# Word Completion and Sequence Prediction in Bangla Language Using Trie and a Hybrid Approach of Sequential LSTM and N-gram

**Conference Paper** · November 2020

DOI: 10.1109/ICAICT51780.2020.9333518



Soumik Sarker
Shahjalal University of Science and Technology



Jillur Rahman Saurav
Shahjalal University of Science and Technology



Md Mahadi Hasan Nahid
Shahjalal University of Science and Technology

2nd ICAICT, 28-29 November 2020, Dhaka, Bangladesh.

# Word Completion and Sequence Prediction in Bangla Language Using Trie and a Hybrid Approach of Sequential LSTM and N-gram

Soumik Sarker
*Dept. of Computer Science and Engineering*
*Shahjalal University of Science and Technology*
Sylhet-3114, Bangladesh
ronodhirsoumik@gmail.com

Md. Ekramul Islam
*Dept. of Computer Science and Engineering*
*Shahjalal University of Science and Technology*
Sylhet-3114, Bangladesh
ei.ekramul.islam@gmail.com

Jillur Rahman Saurav
*Search Engine Pipilika*
*Shahjalal University of Science and Technology*
Sylhet-3114, Bangladesh
sauravsust71@gmail.com

Md Mahadi Hasan Nahid
*Dept. of Computer Science and Engineering*
*Shahjalal University of Science and Technology*
Sylhet-3114, Bangladesh
nahid-cse@sust.edu

*Abstract*— **Autocompletion and sequence prediction is the basis of any assistance systems. When we step out to type something, it is much comforting to get a suggestion of the next word or even the full sentence before we type, which saves keystrokes of typing and reduce misspelling. Despite the several promising works in English language, little prior research in Bangla has shed light on this domain. In this paper, we proposed an integrated methodology of trie, sequential LSTM and N-gram for word completion and sequence prediction in Bangla language. The trie data structure was implemented to store Bangla vocabulary and retrieve the word from user-inputted prefix. For sequence prediction, we explored a hybrid approach of neural network and N-gram. This collaboration of sequential LSTM and N-gram reveals a better performance than any single model implementation. We evaluated this model with both small and large-scale Bangla datasets for better efficiency. The experiments show a promising outcome of our hybrid approach for word completion and sequence prediction. We believe that our framework leads to a profound impact on Bangla search engines, keyboards, and further researches based on recommendation systems.**

***Keywords—Sequence Prediction, Autocompletion, Trie, NLP, N-gram, Sequence-to-sequence Model, Encoder-decoder, LSTM, Neural Network, Bangla Word Completion***

## I. INTRODUCTION

In query related systems or recommendation-based interface, auto-completion, and sequence prediction can be a core feature to incorporate the system with user interaction [1]. It betters the user interaction by guessing the next word(s) that the user wants to type.

Sequence prediction means guessing the next word(s), even the whole sentence from the previous word sequence. To be more specific, sequence prediction is a process that deals with predicting the next value or values in the sequence based on preceding values. This sequence can be letters or words of a sentence. Auto-completion helps the user to complete the word, which decreases the misspelling. Auto-completion and sequence prediction is supposed to be a feature of better UX design for software. Sequence prediction falls under supervised machine learning problem. There are several approaches for the betterment of the solution. Previous studies on this domain have shed light, yet the prediction-based system on Bangla Language does not reach that satisfactory level. Today we have thousands of recommendation or Q/A based software and applications using Bangla with lesser efficiency of autocompletion or user context prediction. With the increasing focus of natural language processing in Bangla and its implementation, this domain catches attention. Some research works propose the implementation of N-gram [2] with deleted interpolations, and backoff model [3] for Bangla word prediction. Some studies also focused on implementing Neural Network for Bangla word prediction. The N-gram model can be fit for word prediction. But to be more efficient and suited for various types of dataset, Neural Network approaches have been implemented in this work. In this paper, a novel approach has been proposed to find a better suggestive model using a collaborative approach of N-gram and sequential LSTM model. The whole framework can be divided into two parts-

*1)* ***Word Completion:*** In this phase, we implemented a dictionary of Bangla words using a trie structure.

