Prediction of Future Word in a Document Using Enhanced LSTM Model

Naresh Alapati, Suvarchala Linga, Sreelekha Kollipara, Pravallika Gude, Afifa Farheen Shaik

Department of CSE, Vignan's Nirula Institute of Technology and Science for Women, Guntur

Corresponding author: Naresh Alapati, Email: alapatinaresh13@gmail.com

Prediction of text for Sentence Completion is the often used technology for speeding up communication and reducing overall text writing time. In most cases, when sending messages to individuals, an individual communicates with a certain group of individuals in a particular way. Our goal is to make instant messaging easier for the user by recommending suitable words. The system remembers the previous words encountered in a semantically sound sentence and learns to connect the input word array to its likely the following word. Proposed system can memorize data and handles the information and data for long period of time. It mainly generates results based on the order of input. The system gets trained by the data given by the user and it remembers long sequence of data. The data is then put to use to create a type of software system that is capable of text data modeling, to generate predictions instantaneously.

Keywords: Deep Learning, Next Word Prediction, Long Short-Term Memory.

2023. In Satyasai Jagannath Nanda & Rajendra Prasad Yadav (eds.), *Data Science and Intelligent Computing Techniques*, 245–253. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-21

1. Introduction

People engage informal conversational exchanges with each other practically every day in the modern world engulfed by social media, which is Instant Messaging (1M) or chatting, and it has also become one of the most popular utilized communication paradigms. In actuality, human communication is primarily personalized, meaning that a person utilizes and maintains a personal lexicon to discover acceptable words to converse with another person. In this project, we'll look at how the next word prediction model we build will work. The model will look at the final word of a sentence and forecast the next word that could be used. This is one of the applications of NLP that deals with prediction. Natural language processing, language modeling, and deep learning will all be used. We'll begin by examining the data before moving on to pre-processing it. After that, we'll tokenize the data before building the deep learning model.

With the introduction and as well as its extensive use of contemporary printed boards, computers with letters, and the use of language as a tool. This needed a thorough understanding of the language's morphology and grammar, as well as the conversation's context. Text generation, particularly next-word prediction, is beneficial to users because it allows them to write more accurately and quickly. The underlying software must be adaptable sufficient to function with a variety of graphical user interfaces tailored to specific applications and preferences. Because fragmentation might be difficult for a user who needs to utilize numerous programs and each forecast differently, single global coordination among all applications is desirable.

When producing the next word, though, faults are frequently visible. As a result, the goal of this research is to examine the available methods for predicting the next word based on typed text and messaging. In today's age of real-time social media, electronic dialogue, as well as communication between the masses, is a common occurrence. The biggest problem with predicting the next word in any regional language is that some of our machines only recognize ASCII values, which is the most widespread text file format on computers and the Internet, in which each alphabetic, numeric, or special character is represented by a 7-bit binary number. While today's recognition pass primarily uses backing-off models ([1]), feed-forward neural network LMs, first described in [2,] have evolved into a significant addition to existing rescoring strategies. To build software based on theoretical computational algorithms, an operational prototyping methodology is utilized.

In actuality, human communication is primarily personalized, meaning that a person utilizes and maintains a personal lexicon to discover acceptable words to converse with another person. The examined approaches for tailored word prediction place a strong emphasis based on the estimation of built-in statistical language models, which are based on the use of a language's dictionary. Without an inbuilt word list, the prediction system could use the existing system's vocabulary, and it could also learn rapidly from previous messaging history, personal documents, and chat logs to be useful right away. The aim of making the system more user-friendly and personalized in the event of a single user or sender is to present various proposals for various receivers.

2. Literature Survey

For the Paper “Prediction of future word in a document using enhanced LSTM model” we have studied the following papers.

