Enhancing Bangla Language Next Word Prediction and Sentence Completion through Extended RNN with Bi-LSTM Model On N-gram Language

Preprint · April 2024

DOI: 10.13140/RG.2.2.33321.81762

CITATIONS
0

READS
79

3 authors:

Md. Robiul Islam
Uttara University
5 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Al Amin
American International University-Bangladesh
15 PUBLICATIONS **29** CITATIONS
SEE PROFILE

Aniqua Nusrat Zereen
BRAC University
12 PUBLICATIONS **47** CITATIONS
SEE PROFILE

2024 3rd International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), 25-27 April, 2024, Gazipur, Bangladesh.

Enhancing Bangla Language Next Word Prediction and Sentence Completion through Extended RNN with Bi-LSTM Model On N-gram Language

Md Robiul Islam[1], Al Amin[2], Aniqua Nusrat Zereen[3]
[1,2] Department of Computer Science and Engineering, Uttara University, Dhaka, Bangladesh
[3] Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh
{robiul.cse.uu, alaminbhuyan321, aniqua.zereen}@gmail.com

Abstract—Texting stands out as the most prominent form of communication worldwide. Individual spend significant amount of time writing whole texts to send emails or write something on social media, which is time consuming in this modern era. Word prediction and sentence completion will be suitable and appropriate in the Bangla language to make textual information easier and more convenient. This paper expands the scope of Bangla language processing by introducing a Bi-LSTM model that effectively handles Bangla next-word prediction and Bangla sentence generation, demonstrating its versatility and potential impact. We proposed a new Bi-LSTM model to predict a following word and complete a sentence. We constructed a corpus dataset from various news portals, including bdnews24, BBC News Bangla, and Prothom Alo. The proposed approach achieved superior results in word prediction, reaching 99% accuracy for both 4-gram and 5-gram word predictions. Moreover, it demonstrated significant improvement over existing methods, achieving 35%, 75%, and 95% accuracy for uni-gram, bi-gram, and tri-gram word prediction, respectively.

Index Terms—Bangla word prediction, Bangla sentence completion, LSTM, Bangla language.

I. Introduction

In the modern era, the ubiquity of communication devices is indisputable. However, their intrinsic dependence on laborious and time-intensive text input procedures constitutes a noteworthy drawback. In response to these limitations, integrating textual word prediction systems has emerged as a promising remedy. Such systems mitigate the challenges associated with repetitive typing, efficaciously streamlining the overall text input process. By utilizing the predictive abilities embedded within text prediction systems, the need for manual input of text is eliminated. As an alternative, the most popular word suggestions are shown to the user, maximizing both user enjoyment and efficiency. This advanced forecasting procedure creates a smooth and user-friendly text entry interface, significantly improving the speed and fluidity of digital communication [1]. This clever application significantly lowers the number of keystrokes needed for word entry, which helps to mitigate writing obstacles. The effectiveness of the result is directly correlated with the number of keystrokes that the user saves, since this lowers the amount of time invested as well as the effort involved in producing text [2]. Once a word or phrase is entered, the program automatically creates a curated list of possible lexical possibilities. When the user finds the word they want in this list, selecting it is a simple process that only requires one click to insert the selected word into the page [3]. Scholars have conducted numerous investigations to recognize the benefits that come with word prediction systems, using a range of approaches to improve word prediction algorithms and enhance user experience in general. Specifically, within the parameters of their research project, they have diligently investigated and applied various methods to enhance the effectiveness and user-friendliness of word prediction systems. Partha et. al. utilized a combination of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) to predict the next word in the Assamese language. For untranslated Assamese text, they obtained an accuracy of 88.20%, whereas for phonetically transcribed Assamese language, they acquired a 72.10% accuracy [4]. The model presented by the Steffen Bickel et. al. accurately predicts the most likely word that will be typed next [5]. They used weather forecasts, food recipes, call center communications, and personal emails to assess their model. They modified N-gram language models to predict the following words to evaluate the model's performance, placing more emphasis on precise prediction than conventional performance indicators. With a focus on the English language, our study has investigated various word prediction methods for other languages. Nevertheless, few thorough investigations have been conducted with the goal of creating word prediction models for the Bangla language or sentence prediction [6]. The model's efficacy is intricately tied to the quality and diversity of the dataset utilized during the training phase. To this end, a comprehensive dataset has been meticulously curated from diverse sources, including bdnews24 [7], Prothom Alo [8], and BBC Bangla news [9]. Despite the considerable research on enhancing word prediction capabilities within the Bengali language using machine learning methodologies, persistent advancements are imperative to augment system efficiency and precision. The ongoing evolution of these systems hinges upon their capacity to adeptly handle expansive datasets, thereby

979-8-3503-8828-2/24/$31.00 ©2024 IEEE

facilitating more nuanced and accurate prognostications.