*2)* ***Sequence Prediction:*** For the sequence prediction phase, we implemented a hybrid model, a combination of n-gram and sequence-to-sequence LSTM model. The Seq2seq LSTM model predicts some probable sequences. Then the N-gram part of the hybrid model sorts out the best probable sequence among the results of the LSTM model.

Through reviewing some previous works on Bangla sequence prediction, we found that most of these works are focused on the use of N-gram for next word prediction. The rationale for using the LSTM module with N-gram is because LSTM is a very powerful and adaptive model [24]. Also, for unseen cases, it suits very smoothly and gives a satisfactory outcome. Several implications of the proposed framework can be on any autocompletion-based feature in Bangla, keyboards, chatbots, recommender systems, Q/A based systems, etc.

## II. RELATED WORK

Neural Network models have been used for generating sequences in various sectors, like poem or music generation [4], handwriting recognition [5] image captioning [6], and most importantly for query suggestion, information retrieval [7]-[8]. Before the implementations of Recurrent Neural Networks (RNNs) and LSTMs in particular, n-gram models play a significant role in word or sequence prediction. In this work, authors developed an evaluation metric and implemented N-gram language model to the problem of predicting particular words, given an initial text fragment [9]. Another significant work [10] on sequence prediction

978-0-7381-2323-3/20/$31.00 ©2020 IEEE

proposed a probabilistic model using Prediction Suffix Trees (PSTs). Using this data structure, authors tried to extend the notion of PST to unbound vocabularies of the language, word, or the line that the user wants to type.

As the implementations of Neural networks outperform the n-gram models with faster classification rate and adaptive capabilities [11], several neural network-based models have been proposed for sequence prediction. In this study [12], the authors proposed a proactive system that automatically recommends user-relevant background information using long short-term memory (LSTM) for prediction. In [13] paper, they suggested a novel loss framework, comprising of two closely linked features for the betterment of the conventional classification framework. One of these is classical cross-entropy with an additional optimizer for KL-divergence and another is estimated target distribution based on word embedding space. Another paper on Assamese Phonetic Transaction [14], proposed an RNN based approach for next word prediction. Sometimes, inappropriate suggestions appear on web search. This work [15] sheds light on detecting these irrelevant query suggestions using convolutional bi-directional LSTM.

In Bangla natural language processing, several approaches have been shed light on word completion and prediction problems. In [3], they used statistical prediction techniques as n-gram, backoff propagation, deleted interpolations on large data set of Bangla newspaper for word prediction. Another paper on Bangla autocomplete approach [2] proposed a novel approach using n-gram to find optimum language model for word(s) prediction. As an implementation of neural network, these studies proposed Bangla sequence generator using LSTM [16]-[17]. A work on Bangla word prediction along with sentence prediction [18] implemented GRU (Gated Recurrent Unit) based RNN on n-gram based on n-gram dataset. So, reviewing all these implementations, we try to incorporate LSTM and N-gram for better sequence prediction. We use the Trie data structure for word completion and correction. For sequence prediction, we proposed a hybrid model of LSTM and N-gram.

## III. METHODOLOGY

### A. Proposed Methodology

In this paper, we proposed a trie model for word auto-completion and a hybrid model of n-gram and sequential LSTM for sequence prediction. As mentioned earlier, we divided our model into two parts. One is word completion and correction. For word completion, we implemented the Trie dictionary and pushed our vocabulary to the dictionary. Another one is the sequence prediction part of the model. For this, we used a combination of LSTM and N-gram. Though, the prediction system is mostly dependent on the LSTM part. Here we use the sequence-to-sequence model for sequence generation. The model generates a list of suggested sequences and then we use the n-gram model to sort out the best-predicted one from the list.

