NEXT WORD PREDICTOR USING LSTM

Vivek Praful Kharate, Jyotiraditya Chavan, Ratna Patil
Artificial Intelligence and Data Science
Vishwakarma Institute of Information Technology
Pune, India
Emails: {vivek.22210158, jyotiraditya.22210606, ratna.patil}@viit.ac.in

Abstract—**Long words tire to type, but predictive text software in keyboards simplifies it. Another name for next-word prediction is language modelling. One's work is simply predicting the first word to be spoken. It has a number of applications and constitutes the basic human language technology work. This technique is letter to letter prediction and it claims that it is predicting a letter when word is constructed in terms of letter. Long short time memory formula can sense previously typed text and predict words which can be repeated again for people to enclose sentences.**

Index Terms—**LSTMs, Activation function, classification, Next Word**

I. Introduction

Word prediction devices have been made to allow people to communicate comfortably and to aid those who write with difficulty. This paper discusses a languages prototype framework for fast electronic communication that predicts the next word most likely given a certain set of present words. From just initial pieces of text, word prediction techniques will need to predict the most likely preceding word. By providing the user with accurate words, we aim to facilitate instant digital communication. Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory Networks (LSTM) were proposed with deep learning technologies' research and implementation. While prediction problems are considered, LSTM and other nonlinear sequence models can process serialised data efficiently. As a fundamental standard time-series predictive model, LSTM can employ long-range temporal features to predict post-sequential time series using both short-term memory and long-term memory within its memory framework. Thus, the suggested words align well with the specific user's vocabulary options. Hence, a word input predictive model with LSTM has been suggested through this research. As the first step, the textual data set of a specific sector is normalized. Second, an LSTM network is used to train the extensively processed text with the aim of creating an industrial keyword prediction model that is utilized finally to an industry's input system. II. LITERATURE REVIEW The Next Word Prediction model utilized by previous systems predicts the next word will validate the previous word. These systems function on a machine learning approach that is restricted in generating correct syntax. The model used in the above-mentioned study utilized RNN (Recurrent Neural Network) methods to forecast or identify the next word with acceptable accuracy of 86Therefore, since longterm dependency problem was already established in the old system, the RNN model would readily predict ”Boy” if we process short words such as: - He is ”. But if the words were long, then the system might have lost its track and thus create a longterm dependency issue. Second, whenever we apply older word prediction techniques, the RNN model is beset with unidirectionality, another problem. In brief, the RNN or Vanish Gradient Descend is an issue when you have a machine learning network with backpropagation. It does have a great effect, the process of updating the model weights gets highly and badly affected, and the model leads become absolutely useless. We prefer using LSTM (Long Short-Term Memory) instead of RNN with some features due to this. Early models work on the word prediction scheme that expects the word to be completed with an immediate subsequent word. These models work with machine learning techniques that are restricted in their ability to give correct syntax. Multi-window necessary, and a residualconnected lowest gated unit (MGU), short form of LSTM, has also been implemented. To be effective in training time and enhance accuracy, CNN tries to bypass a couple of levels in this case. However, applying several layers of neural network models will introduce latency when making a prediction for a large list of words. Models built using lstm Model algorithms can process more knowledge at a faster rate and make higher-quality outputs compared to models built without applying these algorithms. New technologies have been generating more precise results than the existing system technologies. In this study we take the use of an RNN model and LSTMs employed together for Prediction. Utilization of recent logic data rather than all previous data logs. More accuracy with less execution times. The model does not need any timestamps. Implement some basic logic gates (and, or, and xor) to minimize the past data. Feedback is provided after every step to enhance accuracy.

RNN(Recurrent Neural Network) A continuous neural network was an extension of an instant propagation network with memory. The RNN is iterative in the sense that it performs a single operation for every piece of knowledge, but this output at a fixed rate will be a function of the last computation. It relies on a degreelevel calculation of both current input and last input's result. RNNs have the capability to deal with sequences of inputs in a manner that convey neural networks can't due to its internal representation (memory). Inputs to recurrent neural networks are interdependent. Like CNN and

Identify applicable funding agency here. If none, delete this.