The comprehensive allowance of this research work is -

- No previous research has utilized this technique for Bangla language word prediction.
- Utilizing a large dataset for Bangla word prediction, significantly improves the accuracy of our suggested approach.
- The effectiveness of our suggested approach exceeds that of other approaches employed for Bangla word prediction and sentence completion.

The remaining sections of this research are structured like this: part II discusses the previous researcher's work, part III outlines the methodology encompassing dataset creation and implementation with pre-processing, part IV presents the experimental result and part V discussed the constraint and upcoming directions.

II. Related Works

Predicting the next word is a big focus in Natural Language Processing (NLP) research. In this context, discerning the following word presents a multifaceted challenge. A notable obstacle arises from the inherent nature of machines, which predominantly interpret information through ASCII values, essentially binary code represented as 0s and 1s. The need to represent linguistic elements in numerical form becomes imperative. Researchers have diligently addressed these challenges associated with the next-word prediction. Their efforts have been directed towards finding effective solutions to the intricacies posed by machine comprehension limited to binary representations. They aim to transform words into numerical entities, paving the way for more accurate and sophisticated next-word predictions in natural language scenarios. The majority of the work has been LSTM based. To tackle predicting the following word in sequential information, Mikolov et al. suggested a language model using RNN [10], tasks that were previously challenging for RNN to solve were solved by Felix et al. using LSTM [11]. Sutskever et al. used an RNN model to predict the next word in a sequence of data. Their recently presented model was a development of RNNSearch [12], and it was trained from a question-answering model to a language model via backpropagation. Zhou et al. proposed the C-LSTM model a while back, which combines RNN with LSTM for classification and prediction on phrase representations [13]. A method for creating Bengali text using a sequence paradigm was presented by Banik et al. [14]. Another approach was presented by Abujar et al. [15], who employed bi-directional RNN to generate Bengali text In their study, Barman et al. employed an RNN-based methodology to develop a next-word prediction model [4]. In essence, they created a model of the LSTM, a unique kind of RNN. They used their work to suggest the subsequent word with the highest likelihood reliably next. The N-gram has been used to reduce the prediction time while typing in the Kurdish language, and the model was successful in prediction. This model has been developed in the R programming language. The maximum accuracy recorded in this model is 96.3% [16].

III. Methodology

This section describes the proposed methodology as well as the operating principles of the machine learning algorithms. The Bi-LSTM model predicted the next word and sentence in Bangla. Fig. 1. illustrates the full process of the methodology.

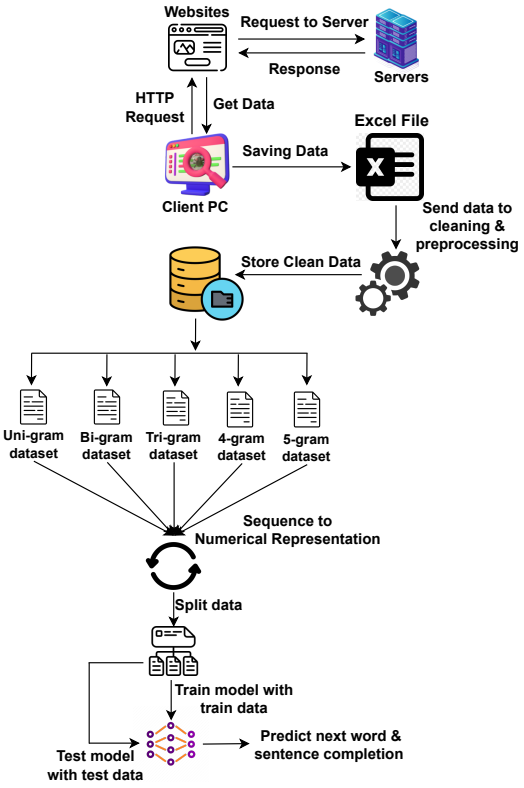

Fig. 1. Proposed methodology

We began by introducing the dataset in subsection A, followed by data preprocessing techniques in subsection B, and culminated in the model implementation details in subsection C.

A. Dataset Outline

To evaluate the proposed approach, we assembled a large corpus of Bangla text, totaling 1.7 GB, gathered from diverse sources. Table I provides a statistical overview of the collected data.