We have begun to evaluate our model with a small-scale dataset. The dataset was prepared from "SHADHASHIDHE KOTHA", a column of a popular writer, Dr. M Zafor Iqbal from Bangla Tribune news portal [19]. We have first extracted all the vocabulary from the corpus. We push these words to our trie model for word completion and correction. Also, another dataset has been prepared for the sequential model. We have extracted all the sentences, added starting and ending markers, and then broken into parts. For example, {START_ আমি যাব _END} is broken into {START_, আমি যাব _END}, {START_ আমি, যাব _END} and {START_ আমি যাব, _END}. These preceding and succeeding sequences of the sentences are treated respectively as *x_train* and *y_train* for the encoder-decoder. And also, for the n-gram, we have used those sentences with markers. Using these datasets, we have got good performance. But we wanted to make it in a more implementable way to prove its efficiency. So, a large-scale dataset has been prepared based on various queries, scraped from several news blogs like Prothom Alo, Kaler Konto, Bangla Tribune, etc. As the dataset is more condensed with query-based corpus, we also scraped this part from the Q/A section of several admission sites.

We prepared the trie model in such a way so that it can push new words itself. If it gets any new one that is not in the tree, then it first checks whether the word is right or wrong through a prebuilt checker. If that is right, then the model pushes the word to the tree.

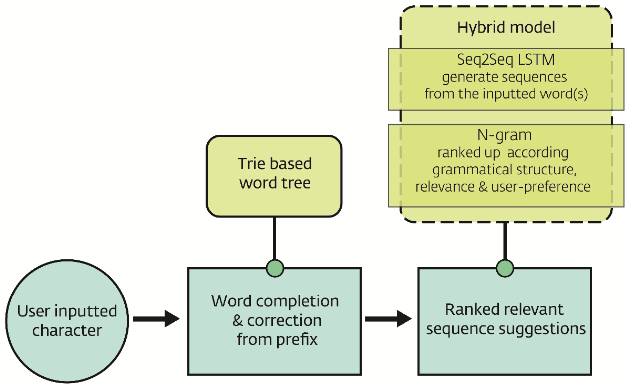


Fig. 1. Overall diagram of the proposed model

### B. Logistics behind The Proposed Method

Three base implementations of our system are Trie, N-gram, and the LSTM model. Trie is an efficient data structure to retrieve a word from the user given prefix. It is one of the simplest and fastest data structures for inserting, searching, and deleting words. The worst case runtime of creating a trie is $O(mn)$, m is the length of the longest key in the trie and n is the total number of keys [20]. Thus it is better than using hash tables.

We used sequence-to-sequence LSTM model for sequence prediction. The strength of the model over n-gram is, it can map sequences of different lengths, which is most important for sequence prediction [21]. Moreover, neural network has the power to suit any dataset and be trained [22]-[23]. Now it may seem confusing or redundant about the use of n-gram. As usual, every recommendation system keeps track of users' chosen words. Using these, we trained our n-gram model, where it fits the most chosen words as most frequent. Thus from the most probable predictions of the LSTM model, it gives us the best of them, the most suitable one. As these frequently chosen word segments can be changing from time to time, we cannot solve that using LSTM. N-gram model is supposed to be significantly dependent on the training corpus. So, with the change of the frequently chosen world list, we lifted that concern on n-gram rather than focusing on LSTM. In this case, there is no chance of unseen n-gram or non-zero probability and thus a short possibility of sparsity issues.

## IV. Background

### A. Trie

Trie is a type of data structure, mainly used in information retrieval system. To be specific, trie is used to reduce search complexities at an optimal limit (key length). Using trie, we can search the key in $O(m)$ where $m$ is the maximum length of the string. As a tree-like structure, it keeps every unique character in every node. The root node is always supposed to be null. In our work, we push all possible vocabulary sets of Bangla language in the trie model so that it can complete any word from the user-given prefix and also correct if the user entered the word wrong. In our model, the trie is supposed to be dynamic. As vocabulary keeps increasing, it checks if that particular word exists in the structure. If not, then it approves the new data itself through some spelling check.