ANN, RNNs also consist mainly of three levels: an input nodes, a dense nodes, and an output units. These levels run one after another as mentioned above. Information is first reached by input layers that are also involved in data preprocessing. After filtering data, the data is then fed into the hidden layers, where the activation function and algorithms of multiple neural networks are executed in an effort to regain useful information. This resulting set of information is then fed into the output layer. The key function of RNN is to provide the same and same weight and bias to every layer, thus making the control variables dependent variables. It will also compress the parameters and remember each earlier output by sending each output to the following hidden layer so that all four layers are combined into one recurrent layer with the same weight and bias applied to all hidden layers. The use of recurrent neural networks differs from the other neural networks in that recurrent neural networks all of them have loops for storing and processing information, though neural networks do not. Another possibility is that recurrent neural networks are able to relate past information to the context at hand. The quality of recurrence neural networks in being able to store information persistently is something which doesn't take place in standard neural networks. The central ingredient of an RNN, or the key to its long memory, is an LSTM. A gated recurrent unit, or a GRU for short, is as good as an LSTM and sometimes better since it has higher speed and precision.

II. THE ALGORITHM A. RNN(RECURRENT NEURAL NETWORK)

Recursive neural networks were an extension of instant propagation neural network in memory management. The RNN is recursive because it does a fixed operation on each bit of information, but at a fixed rate, the outcome relies on the calculation from before. The decision was based on a degreelevel comparison between the immediate past input result and this input. RNNs have the ability to handle sequences of inputs in a manner that convey neural networks are not able to due to its internal representation (memory). The Recurrent neural network inputs are connected. Similar to CNN as well as ANN(artificial neural networks), RNNs also possess a structure with primarily three layers: an input nodes, a dense nodes, and an output units. Once more, these levels follow one another. Information is first accessed via input layers, which also pre-process information. Information is then fed into hidden layers, where multiple neural networks' activation functions and algorithms are executed to collect useful information. Finally, this reservoir of information is forwarded to the output layer. The primary task of RNN is to provide equal and precise same weight and bias to all the layers so that the control variables are dependent variables. It will also reduce the parameters and recall all the previous outputs by feeding each output into the subsequent hidden layer so that the four levels can feed into one recurrent layer with equal weights and biases in all the hidden layers. What makes recurrent neural networks different from the other neural networks is that recurrent neural networks all of them possess loops for data processing and storage, whereas neural networks do not. Another potential way may be that recurrent neural networks can connect past data to the current situation. Information persistence is one of the features of recurrent neural networks that are not present in typical neural networks. The fundamental unit of an RNN, or what is responsible for its long-term memory, is an LSTM. Gated recurrent unit, or GRU, has performed as well as an LSTM and sometimes better due to it being faster and more accurate.

LSTM(LONG-SHORT TERM MEMORY)

Since these networks were created for long-term dependency, their ability to recall information for an extended period of time without having to learn it over and over again is what distinguishes them from other neural networks and makes the entire process easier and quicker. This particular sort of RNN has a built-in memory to store data.

When back propagation is taken into account, neural networks may have vanished gradient descent as a drawback. The value updating mechanism is significantly impacted by the Brobdingnag effect, and the model is now worthless. As a result, LSTM has a hidden layer and a storage cell with three forget, scan, and inputs gates.

This forget gate is mostly used to govern what information has to be deleted that is unnecessary. The input gate stops the addition of new information into the cell, and the output gate stops the transmission of the parts of the cell to a future hidden state. Both gate equations use the sigmoid function, which ensures the value is reduced to 0 to 1.

The input gate suppresses the incorporation of new information into the cell and the output gate suppresses the passing on of components of the cell to a subsequent hidden state. The two gate functions both employ the sigmoid function, reducing the value to 0 to 1.

It has chain-like structure and belongs to RNN but differs from having one-layer neural structure to having four-layer neural. Architectural gates of the structure of LSTM are capable of inserting or deleting any information. Within LSTM, five structural components are there. These are: 1. Input gate

2. Forget gate
3. Cell
4. Output gate
5. Hidden state output

The input gate is given the input information, and then comes the forget gate, which instructs the cell to disregard or forget anything that is not required. It does so by multiplying the value of the irrelevant data by zero, rendering it valueless. The information is then passed back to the cell, where another output gate inspects the output.

A basic LSTM prediction model given our case is illustrated in Fig. 2 below. The model decrypts a word sequence that symbolizes a potential future sub event from the embedding upon converting the given input word sequence symbolizing previous sub events to an associate degree embedding.

III. D. Soft Max activation function

The Softmax activation function scales logits and numbers into probabilities. A Softmax gives a vector (say v) with probability for every possible outcome. The probability in vector v for all possible outcomes or classes sums to one.

Softmax formula is:
probability = exp(value) / sum v in list exp(v)

IV. C. Relu activation function

A rectified linear unit (ReLU) is simply an activation function that gives a deep learning model the freedom to be non-linear and solves the vanishing gradients problem. It recognizes the conclusive part of its case. One of the most popular deep learning activation functions is this one.