Natalia Kryvinska Et al. [1] wrote, “An Approach to Predicting the Next Word in Ukrainian. LSTM, Bi-directional recurrent neural network (RNN), and gated recurrent units (GRU)” are all used in this study. The federated learning model was compared to the long short-term memory (LSTM) with a Coupled Input and Forget Gate (CIFG) language model trained on the server and baseline n-gram model. The LSTM has the most consistent results, followed by the GRU and, last but not least, the bidirectional RNN. The loss study shows that bidirectional RNN has the least loss, followed by LSTM, while GRU has the most. LSTM takes the shortest time to execute, GRU takes a little longer, and bidirectional RNN takes the longest.

Jingyun Yang Et al. [2] wrote, "Multi-Window Convolution and Residual Network Model for Natural Language Word Prediction". Using the MCNN-ReMGU model based on the multi-window convolution and residual-connected minimal gated unit (MGU) network for the prediction of the natural language word prediction.

Nishita Aggarwal Et al. [3] wrote, "Deep Learning Techniques for Predicting Next Words in Hindi". Using the LSTM and Bi-LSTM the consequences and issues faced by the Recurrent Neural Networks (RNN) have been solved. The gates have been used to forget the selected information. The Bi-LSTM is the effective model than the LSTM. And the Bi-LSTM model produces better outcomes. The precision of the Bi-LSTM model is high (79.54%) when compared to the LSTM model (70.89%). But the accuracy during the validation process of the LSTM model is 59.64% and for the Bi-LSTM model is 81.07% respectively. The learning process of the Bi-LSTM model is faster when compared to the LSTM model.

Abhijit Boruah Et al. [4] wrote, "In Assamese Phonetic Transcription, an RNN-based approach for next word prediction was developed." Using LSTM and RNN models. We introduce a Long Short Term Memory network (LSTM) model for instant messaging, which is a type of Recurrent Neural Network (RNN) to predict the user's future word(s) given a collection of current words. Aside from English, this strategy is more complicated in different languages. In Assamese, for example, there are three distinct terms for a person: tumi, toi, and alumni. For a single assertion in English, there might be three or more different sentences in Assamese based on these three styles of addressing. The data set has a maximum accuracy of 88.20 percent.

Sheikh Muhammad Sarwar Et al. [5] wrote," By grouping language models, we can predict the next word for phonetic typing". Using the Phonetic Typing, Next Word Prediction, Language Models. People engage informal conversational exchanges with each other virtually every day in the current world embraced by social media, i.e., Instant Messaging (1M) or chatting, and it has become one of the most widely utilized communication paradigms. We hypothesize that while sending messages by text to a certain group of individuals, user u would utilize a specific linguistic style. On this foundation, we utilize a simple approach to group similar user experiences and create LM based on their textual content. Finally, we choose a certain LM from a set of LMs to create the next word suggestions for a user. However, this is our first try at doing it to categorizing people according to their linguistic styles, and we are currently working on developing an application that can anticipate the next word. For common LMand group-based LM, User1 reported a hit ratio of 40.55 percent and 56.13 percent, respectively. User2 reported a hit ratio of 53.34 percent for common LM and 55 percent for group-based LM, respectively.

Md. Tarek Habib Et al. [6] wrote, "Stochastic Language Models are used for Automated word Prediction in Bangla Language". Using the Word forecasting, deleted interpolation, stochastic model, natural language processing, corpus, N-gram, backoff method. We have three models for measuring accuracy: unigram, bigram, and trigram. The paper's accuracy is 63.5 percent. The Bigram model performs admirably, but the Unigram model performs dreadfully. Although N-gram-based word forecasting works well for English, we apply it for Bangla, which is more difficult to get 100% accuracy since it relies on a huge training corpus. To improve performance, we will employ data from the corpus that has been smoothed out and expand the data corpus size possibly in the future.

Giorgio Vassallo Et al. [7] wrote, "Automatic sentence completion methodology based on word prediction". Using Truncated Singular Value Decomposition (TSVD), LSA, Language Model, Impulse-Response Language Model (IRLM). Word prediction is often based on n-grams frequency statistics, which might demand a lot of data collection and do not take into consideration the text's overall significance. For automated sentence completion, we described a word prediction algorithm. To enhance scalability for huge training datasets, we demonstrated how to make a Word-Word occurrence matrix that may be generated instead of a higher quality standard grid of Word documents. This function estimates the likelihood of a given candidate word occurring. Microsoft claimed a 45.6% accuracy rate. Word Document has a 52.3% accuracy rate. The sum rule has a higher accuracy rate (52.3%) than the product rule (5l. 0%).