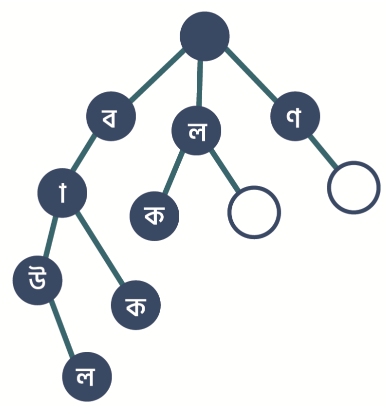


Fig. 2. Implementation of the Trie data structure for word completion.

### B. Sequential Model

An LSTM model aims to map the input sequence of any length with a fixed-length output, where the length of both may differ from each other. It is a widely used model for machine translation, speech recognition, and video captioning, text generation.

In this predictionary system, we use the sequence-to-sequence model [23] for predicting sequence from user given query input of any length. The main strength of our model is it can map sequences of different lengths to each other [21].

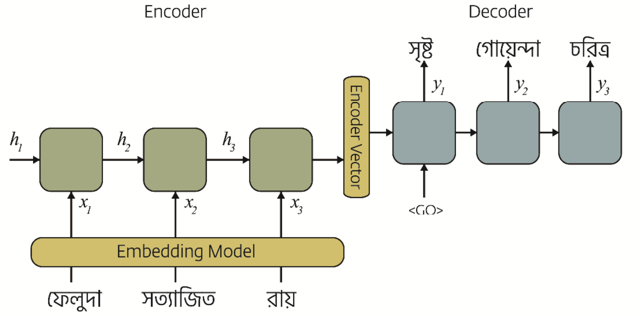


Fig. 3. Implementation of Seq2seq Encoder-Decoder model for sequence generation.

This model comprises three parts- encoder, intermediate vector, and decoder [24]. The encoder part is an LSTM of several units where each of them uses each unit of the input sequence and propagates it forward. In our system, the encoder part is fitted with sentences of several lengths. Each word of the sequence is represented as $x_i$ where $i$ is the order of the word on that sequence. The hidden $h_i$ state is computed using the formula-

$$h_t = f(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$$

Through several trial and errors, we can define the appropriate weight for accurate prediction. The intermediate vector encapsulates the information with each element on the hidden state produced from the encoder part. It acts as the initial state for the decoder part. The decoder part aims at making predictions. It is also an LSTM of several units. Here the decoder model has been fitted with the succeeding sequences of the sentences that we used in the encoder model. Each word of the sequence is represented as $y_i$ where $i$ is the order of the word on that sequence. The hidden $h_i$ state is computed using the formula-

$$h_t = f(W^{(hh)} h_{t-1})$$

The output at timestep is computed using-

$$y_t = softmax(W^S h_t)$$

### C. N-gram

N-gram model is a probabilistic language model. As a statistical language model, it approximates the probability of a word given the succeeding words by using the conditional probability. As the basic concept of n-gram, for limit history to fix number of words n, the general equation to the conditional probability of the next word in sequence $w_1$, $w_2$,........ $w_n$ is:

$$p(w_n \mid w_1^{n-1}) = p(w_n \mid (w_{n-N+1})^{n-1})$$

It can be uni-gram, bi-gram or tri-gram according to N=1, 2 or 3. For example, for the sentence আজ রাতে যাব, the tri-gram probability:

$$p(\text{আজ রাতে যাব}) = p(\text{আজ} \mid \langle s \rangle \langle s \rangle) \times p(\text{রাতে} \mid \langle s \rangle \text{ আজ}) \times p(\text{যাব} \mid \langle s \rangle \text{ আজ}) \times p(\text{যাব} \mid \text{আজ রাতে})$$

As explained earlier, we use n-gram in our hybrid model to narrow down the final suggestions list that results from the LSTM model. To be specific, we prepared the hybrid model so that we can get the top five or six possible predictions. Then we used the list as the input of the N-gram. As n-gram is based on frequency probabilistic model, it can result in the most suitable prediction for the user. If $s_1, s_2, s_3, s_4, s_5$ are the outputted sequences generated from the seq2seq model, then the task of the n-gram is to sort the sequences out in ascending order.