ReLU formula is : f(x) = max(0,x)

Methodology

System inputs have to be collected. Tokenizer will be utilized to split inputs into tokens. Tokenized inputs split into segments that will be used in Model 1 of LSTMs. Model 1's LSTM outputs will serve as Model 2's input. Model 2 outputs should be used with the RNN- ReLU activation algorithm (derivatives). In addition to the latest output, other RNN (Softmax) activation functions will be utilized. Model 3 is a prediction model that will be constructed after getting the final derivative output. This Model helps with accuracy and will enable us to pass input from past nodes. :

1) DATA PREPROCESSING : These are simple cleanup operations that make using the information in later phases simpler. Tensor flow library is used to help manage this strategy.

The following are some pre-processing stages that are usually performed:

1. Indicating white spaces
2. Conversions to lower case
3. Eliminating numbers
4. Deleting the punctuation
5. Deleting offensive words
6. Taking out foreign words

2) Preprocessing Text : Removing special characters, stop-words, and text format normalizing.

3) Feature Engineering : Text analysis is the systematic process of reading and comprehending human-written text using computer tools to get business insights. Text analysis technology can categorise, sort, and extract data on its own from texts to find patterns, connections, attitudes, and other useful information. The Term Document Matrix function was used to create term matrices to obtain an accounting of term frequency to determine the rate of occurrences of words. classification accuracy.

4) Tokenizing : The practice of breaking up a huge amount of text into tokens is known as tokenization. These tokens serve as a great starting point for stemmed and inflectional forms the and are particularly helpful for identifying classification. trends. Tokenization is one of the essential social control techniques. It just divides the continuously flowing text into separate word parts. One really simple method would be to divide inputs by home and give each word its own identity.

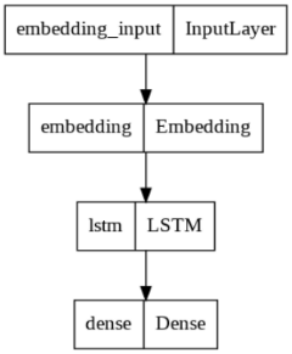

Fig. 1. Architecture model

5) PAD SEQUENCE : It remains challenging to give our neural networks inputs of comparable length when converting texts to numerical values. Not all sentences are the same length. The pad sequence's function is used to truncate some of the larger sequence and replace some of the shorter phrases with zeroes.

Furthermore, since it is frequently indicated if it is necessary to cushion and truncation either at the start or the termination, relying mostly on pre-setting and post setting again for padding arguments and the truncating arguments by default, truncation and artefact arguments may occur at the beginning of the sequence.

V. Results

We have met accuracy and prediction as targeted at the initial stage of the project as presented below accuracy as been mapped in the graphical form as well. The entire dataset is divided up into discrete word phases. A word index is created during segmentation using a predetermined characteristic variety of words. The deployment involved:

The next-word prediction model based on LSTM was effectively trained on the preprocessed text database. During training, the model indicated steady improvement in accuracy as the loss factor decreased consecutively over 20 epochs—from an initial loss of 6.77 to a final value of 2.13. Such a drastic reduction is testament to the model's ability to learn complex sequences of words.

Checkpointing using ModelCheckpoint to store the best model on training loss.

Input testing using an interactive prompt where users input sequences of three words.

Predictions using the trained model, which output contextually relevant next-word suggestions.

Training of the LSTM-based next word prediction model was conducted for 20 epochs with a batch size of 64. Categorical cross-entropy loss was employed as the objective function, suitable for multi-class classification problems like predicting a next word from a huge vocabulary.

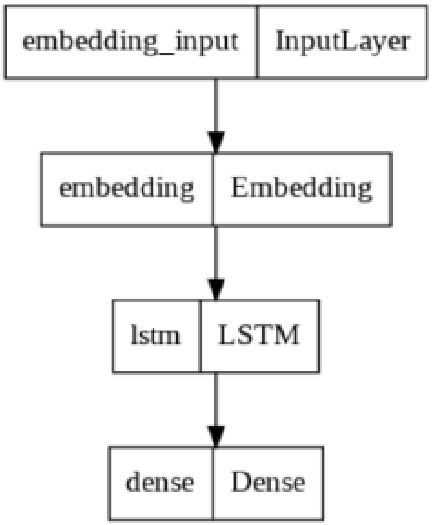

Fig. 2. total reduced loss

1) *Insights:*

- Early rapid improvement: There is a steep decline in loss between the first 5 epochs, since the model picks up simple sequence patterns at a fast rate.
- Slow convergence: From epoch 6 onwards, the decrease in loss is slower, since weights are refined and learning stabilizes.
- Over-all decrease: Training loss reduced by over 68

A. Prediction Accuracy

To test the performance of the LSTM model above numerical measures such as loss, a qualitative review was done using three-word sentences as input and seeing the prediction of the model's next word. These experiments show the capacity of the model to comprehend semantic context and syntax.