Sharvari Bhosale Et al. [8] wrote, "Text forecast Systems for Sentence Completion". Using statistical NLP, N-gram model. Text forecasting for Sentence Completion is a frequently used technology for improving communication speed and lowering overall text composition time. Although search and insertions occur in real-time for the implementation of the hash table in Section 4.2, the model's overall physical memory use was above 3 GB, making it unsuitable for practical application. Tries took up 37% of the space used by the implementation of the hash table in Section 4.3, making them unfeasible for reads into physical memory. N-Gram Language Models were used to create a powerful word prediction system. We have concluded that (a) a statistical technique with instance-based learning performs well, improving the precision with time, and (b) Sorted Array Tries, a space economical linear data structure, may be used to create a prediction made in a real-time system, based on our findings.

Chandana Surabhi.M Et al. [9] wrote, "Future of Natural Language Processing". Using transformational grammar, computational linguistics, grammar, languages, and parsing. Natural Language Processing (NLP) makes machines more human by minimizing the gap between humans and machines. In a nutshell, NLP facilitates human-machine communication. NLP systems will become less proprietary and consequently less expensive if the NLP community promotes the evolution of open-source. The subject is followed by the predicate in a simple phrase., Conjunctions, nouns, pronouns, adjectives, verbs, adverbs, prepositions, and interjections are the components of speech used in an English sentence. The majority of us can communicate in both written and oral forms. We can create a single picture of phrases with a similar connotation by using a sequence of modifications and adding some extra components. Transformational grammar refers to this enlarged grammar. Given a set of criteria, a language may be produced. G = (V, S, P), where V is a collection of variables, is a set of terminal symbols that occur after generation, S is a start symbol, and P is a set of production rules, that can be used to produce a language. L is G's related language (G). Parsing is the conversion transformation of a flat input sentence into a hierarchical structure corresponding to the meaning units of the sentence.

Martin Sundermeyer Et al. [10] wrote, "Modelling Language Using LSTM Neural Networks". Using LSTM neural networks, Language modelling, recurrent neural networks. In terms of perplexity, the performance of typical recurrent neural network designs may be enhanced by roughly 8%. Interpolating an LSTM LM with a big Kneser-Ney smoothed backing-off model in addition to a cutting-edge French recognition system yielded fairly substantial gains.

J. Luis Garcia Rosa Et al. [11] wrote, "In a Connectionist Distributed Representation System, Next Word Prediction". Using Word prediction First, Natural language processing, Neural Network Distributed representation, there were 986 input units and 43 output units make up a connectionist network. in the previous edition. Each of the input units was in charge of representing 42 words, each with 24 semantic attributes. The 42 words plus the end-of-sentence marker were represented by the output units. The suggested system employs a single connectionist architecture with 61 output units and80 input units to handle four words with twenty micro features each as input and three words with twenty micro features each as output, as well as the end-of-sentence marker. The other change is in the way the system works: in the previous version, the output was local, which meant that each unit was accountable for a single word. Both the input and output of Pred-DR have distributed words representations. This permits the system to become more generalized across verbs and nouns, adding new words to the lexicon without dedicating more hardware, as long as their semantic micro features are provided. This eliminates the need for the system to be retrained. There is approximately about 60% of accuracy.

3. Proposed Mechanism

This proposed mechanism is used to predict the following words in a data series. Here we predict output using LSTM as it stores the long memory of trained data. Using other algorithms, the accuracy and the predictions are not appropriate. While using the RNN model, it fails to remember data once a lot of words are fed in. We proposed an LSTM model to predict the word after a long gap.