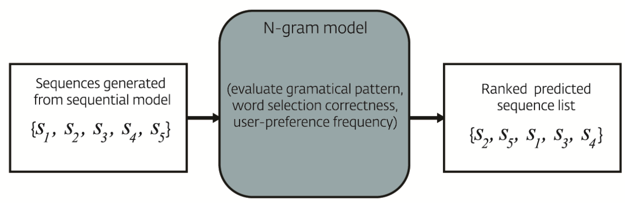


Fig. 4. Implementation of N-gram in our hybrid model.

## V. IMPLEMENTATIONS AND EXPERIMENTS

Word completion or auto-completion is one of the most common features of any recommendation system. Some of the recommendation systems use cached-based data for auto-completion. But there always remains a chance to take misspelled words as trained data. In this paper, a word completion model is recommended using trie data structure.

So, first we segregated all the vocabulary from our corpus and listed all the unique words. Then we pushed all the words to the trie. The trie has an empty node linked to other nodes. So, it is a tree with an empty root node and other child nodes of it, each containing a character. The worst-case runtime for creating a trie is $O(mn)$, where $m$ is the longest word in the vocabulary and $n$ is the number of the total words. The time complexity of searching, inserting, and deleting a word is $O(l)$, where $l$ is the length of the word [20]. In our dataset, we find the largest word is অঘটনঘটনপটিয়সী.

As previously explained, this trie approach is dynamic. So, for the unseen cases, the model first checks the word using spell checker API. If the word is correct, then it has to be assumed that the word is not in the node. So, the word will be pushed to the tree and thus added to the dictionary leaving no chance of misspelling.

Apart from the word completion and correction part, sequence prediction is the most vital segment of our proposed model. It predicts word(s), in fact, a sequence from the user given input sequence. Though many works have done on sequence prediction, we feel less compatible with work during our working on Bangla prediction.

As explained earlier, for sequence prediction, we categorized the model into two parts. First, we implemented the sequence-to-sequence LSTM model for generating sequences from user-given input. For three inputted sequence, the sequential model predicts at most 4-5 predicted sequences. Next, the n-gram model has been implemented for shortlisting the predictions in a ranked manner. Using the frequency of the sequences, the model ranks up these predicted sequences generated by the sequential model according to several criteria. The core focus of this framework is to make the prediction more accurate and also relatable. As the grammar of a language is changing and so do the choice of words, it is difficult to keep track of that through the sequential model. On the contrary, the n-gram model can be trained with that. It can keep track of the frequency of the word and its occurrence probability on a sequence based on the preceding word(s). Another way to track that using sequential model is to make it more attention-centric model. Then, with the change in frequency, we have set another weight to the layer. Changing the attention of the layer can impact the optimizer function and fail to give accurate predictions [25].

### A. Dataset

One of the main differences between Bangla and English language is, Bangla sentences have more grammatical diversity than English sentences. A sentence with the same words can represent various emotions according to the formation of the words. As explained earlier, we used two (small scale and large scale) datasets for the evaluation of the model. Table I represents a quick glimpse of the sources and patterns of the two datasets.

TABLE I. DATASETS

| Dataset | Source | Description |
|---|---|---|
| A | SHADHASHIDHE KOTHA by Dr. M Zafor Iqbal [19] | It is a narrative corpus with 1243418 Bangla words. Average word length is 13. |
| B | Scraped from several Bangla news portal like Prothom Alo, Bangla Tribune and also FAQ part of admission sites. | It is a query-based corpus having almost 3 million words and average word length is 9. |

This **dataset A** can validate that our system can work on auto-completion and sequence prediction. But to justify its efficiency, we prepared this **dataset B**. We partitioned the total dataset at the proportion of two-third as training dataset and one-third as testing dataset.