1) *Observations:*

- The LSTM model accurately predicted grammatically correct and context-sensitive next words in most instances.
- The model works best when:
- The input sequence has distinct syntactic patterns (e.g., He was quite → a)
- The context is common in training data (e.g., frequent phrases, travel-related words)
- Predictions such as tm can capture biases or oversampling of some tokens in the training data (e.g., ”tm” being part of copyright/licensing notices in Project Gutenberg books).

2) *Strengths:*

- Handles short-term dependencies quite well.
- Identifies typical patterns in English (subject + verb + adjective/noun patterns).
- Acquires domain-specific words from the training corpus.
- Infrequent repetitive predictions (e.g., tm output several times).

3) *Limitations:*

- Overfitting to frequent sequences without semantic knowledge (frequent in low vocabulary scenarios).
- Can struggle with rare/unseen contexts because of vocabulary constraints or lack of training data diversity.

4) *Suggestions for Enhancement:*

- Increase training data with more varied and richer corpora.
- Use beam search during prediction to produce several candidate next words ranked by probability.
- Fine-tune the model with specialized review datasets (e.g., airline or literature sets).

VI. CONCLUSION

On the basis of the presented dataset, the ensuing predictive analytic model is fairly accurate. Applying multiple pattern-discovery techniques is necessary for NLP in order to get rid of noisy data. In around one hundred epochs, the loss was greatly decreased. The processing of huge files or datasets still requires considerable efficiency. To improve the model's predictions, though, limit pretreatment processes and bound model adjustments are frequently developed. As a result, the LSTM-based keyword input prediction system developed in this study may be successfully used in particular sectors and increase user input efficiency. The appropriate industry language prediction model is created and implemented into the input device of the industry by processing the text data set of that industry and training the preprocessed text using LSTM

REFERENCES

[1] Md Robiul Islam, Al Amin, Aniqua Nusrat Zereen, ”Enhancing Bangla Language Next Word Prediction and Sentence Completion through Extended RNN with Bi-LSTM Model On N-gram Language,” arXiv preprint arXiv:2405.01873, 2024

[2] Prakhar Mathur, Khushi Sharma, Shubhankar Kavya, Aman Sharma, Shazia Haque, ”Enhancing Language Modelling with RNN and LSTM-based Next Word Prediction,” Journal of Advanced Database Management Systems, vol. 10, no. 1, pp. 1–7, 2023.

[3] Alwizain Almas Trigreisian, Nisa Hanum Harani, Roni Andarsyah, ”Next Word Prediction for Book Title Search Using Bi-LSTM Algorithm,” The Indonesian Journal of Computer Science, vol. 12, no. 3, pp. 1045–1050, 2023.

[4] R. Sumathy, Shaik Fiza Sohail, Shaik Ashraf, Sudha Yaswanth Reddy, ”Next Word Prediction While Typing using LSTM,” ResearchGate, 2023.

[5] P. Niharika, S. John Justin Thangaraj, ”Long Short Term Memory Model-Based Automatic Next Word Generation for Text-Based Applications in Contrast to the N-gram Model,” Journal of Survey in Fisheries Sciences, vol. 10, no. 1S, pp. 1234–1240, 2023

[6] ”Next Word Prediction Using LSTM,” Journal of Information Technology and Its Utilization, vol. 5, no. 1, pp. 10–13, 2022.Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[8] Vilém Zouhar, Marius Mosbach, Dietrich Klakow, ”Fusing Sentence Embeddings Into LSTM-based Autoregressive Language Models,” arXiv preprint arXiv:2208.02402, 2022.

[9] P. Niharika, S. John Justin Thangaraj, ”Long Short Term Memory Model-Based Automatic Next Word Generation for Text-Based Applications in Contrast to the N-gram Model,” Journal of Survey in Fisheries Sciences, vol. 10, no. 1S, pp. 1234–1240, 2023.

[10] S. Nithin, Sameer Pandit, Tanuja Shastri, Yash Joshi, Dr. Rashmi Amardeep,”Survey on Next Word Prediction and Paraphrasing Using Latent Semantic Analysis”2022