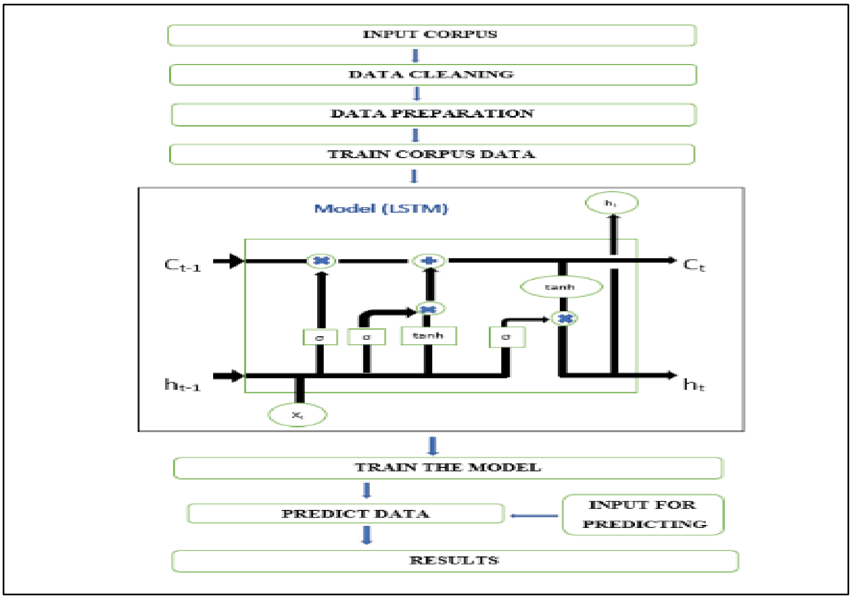

Fig. 1. System Architecture of the LSTM model

The architecture depicts the procedure of our system. The data can be remembered after a time gap according to the layers in the LSTM model. The first layer aids in the data embedding on the hidden layer. The system outcomes are processed in the second layer, and the results are shown in the fully connected layer. Because of the Vanishing Gradient effect, there is evidence that the LSTM Recurrent Neural Network model has a larger memory capacity than a standard RNN.

Algorithm For Proposed System:

```
Input    :    Text Corpus
Output   :  Predicted next word of the sequence
Step 1   :     We reduce the incoming corpus data to lowercase and replace punctuations with
whitespaces.
                         corpus.lower()
                      re.sub(‘[^a-z0-9] +’,’ ‘, corpus)
Step 2   :    The cleaned data is tokenized into words in the form of a list of needed sequences
and then
              transformed into integer sequences in this stage.
```

Step 3 : We use the input sequence length to train the processed data. On the targets, it entails
conducting one-hot encoding.
Step 4 : We construct the LSTM model and set to train. The model foresees the outcome by clearing
the gradients compute's model loss and backpropagation.

$$S_t = \tanh(x_t U^g + p_{t-1} W^g)$$

$$p_t = \tanh(S_t) * o_t$$

Step 5 : The model is moved to evaluation mode at this stage.

In the above algorithm, we take input as corpus data, process the data, and replace punctuations with whitespaces. The processed data is tokenized into a list of words. Using the one-hot encoding technique, we train the processed data according to the user input sequences. Here the use of LSTM is to model the data by calculating the states using the activation functions as sigmoid, tanh. After building, the model is trained where it clears the gradients and backpropagates. It computes the loss of the system using epoch. At last, predictions are evaluated.

4. Results

The proposed LSTM model will be tested against the current RNN model in this part using a confusion matrix with four parameters: accuracy, precision, recall, and f-score. " True, False Positive Rates, True, False Negative Rates, True, False Negative Rates have all been computed for our system. Instructions are referred to as the true negatives as TN that are properly identified as benign, whereas the harmful is the number of true positives TP refers to instructions. Fake positives (FP) and fake negatives (FN) are shown by the number of FP false positives and FN false negatives, respectively.