### B. Parameters

For trie structure, we extracted all words from the datasets in a set and pushed those words into the tree. We added the start and end markers to the sentences and trained the n-gram model. We used One-Hot Embedding model. But we were trying to explore better models like Glove, FastText to explore what suits best for Bangla. The most crucial part of predicting sequences was the sequential model. The number of cells in the encoder and decoder models is 256. In the decoder, we use softmax as the activation function. In model compilation, we used Adam as an optimizer function. We implemented Categorical Crossentrophy as the loss function as it seems best suited for softmax. We fed the dataset into the model till 750 epochs. For a sentence with n words (including staring and end markers), we split the sentence (n-1) times. Here is an example of the splitting:

TABLE II. DATASET SPLITTING FOR MODEL

| START_ আমি আমার দেশকে ভালোবাসি _END | |
|---|---|
| ***x_train*** | ***y_train*** |
| START_ | আমি আমার দেশকে ভালোবাসি _END |
| START_ আমি | আমার দেশকে ভালোবাসি _END |
| START_ আমি আমার | দেশকে ভালোবাসি _END |
| START_ আমি আমার দেশকে | ভালোবাসি _END |
| START_ আমি আমার দেশকে ভালোবাসি | _END |

As **dataset B** is larger than **dataset A**, the train and test datasets are carefully split to avoid overfitting and generalization error. We set the batch size to 128 and use k-fold cross-validation for estimating performance on new data.

## VI. RESULT ANALYSIS

### A. Experimental Results

In this section, we mainly evaluate our hybrid model for sequence prediction, especially the sequential model. As mentioned earlier, we used two datasets for the evaluation of our model. First, we fed the model with **dataset A** to 750 epochs. This model finished training by attaining a maximum of 84% accuracy. For **dataset B**, it completed training with a maximum of 81% accuracy. Fig. 5 and Fig. 6 respectively represent the accuracy and loss rate after every

50 epochs (from 200th - 700th epoch) of the first and second datasets.

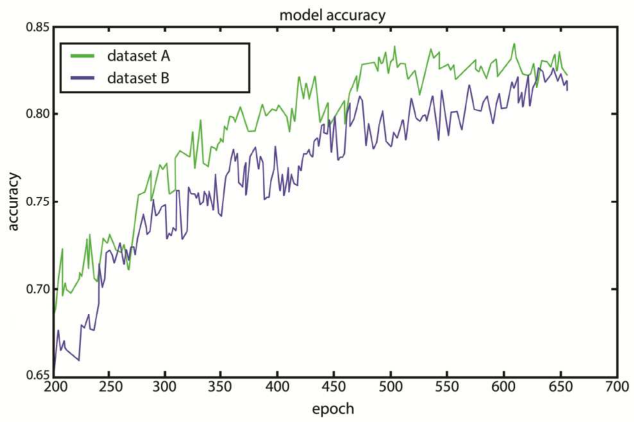


Fig. 5. Average accuracy results of training model for dataset A (small-scale) and dataset B (large-scale).

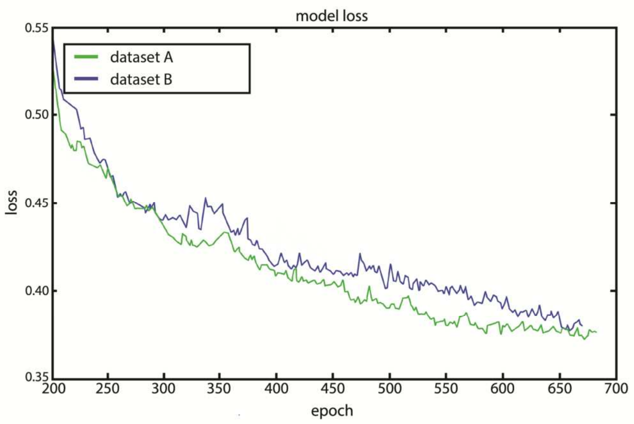


Fig. 6. Average loss results of training model for dataset A (small-scale) and dataset B (large-scale).