TABLE I
Data Summary

Data Source	Distinct words	Total
Bdnews24 [7]	45789	3892651
Prothom Alo [8]	97000	2087916
BBC News Bangla [9]	67004	4045683

B. Data Preprocessing

After collecting the data, we used several functions to remove unwanted commas, numbers, etc. from our dataset and five distinct datasets were generated from the cleaned dataset: Unigram, Bigram, Trigram, 4-gram, and 5-gram. Fig. 2. effectively illustrates the data-cleaning workflow and the creation of different n-gram datasets. Language models, known as N-grams, are statistical models that calculate the likelihood of a specific word sequence occurring in a language. In this study, we have used n-gram language models to capture the sequential relationships found in Bangla text [17]. When attempting to predict the next word or words, the quantity of input words can typically vary from one attempt to the next. One word, a 1-gram or uni-gram, is produced whenever there is only one input word [18]. Bi-gram is used when the input consists of two words, and the output is one word [18]. Furthermore, a Three-gram, sometimes known as a Tri-gram, is created when three words are input and one word is output [18], and analogously for Four-gram and Five-gram. Usually, the last four or five words are enough to grasp the sequence's dependency. To better understand n-gram language models, consider an example of a Bangla sentence. Analogous to uni-grams and bi-grams, tri-grams, and further n-grams maintain the pattern of taking multiple input values and generating a single output value, demonstrating a consistent approach to sequence processing.

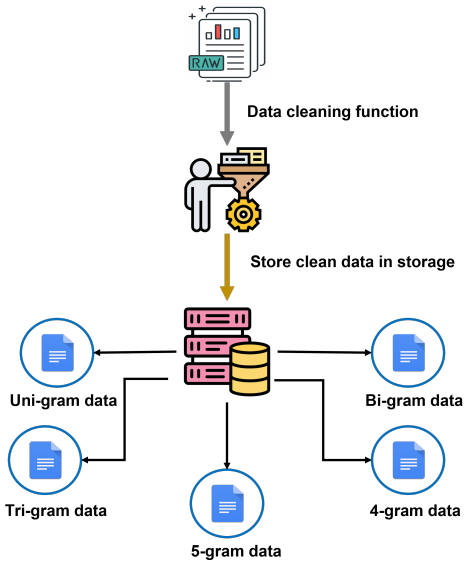

Fig. 2. Data Preprocessing Workflow

Table II and Table III showing the Bi-gram and Tri-gram models. It shows how it predicts the next word based on the inputted word. For Bi-gram, only two words are used to predict the next word; in the case of Tri-gram, there are three words used to predict the next word.

TABLE II
Bi-gram Model Data Sample

Input_1	Input_2	Output
বাংলাদেশ	একটি	সমৃদ্ধ
একটি	সমৃদ্ধ	সংস্কৃতির
সমৃদ্ধ	সংস্কৃতির	দেশ

TABLE III
Tri-gram Model Data Sample

Input_1	Input_2	Input_3	Output
বাংলাদেশ	একটি	সমৃদ্ধ	সংস্কৃতির
একটি	সমৃদ্ধ	সংস্কৃতির	দেশ

C. Implementation

The n-gram language model determines the likelihood of each potential next word and chooses the one with the greatest likelihood to be the next word or words. The utilization of a probabilistic viewpoint enables us to determine the probability of the next value, empowering us to predict the most probable next value based on the input value ক্রেমলিনে ড্রোন, equation (1) has been used to get the highest likelihood of the next word.

Word after that = Max (P (হামলার | ক্রেমলিনে ড্রোন, P (ঘটনাটি | ক্রেমলিনে ড্রোন), P (আসলে | ক্রেমলিনে ড্রোন ---------- (1)

Due to the frequent emergence of zero probability, the model cannot recommend the next word with the highest probability, leading to the failure of the entire method in predicting the most suitable value. The effectiveness of n-gram models diminishes when confronted with large datasets featuring lengthy input sequences or a considerable number of n-grams. To alleviate the impact of data sparsity, Back-off, and Katz Back-off techniques were employed to refine the probability distribution, enabling the effective management of n-grams with low counts [19]. RNNs keep updating their memory with past information to maintain a consistent understanding.