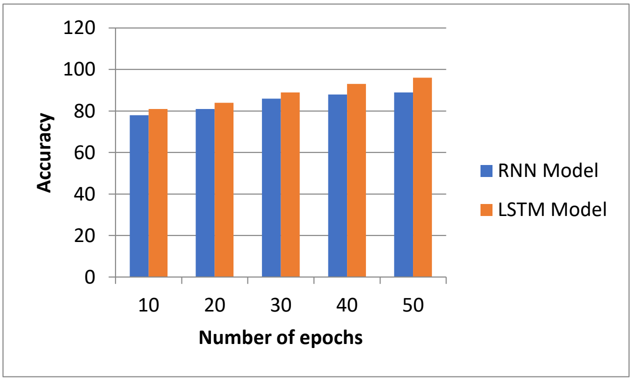

Fig. 2. Accuracy for proposed LSTM

The accuracy of a model is a metric that measures how well it performs across all classes. Accuracy can be calculated by dividing the number of correct guesses by the total number of forecasts.
Accuracy= (TP+TN)/(TN+FP+TP+FN)

The graph compares the existing RNN model and the proposed LSTM model. The existing model fails to produce results based on the large trained memory. Proposed model contains a memory unit which can store large amount of trained data & it can give improved accuracy while the quantity of epochs rises.

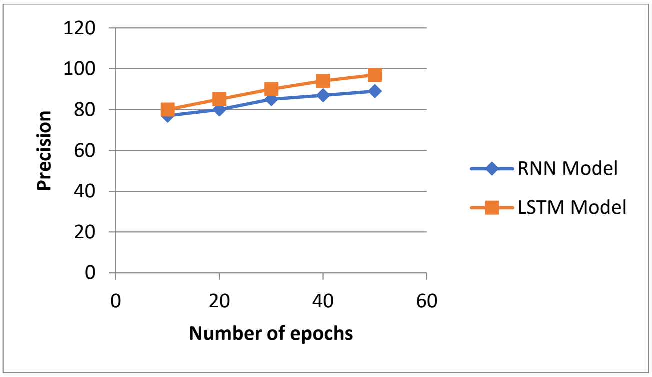

Fig. 3. Precision for proposed LSTM

Here Figure 3 represents the classification of the Number of epochs and Precision. The graph compares the existing RNN model and the proposed LSTM model. Precision can be calculated as ratio of number of true positives and number of positive predictions.
Precision = TP/(TP+FP)

The graph compares the existing RNN model and the proposed LSTM model. The existing model fails to memorise the past memory of trained data. The proposed model can recall great amount of memory it can give outputs with good accuracy.

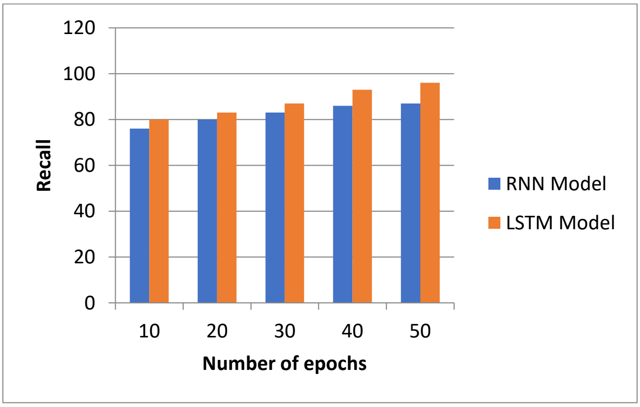

Fig. 4. Recall for proposed LSTM

Here Figure 4 represents the recall for the classification of several epochs and beginning samples. The graph compares the existing RNN model and the proposed LSTM Model. RNN model fails to give better recall of storage and benign examples. But the LSTM model contains a memory unit that can give better recall while the number of epochs increases.
Recall=TP/(TP+FN)

The graph compares the existing RNN model and the proposed LSTM model. The RNN model fails to store more memory and the RNN has a vanishing gradient. The LSTM model contains a memory unit it can give improved accuracy while the quantity of epochs rises.