Similarly, we also trained the n-gram part of the hybrid model. For this training session, we ran unigram, bigram and trigram. Both the first and second datasets have been used here. We also kept trace of the different accuracy and failure rate for different lengths. These rates led to a significance in trigram than others (unigram or bigram). As mentioned earlier, we tried to figure out the usefulness of n-gram smoothing techniques in our model but didn't pay much attention to it. Though using process interpolation or backoff could have uplifted the rate, but we had not examined them in this training phase. But we keep it on the list as an extensive task. Unlike the sequential model, here the accuracy and failure rate for both datasets are quite similar.

During the evaluation phase of the sequential model, there have been two circumstances of inputs. One is the input with known sequence or user input that is already seen by the model during the training phase. Another one is the out-of-vocabulary (OOV) case. For seen cases, our model prediction is good enough. To be specific, it can generate sequence that is close enough to the corpus. For unseen cases of OOV items, the prediction accuracy is much depending on the embedding model. Table III represents the outputted prediction of the model for both seen cases and OOV cases.

TABLE III. RESULTS OF PREDICTION FOR INPUTTED SEQUENCES

| | Predicted Sequence | Actual Sequence |
|---|---|---|
| Known Cases | আজ প্রচুর **বৃষ্টি হবে বলে মনে হয়** | আজ প্রচুর **মজা হবে** |
| | জলে কুমির **থাকতে পারে** | জলে কুমির **ডাঙায় বাঘ** |
| | হরতাল হলে তো **বাইরে যাবে নাহ** | হরতাল হলে তো **দেশের মানুষের ব্যাপক ক্ষতি হবে** |
| OOV Cases | আজ এর একটা এসপারওসপার **করবে** | আজ এর একটা এসপারওসপার **করে ছাড়বো** |
| | আমি কিংকর্তব্যবিমূঢ় **বলছি** | আমি কিংকর্তব্যবিমূঢ় **হয়ে বসে রইলাম** |
| | বৃষ্টি পরে টাপুর-টুপুর **টাপুর-টুপুর** | বৃষ্টি পরে টাপুর-টুপুর **নদে এলো বান** |

## B. *Discussion*

Through the evaluation, we tried to present the performance along with the efficiency of our model. The figures for performed experiments provide evidence that alone the sequential model or the n-gram model cannot be a better solution for predicting sequence in Bangla. So we proposed the concept of the collaboration of sequential model and n-gram. Here, the sequential model is predicting some sequences for one inputted sequence. These sequences are then inserted into the n-gram part of the model to sort out the best sequence among them based on best grammatical pattern like the right use of a particular word, most used word in the sequence.

Through our journey, we have no particular milestone in Bangla to follow, so we experimented on various models. Especially for sequence prediction, we first tried using just n-gram models following several kinds of literature [2]-[3] on Bangla. We then moved to RNN (Recurrent Neural Network) model and then focused on the sequential model as it is well known for machine translation, sequence generation, etc.

Talking about limitations, It is not an all okay situation during our whole journey. Working on an uncommon language like Bangla is one of the obstacles to tackle. It is hard to fit the model on Bangla corpus and fetch out the right output. There is no fixed deterministic accuracy scale to check model accuracy for Bangla language. We had to explore several other methods to make the output more accurate.

# VII. CONCLUSION

In this work, a hybrid framework has been proposed for Bangla Language. This proposed scheme comprises autocompletion, completing word from a user-given prefix using the trie data structure. Also, a better sequence prediction model using a hybrid implementation of sequence-to-sequence LSTM model and N-gram. Evaluation of this work presents how our model outperforms several n-gram based prediction models for Bangla. From the results, it is seen that **dataset A** has better accuracy and lesser failure rate than the **dataset B**. However, the result of **dataset B** is still promising.