Long sequences cause RNNs to lose the impact of previous layers, which gives rise to the vanishing gradient problem. Two well-known gating mechanisms, LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) were used in response to the vanishing gradient problem in RNN. These mechanisms allow the network to efficiently capture long-term dependencies in sequential data. The use of three gating mechanisms—input, forget, and output—by Bi-LSTM in this study justifies its adoption since they improve the model's capacity to learn from and retain information from distant past inputs, hence resolving the drawbacks associated with vanishing gradients in RNN [tweentyone]. GRU employs update and reset gates to handle sequential data, whereas LSTM's ability to maintain internal memory over time makes it more effective for processing complex datasets compared to GRU, which doesn't have a specialized

forget gate.

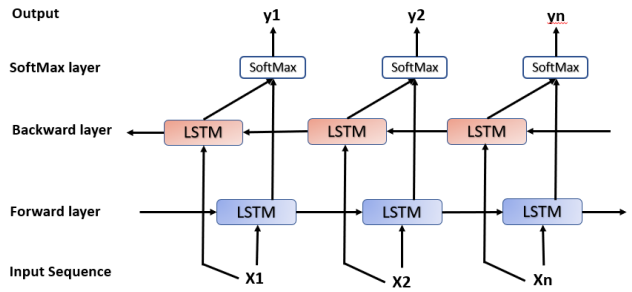

Fig. 3. Structure of Bi-LSTM recurrent neural networks

Two separate recurrent neural networks (RNNs) are integrated to create bidirectional RNNs. This setup lets the network look at information from both directions (like before and after) for each point in the sequence. When employing bidirectional processing, the input data is traversed in two directions: from past to future and from future to past. The key distinction from unidirectional RNNs lies in using a bidirectional long short-term memory (Bi-LSTM) structure, where the backward LSTM retains information from the future. By combining the hidden states from the forward and backward LSTMs, the model can capture contextual information from the sequence's preceding and succeeding elements. This bidirectional approach, exemplified by Bi-LSTMs, yields notable performance improvements as it enhances the network's understanding of contextual nuances. Fig. 3 showing the Bi-LSTM layer architecture.

Fig. 4. shows the architecture of the trained models with five hidden layers named Embedding, Bi-LSTM Layer1, Bi-LSTM Layer2, Fully Connected Layer1, Fully Connected Layer2. For Bi-LSTM layers we used 100 LSTM units with $\tanh$ activation function and for Fully Connected Layer1 we used ReLU and for Fully Connected Layer2 we used $\mathrm{softmax}$ activation function. In the Embedding layer v denotes input dimension or vocabulary length, o represent the output dimension and i represent the input length.

D. Word Prediciton

Upon effectively completing training on all five datasets (U-gram to 5-gram), we have obtained five unique trained models optimized for processing input of varying lengths.

These models take in sequences of words of different lengths and guess what word is likely to come next after the input sequence. When the input sequence contains only one word, the Uni-gram model is utilized for training. With a single-word input as its input, the model will most likely anticipate the next word. Systematically, In the case of two input words, the inputted sequence is directed to the trained Bi-gram model, designed to handle two-word inputs and predict an output word. Similarly, for the other trained models. Fig. 5. demonstrates the word prediction methodology for input sequences of different lengths using

