aspect of the machine learning process. However, it is a complex task. It, therefore, needs to be conducted carefully
6. **Results:** the last and very important step is the result and accuracy of the model.

IV. RESULTS AND DISCUSSION

4.1 Experimental Results

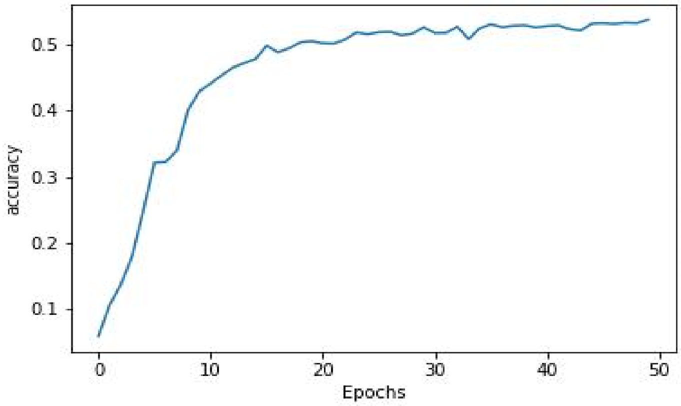

Fig.4 Accuracy of the model

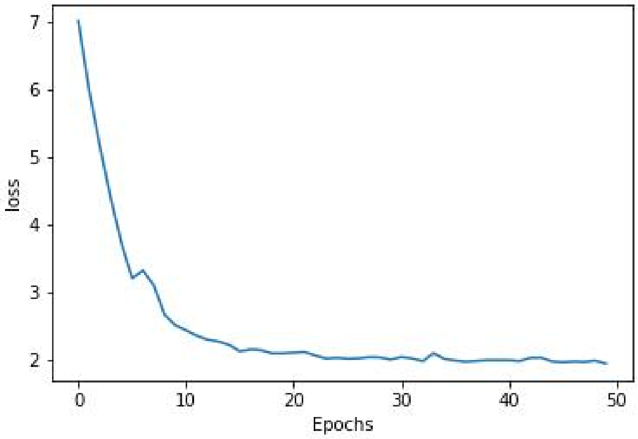

Fig.5 Loss of the model

The table given below compares the accuracy between the existing models and the current model:

Table 1: Performance of Various Model

Sr. No.	Model Name	No. of Word Prediction	Accuracy
1	N-Gram Model	The highest frequency for the top five words Is projected based on the n-gram frequency	92%
2	Long Short-Term Memory (LSTM)	It will count 10 words and give the user a list of them.	58.6%
3	Bi-Directional LSTM (BI- LSTM) [Proposed Model]	Predicts N number of words as per the need	93%

Consider the N-Gram models mentioned in table 1, All 5 types of n-gram models are used whereas the model only works with a specific type of text corpus which is not suitable for all languages. When the system is unable to identify sufficient evidence to anticipate the following word, the N-gram is reduced. Our model works well and does not decrease the accuracy in any instance.

The next model is a Long short-term the BI-LSTM model shows good accuracy of 93%. A BI-DIRECTIONAL LSTM differs from a standard LSTM in that the input flows in both directions. With a conventional LSTM, we may make input flow in one direction, either backwards or forwards. memory (LSTM) in which the accuracy of the model itself is low and also Because the only inputs it has seen are from the past, LSTM only saves information from the past. Our model outruns Long Short-Term Memory in terms of accuracy and storing more information.

We can have information flow in both directions with bi1directional input, maintaining both the future and the past. . Due to this BI-LSTM proves to be the best model for next word prediction. This architecture offers numerous benefits in real-world issues, particularly in NLP. The major reason for this is that every component of an input sequence contains data from the past as well as the present. As a result, by merging LSTM layers from both directions, BI-LSTM can create a more relevant output

4.2 Future scope

Prediction, in the next term, is a critically important skill both now and in the future. This strategy is being used by transitional businesses since it makes them more user-friendly. Although there is still much more research to be done in this particular sector. The BI- LSTM is then used to address the drawn-out dependency issue because it includes memory cells to recall the one set. Our goal in this model is to train and test an algorithm that is suitable for this task and achieves a high level of accuracy. This paper demonstrates how the system uses some mechanisms to predict and correct the next/target words, how the scalability of a trained system can be increased using the Tensor Flow closed-loop system, and how the system will decide that the sentence has more misspelled words and how the system's performance can be improved using the perplexity concept.

Paraphrasing of something is the same thing written or spoken using different words, often in a simpler and shorter form that makes the original meaning clearer. Here our algorithm will predict more relatable words making it easier to form nnumber of sentences with the same meaning. This approach can help end-users predict the next phrase in songs by producing lyrics and tunes, which is a major field in which this approach can help .Smart Compose builds on Smart Reply by predicting what you'll type next in the email body as you type. In this hybrid approach, the topic and previous email are encoded by averaging the word embedding in each field. The averaged embedding's are then combined and sent to the target sequence RNN-LSTM at each decoding step.

V. ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of our teachers and our college. We would like to extend my sincere thanks to all of them.

We are highly indebted to Mrs. Jyoti Kundale Ma'am for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would also like to express our special gratitude and thanks to Dr. Reshma Gulwani Ma'am for giving us such attention and time.

Our thanks and appreciations also go to our college for giving us the opportunity to present and work on this project with all the support and guidance

REFERENCES

[1] Hochreiter S., Schmid Huber J. (1997) .Lstm can solve hard long-time lag problems. Advances in neural information processing systems (1997), pp. 473-479

[2] Bengio Y., Simard P., Frasconi P. (1994). Learning long-term dependencies with gradient descent is difficult IEEE transactions on neural networks, 5 (1994), pp. 157-166.

[3] Asma Rashid, H.K., Saeed, S.A. Rashid, T.A. (2021) Next word prediction based on the N-gram model for Kurdish Sorani and Kur Manji. Neural Compute Applica 33, 4547–4566 (2021).

[4] F. Ganai and F. Khursheed, (2019)" Predicting next Word using RNN and LSTM cells: Statistical Language Modeling," 2019 Fifth International Conference on Image Information Processing (ICIIP), 2019, pp. 469-474

[5] S. Siami-Namini, N. Tavakoli and A. S. Nami, (2019)" The Performance of LSTM and BiLSTM in Forecasting Time Series," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3285-3292

[6] R. Sharma, N. Goel, N. Aggarwal, P. Kaur, and C. Prakash, (2019)" Next Word Prediction in Hindi Using Deep Learning Techniques," 2019 International Conference on Data Science and Engineering (ICDSE), 2019, pp. 55-60.

[7] Kyuhyun Yeon, Kyunghyang Min, Jaewook Shin, Myoungho Sun woo, Manabe Han," Ego-Vehicle Speed Prediction Using a Long Short-Term Memory Based Recurrent Neural Network", International Journal of Automotive Technology, vol. 20, 2019, pp. 713.

[8] 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