Journal of Data Analysis and Information Processing, 2023, 11, 388-399
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

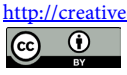
Next Words Prediction and Sentence Completion in Bangla Language Using GRU-Based RNN on N-Gram Language Model

Afranul Hoque[1], Busrat Jahan[1], Shaikat Chandra Paul[1], Zinat Ara Zabu[1], Rakhi Mondal[1], Papeya Akter[2]

[1]Department of Computer Science and Engineering, Feni University, Feni, Bangladesh
[2]Department of Computer Science and Engineering, National University, Dhaka, Bangladesh
Email: ahrafi4554@gmail.com, hossenbipasa980@gmail.com, shaikatpal56@gmail.com, zinatarazabu1997@gmail.com, mondalrakhi573@gmail.com, papeyataha@gmail.com

How to cite this paper: Hoque, A., Jahan, B., Paul, S.C., Zabu, Z.A., Mondal, R. and Akter, P. (2023) Next Words Prediction and Sentence Completion in Bangla Language Using GRU-Based RNN on N-Gram Language Model. *Journal of Data Analysis and Information Processing*, **11**, 388-399.
https://doi.org/10.4236/jdaip.2023.114020

Received: June 29, 2023
Accepted: November 3, 2023
Published: November 6, 2023



Abstract

We use a lot of devices in our daily life to communicate with others. In this modern world, people use email, Facebook, Twitter, and many other social network sites for exchanging information. People lose their valuable time misspelling and retyping, and some people are not happy to type large sentences because they face unnecessary words or grammatical issues. So, for this reason, word predictive systems help to exchange textual information more quickly, easier, and comfortably for all people. These systems predict the next most probable words and give users to choose of the needed word from these suggested words. Word prediction can help the writer by predicting the next word and helping complete the sentence correctly. This research aims to forecast the most suitable next word to complete a sentence for any given context. In this research, we have worked on the Bangla language. We have presented a process that can expect the next maximum probable and proper words and suggest a complete sentence using predicted words. In this research, GRU-based RNN has been used on the N-gram dataset to develop the proposed model. We collected a large dataset using multiple sources in the Bangla language and also compared it to the other approaches that have been used such as LSTM, and Naive Bayes. But this suggested approach provides excellent exactness than others. Here, the Unigram model provides 88.22%, Bi-gram model is 99.24%, Tri-gram model is 97.69%, and 4-gram and 5-gram models provide 99.43% and 99.78% on average accurateness. We think that our proposed method profound impression on Bangla search engines.

Keywords

Bangla Language, Words Prediction, Sentence Completion, GRU, RNN, Corpus, N-Gram

1. Introduction

Next word prediction or sentence completion works when the user types a single word of a sentence and the program delivers one or more than one most feasible word. In communication in this modern world, we use different types of devices. When we use a device, we type many words to communicate. Sometimes the next word is predicted, but some other times it is not. In the second case, we have to type the next word, and then the prediction is recomputed. This is excessive to type many words and also takes more time to communicate. In this case, the next word prediction method will help to complete work easily because it will predict the next reasonable word. Here we can choose the highest predictable word instead of typing them. Guessing or finding which words are predicted to chase an order of words or part of the text is called word prediction [1]. The next word prediction method predicts new words by scrutinizing former word flow for concluding sentences with much exactness. This technique reduces the number of keystrokes required to misspell and type words. Predicting the next word can help elementary students such as researchers, programmers, or students avoid further spelling mistakes and speed up their typing. We use a big data set to train the N-gram model to predict the proper next words for accurately concluding a Bangla sentence. Physically, perceptively, or cognitively challenged, many people on the earth are slow typists. We studied many methods for the next word prediction and complete sentences in different languages like English, Arabic, and Urdu, but most of them are English texts. Some researchers are working with the Bangla language to predict the next words or complete the full sentence [2]. However, we try to make a new model with better accuracy in our work. This work is used GRU (Gated Recurrent Unit) on the N-gram language model.

The performance of this study outcome is:

- ✓ To complete sentences together including the most probable word forecast for the Bangla language.
- ✓ To use huge data to find better accurateness, which was not previously used for predicting Bengali words.
- ✓ To give more reasonable exactness than other ways that have been used.
- ✓ To give 92.64% average accuracy in using 5 models and also gives 99.78% best accuracy in the 5-gram model.