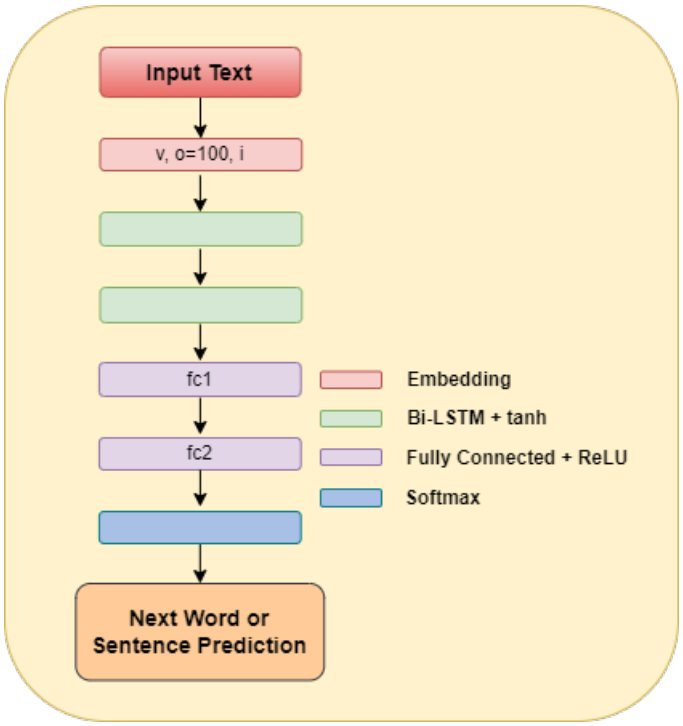

Fig. 4. The proposed Bi-LSTM model architecture

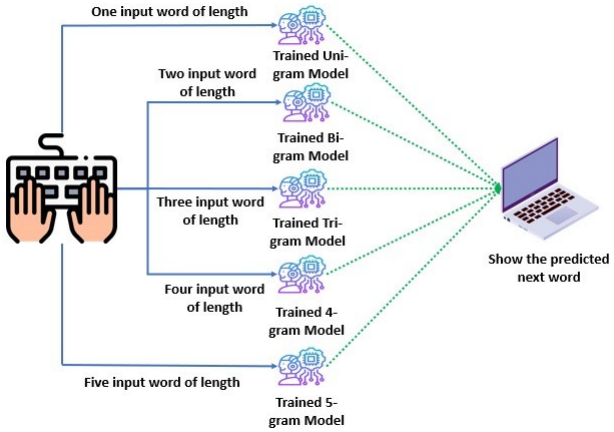

Fig. 5. Next word prediction from the model

five distinct trained models. In cases where the input word sequence exceeds five words, the 5-gram model relies solely on the last five words to predict the next word. The last five words typically provide enough context to determine the next word.

E. Sentence Prediction

Our innovative model combines N-gram training with Bi-LSTM-based RNNs. It iteratively predicts and appends the next word to the input sequence, capturing evolving context for accurate predictions. The process ends upon encountering sentence-ending punctuation. In Bangla, sentence

termination is signified by the symbols "|" for declarative statements and "?" for interrogative statements. Once the model detects an end punctuation mark, it ceases prediction and outputs the complete sentence, representing a cohesive and meaningful continuation of the given word sequence. Fig. 6. shows the word and sentence prediction by the proposed model.

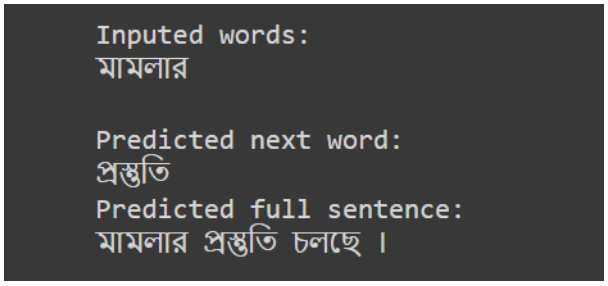

Fig. 6. Next word prediction and sentence completion

IV. Result Evaluation

We have evaluated our proposed approach using a corpus dataset, training five different models with identical configurations for up to 300 epochs. Fig. 7. depicts the accuracy across epochs, while Fig. 8. showcases the loss over the same epochs. Our 4-gram and 5-gram models exhibit an average accuracy of 99% and 99.74%, respectively, accompanied by average losses of 2.04% and 1.11%.

TABLE IV
Comparative Analysis

Reference	Accuracy
Barman et. al. [4]	88.20%
Hamarashid et. al. [16]	96.30%
Proposed Model	99.00%

Table IV shows the comparison between Barman et. al. [4] and Hamarashid et. al. [16] and our proposed system. In Author [16], they used Tri and a Hybrid Approach of Sequential LSTM and N-gram; their accuracy was 88.20%. On the other hand, we have a maximum efficiency of 99% for high-order sequences.

V. Conclusion

The word prediction and sentence completion model shows promising accuracy based on the provided dataset. Using multiple pattern-discovery techniques is essential for effectively removing noisy data in NLP. Next-word prediction is an example of an NLP application because it involves text mining. The Bi-LSTM model was used for forecasting over multiple epochs, resulting in a significant decrease in loss. Specific preprocessing steps and model modifications could be implemented to improve the model's prediction performance. We will make our data set public so that researchers can conduct research in the near future. A more diverse dataset will improve our work.