Finally, auto-completion and sequence prediction is catchy features for any type of recommendation system. For cognitively disabled users, sequence prediction can be a necessary issue for writing assistance systems. For Bangla search engines, this can be very useful for query extraction and information retrieval. So, we believe this work will help these systems and also encourage future researches in a very impactful way.

## REFERENCES


[1] Cai, Fei, and Maarten De Rijke. "A survey of query auto completion in information retrieval." *Foundations and Trends® in Information Retrieval* 10.4 (2016): 273-363.

[2] Eyamin, Md Iftakher Alam, et al. "An investigative design of optimum stochastic language model for bangla autocomplete." Indonesian Journal of Electrical Engineering and Computer Science 13.2 (2019): 671-676.

[3] Haque, Md, Md Habib, and Md Rahman. "Automated word prediction in bangla language using stochastic language models." arXiv preprint arXiv:1602.07803 (2016).

[4] Yu, Lantao, et al. "Seqgan: Sequence generative adversarial nets with policy gradient." Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[5] Messina, Ronaldo, and Jerome Louradour. "Segmentation-free handwritten Chinese text recognition with LSTM-RNN." 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015.

[6] Lu, Jiasen, et al. "Knowing when to look: Adaptive attention via a visual sentinel for image captioning." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[7] Sordoni, Alessandro, et al. "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion." Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. 2015.

[8] Zhou, Xiaoqiang, et al. "Recurrent convolutional neural network for answer selection in community question answering." Neurocomputing 274 (2018): 8-18.

[9] Bickel, Steffen, Peter Haider, and Tobias Scheffer. "Predicting sentences using n-gram language models." Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. 2005.

[10] Pereira, Fernando C., Yoram Singer, and Naftali Tishby. "Beyond word n-grams." Natural Language Processing Using Very Large Corpora. Springer, Dordrecht, 1999. 121-136.

[11] Babu, A. Suresh, and P. N. V. S. P. Kumar. "Comparing neural network approach with n-gram approach for text categorization." International Journal on Computer Science and Engineering 2.1 (2010): 80-83.

[12] Luukkonen, Petri, Markus Koskela, and Patrik Floréen. "LSTM-based predictions for proactive information retrieval." arXiv preprint arXiv:1606.06137 (2016).

[13] Inan, Hakan, Khashayar Khosravi, and Richard Socher. "Tying word vectors and word classifiers: A loss framework for language modeling." arXiv preprint arXiv:1611.01462 (2016).

[14] Barman, Partha Pratim, and Abhijit Boruah. "A RNN based Approach for next word prediction in Assamese Phonetic Transcription." Procedia computer science 143 (2018): 117-123.

[15] Yenala, Harish, Manoj Chinnakotla, and Jay Goyal. "Convolutional Bi-directional LSTM for detecting inappropriate query suggestions in web search." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham, 2017.

[16] Islam, Md Sanzidul, et al. "Sequence-to-sequence Bangla sentence generation with LSTM Recurrent Neural Networks." Procedia Computer Science 152 (2019): 51-58.

[17] Abujar, Sheikh, et al. "Bengali Text generation Using Bi-directional RNN." 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2019.

[18] Rakib, Omor Faruk, et al. "Bangla Word Prediction and Sentence Completion Using GRU: An Extended Version of RNN on N-gram Language Model." 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE, 2019.

[19] Dr. M Zafor Iqbal, “SHADHASHIDHE KOTHA” , Blog, https://www.banglatribune.com/topic/মুহম্মদ+জাফর+ইকবাল/

[20] Connelly, Richard H., and F. Lockwood Morris. "A generalization of the trie data structure." Mathematical structures in computer science 5.3 (1995): 381-418.

[21] Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning." arXiv preprint arXiv:1506.00019 (2015).

[22] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv preprint arXiv:1409.1259 (2014).

[23] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[24] Olah, Christopher. “Understanding LSTM Networks.” Colah's Blog, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/#fnref1.

[25] Karpathy, Andrej, Justin Johnson, and Li Fei-Fei. "Visualizing and understanding recurrent networks." arXiv preprint arXiv:1506.02078 (2015).