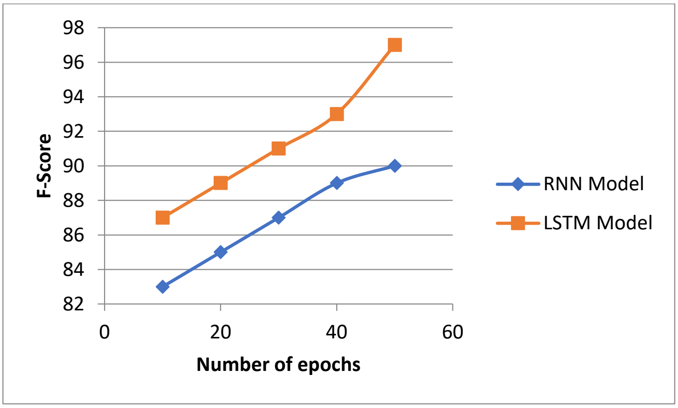

Fig. 5. F-Score for Proposed LSTM

Here figure 5 represents the classification of F-Score and Proposed LSTM. The F-Score is commonly known as F1-Score, is a metric used to analyse how accurate a model is and a dataset. It's used to evaluate a binary categorization system that categories examples into positive and negative groups.

F-Score=2(precision*recall) / (precision +recall)*

The graph compares the existing RNN model and the proposed LSTM model. The existing model becomes very tedious when producing results based on large dataset., whereas the proposed model overcome that tedious working by remembering long sequence trained data.

5. Conclusion

Everyone frequently text to each other and notice that anytime they try to enter a message, a recommendation pops up, attempting to guess the next word they want to type. Word Forecasting is the task of estimating words that are expected to follow the given text. It should also be emphasized that the text corpus's content has a significant impact on the outcomes. This system is helpful for users because it can increase the type speed and exclude errors. It effectively works on the larger datasets. To predict the good results, the model can be trained with more data which will re-evaluate the weights to understand the text corpus. The constructed LSTM model produces an average accuracy of 70-80%. This system is best suitable when predictions have to be done on a previously trained large amount of data. But the system likely to have the issue of overfitting and it takes long time & huge memory to train the model. This not much suitable for the small datasets.

References

[1] Shakhovska, K., Dumyn, I., Kryvinska, N. and Kagita, M.K., 2021. An Approach for a Next-Word Prediction for the Ukrainian Language. *Wireless Communications and Mobile Computing*, *2021*.

[2] Yang, J., Wang, H. and Guo, K., 2020. Natural Language Word Prediction Model Based on Multi-Window Convolution and Residual Network. *IEEE Access*, *8*, pp.188036-188043.

[3] Sharma, R., Goel, N., Aggarwal, N., Kaur, P., and Prakash, C., 2019, September. Next word prediction in Hindi using deep learning techniques. In *2019 International Conference on Data Science and Engineering (ICDSE)* (pp. 55-60). IEEE.

[4] Barman, P.P. and Boruah, A., 2018. An RNN based Approach for next word prediction in Assamese Phonetic Transcription. *Procedia computer science*, *143*, pp.117-123.

[5] Sarwar, S.M., 2016, May. Next word prediction for phonetic typing by grouping language models. In *2016 2nd International Conference on Information Management (ICIM)* (pp. 73-76). IEEE.

[6] Haque, M., Habib, M., & Rahman, M. (2016). Automated word prediction in Bangla language using stochastic language models. *arXiv preprint arXiv:1602.07803*.

[7] Spiccia, Carmelo, Agnese Augello, Giovanni Pilato, and Giorgio Vassallo. "A word prediction methodology for automatic sentence completion." In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 240-243. IEEE, 2015.

[8] Asnani, Kavita, Douglas Vaz, Tanay PrabhuDesai, Surabhi Borgikar, Megha Bisht, Sharvari Bhosale, and Nikhil Balaji. "Sentence completion using text prediction systems." In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, pp. 397-404. Springer, Cham, 2015.

[9] Surabhi, M. C. (2013, July). Natural language processing future. In *2013 International conference on optical imaging sensor and security (ICOSS)* (pp. 1-3). IEEE.

[10] Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. "LSTM neural networks for language modeling." At the *Thirteenth annual conference of the international speech communication association*. 2012.

[11] Rosa, J. L. G. (2002, October). Next word prediction in a connectionist distributed representation system. In *IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, pp. 6-pp). IEEE.