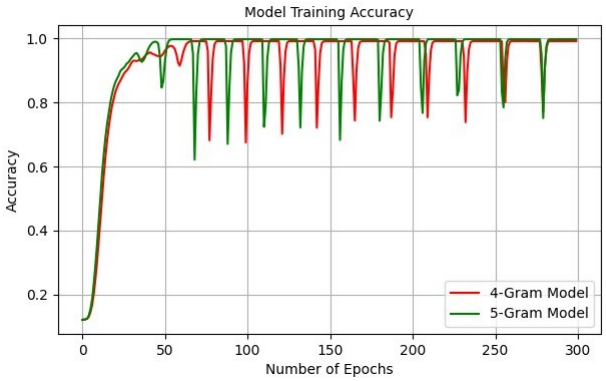

Fig. 7. 4-Gram and 5-Gram model accuracy per epochs

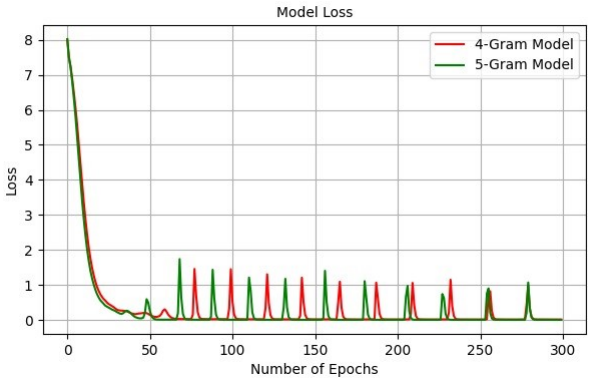

Fig. 8. 4-Gram and 5-Gram model loss per epochs

References

[1] What is Word Prediction. [Online]. Available: https://www2.edc.org/ncip/library/wp/What$_i s.htm$ [*Accessed* : 20/01/2024].

[2] Md. Masudul Haque, Md. Tarek Habib, and Md. Mokhlesur Rahman. "Automated Word Prediction in Bangla Language Using Stochastic Language Models". In: Computing Research Repository (2016). url: http://arxiv.org/abs/1602.07803.

[3] Md Habib et al. "An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction". In: International Journal of Intelligent Systems and Applications (2018), pp. 47–54. doi: 10.5815/ijisa.2018.02.05.

[4] Partha Barman and Abhijit Boruah. "A RNN based Approach for next word prediction in Assamese Phonetic Transcription". In: Procedia Computer Science (2018), pp. 117–123. doi: 10.1016/j.procs.2018.10.359.

[5] Steffen Bickel, Peter Haider, and Tobias Scheffer. "Predicting Sentences using N-Gram Language Models." In: 2005. doi: 10.3115/1220575.1220600.

[6] Nestor Garay-Vitoria and Julia Abascal. “Text prediction systems: a survey”. In: Universal Access in the Information Society (2006), pp. 188–203. doi: 10.1007/s10209-005-0005-9.
[7] Bdnews24.com. [Online]. Available: https://bangla.bdnews24.com/ [Accessed: 25/01/2024].
[8] Prothom Alo. [Online]. Available: https://www.prothomalo.com// [Accessed: 26/01/2024].
[9] BBC Bangla. [Online]. Available: https://www.bbc.com/bengali [Accessed: 28/01/2024].
[10] Tomas Mikolov et al. “Recurrent Neural Network Based Language Model”. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH (2010), pp. 1045–1048. doi: 10.21437/Interspeech.2010-343.
[11] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to Forget: Continual Prediction with LSTM”. In: Neural Computation (2000), pp. 2451–2471. doi: 10.1162/089976600300015015.
[12] Ilya Sutskever, James Martens, and Geoffrey Hinton. “Generating Text with Recurrent Neural Networks”. In: Proceedings of the 28th International Conference on Machine Learning, ICML (2011), pp. 1017–1024.
[13] Chunting Zhou et al. “A C-LSTM Neural Network for Text Classification”. In: 2015. url: http://arxiv.org/abs/1511.08630.
[14] Nayan Banik et al. “Bangla Text Generation System by Incorporating Attention in Sequence-to-Sequence Model”. In: World Journal of Advanced Research and Reviews (2022), pp. 80–094. doi: 10.30574/wjarr.2022.14.1.0292.
[15] Sheikh Abujar et al. “Bengali Text Generation Using Bi-directional RNN”. In: 2019. doi: 10.1109/ICCCNT45670.2019.8944784.
[16] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. In: Neural Computation (1997), pp. 1735–80. doi: 10.1162/neco.1997.9.8.1735.
[17] BBC Bangla. [Online]. Available: https://web.stanford.edu/ jurafsky/slp3/3.pdf [Accessed: 26/03/2024].
[18] Understanding GRU Networks - Towards Data Science. [Online]. Available: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9b [Accessed: 15/02/2024].
[19] How the LSTM improves the RNN. [Online]. Available: https://towardsdatascience.com/how-the-lstm-improves-the-rnn-1ef156b75121 [Accessed: 13/03/2024].