We have applied GRU-based RNN on an n-gram dataset to develop this proposed model. It can predict the next probable words of the input words. We deal with the sequential data, find the most predicted word, not only one word but also one or several words and finally find the predicted sentence in Bangla Language. A total of 310 million people all over the world speak Bangla as a first or second language. Among these, 150 M speakers are from Bangladesh, and another 95 M are from India. West Bengal, Tripura, Assam, and some Indian immigrants in the USA, UK, and the middle east are included as Bangla speakers. Bangla is also the state Language of Bangladesh, and it is an officially recog-

nized language in India. But a huge amount of people is not satisfied when they write a text using Bangla on a device for sending their data. They face problems When they try to guess the next word and try to complete a sentence.

2. Related Works

Some recent studies have been done on text word prediction, and next letter prediction systems in the Bangla word language to find the higher probable words. Next word prediction helps to complete sentences which is very important for saving time. P. Burman *et al.* mentions in their work [3] that they applied Long short-term memory (LSTM) network with RNN (Recurrent Neural Network) to the Assamese to predict the next probable word. They got 72.10% exactness for Assamese transcripts and 88.20% for text. S. Bickel, P. Haider, and T. Scheffer [4] proposed a model using individual emails, climate information, cooking recipes, and call-center email data to predict the most preferred words while typing.

Haque *et al.* [5] used N-gram-based word prediction for the Bangla language, where the Bigram model performs politely, but the unigram model's performance is evidently poor. But the exactness (63.5%) of the backoff model is very good. They have tested personal and professional email, cooking recipes, weather news and others. The Authors create an assessment metric and adapt N-gram models to the difficulty of predicting the subsequent words, given a primary text fragment. 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% [6]. Al-Mubaid and Hisham have [7] proposed a new word prediction machine learning model. By supplying word predictors with highly discriminatory features that were selected using various feature selection procedures, this method presents the problem as a learning-classification task. The unique mix of the best performer in machine learning, SVM, with several characteristic selections, approaches MI, X2, and more is what makes this study unique in its approach to presenting this problem. The experimental findings clearly show that the approach is adequate for predicting the right phrases from limited contexts.

In this research [8], M. Soam and S. Thakur studied NLP, and various deep learning techniques such as LSTM, and BiLSTM, and executed a comparative study. The accuracy acquired using BiLSTM and LSTM are 66.1% and 58.27% respectively. Ambulgekar *et al.* [9] used Recurrent Neural Networks (RNN) for next words prediction and their model comprehends 40 letters and anticipates impending top 10 words which will be executed utilizing TensorFlow. In this research [10], A. Rianti and S. Widodo used the LSTM model to complete the prediction with 200 epochs. The outcome displayed that it claimed to get an accuracy of 75% while the loss was 55%. Kumar *et al.* [11] used this machine learning technique and TensorFlow, Keras, dictionaries, pandas, and NumPy packages.

They have trained the model in 500 iterations (Epochs). In this research [12], R. Sharma, N. Goel used two deep learning techniques namely Long Short-Term Memory (LSTM) and Bi-LSTM explored for the task of predicting the next word, and an accuracy of 59.46% and 81.07% was observed for LSTM and Bi-LSTM respectively. Endalie *et al.* [13] present a Bi-directional Long Short Term-Gated Recurrent Unit (BLST-GRU) network model for the prediction of the next word for the Amharic Language. They evaluated the proposed network model with 63,300 Amharic sentences, producing 78.6% accuracy.

3. Methodology

We know methodology is the most important part of a work. It helps to achieve our goals and get significant results. In this paper, we use an advanced technique to achieve a good result. We divided our methodology into two parts. One is a dataset summary, and the other part is implementation.

3.1. Dataset Summary

For this work, we collect a vast amount of data in the Bengali language. We tried a unique method for this work and used 114,852 amounts of data. We collected all data from different sources. **Table 1** shows the data collection summary from several sources.

After collecting all data we remove unwanted objects like (“, (), /, ! |, ?, #, [,]) and also removed the English word, another language word that is found in our dataset. Because we need only Bangla words to get a good result. We can use this cleaning dataset to get standard values for other purposes.

We divided collecting data into two parts which are train data and test data. In this model Train and test data are used for the prediction of the next word. We proposed a sequence model for next words prediction and also used an embedding layer for Natural Language Processing. In this model, there is a dense layer for connecting with every preceding layer.

✓ **Dataset Cleaning Process**

For data cleaning, we take our total dataset into a function. This function removes all unwanted objects like (“, (), /, ! , |, ?, #, [,], ’, ", \) and also remove English and other language word. We create 5 various datasets from the cleaning standard dataset using the n-gram idea which includes uni-gram, bi-gram, tri-gram, 4-gram, and 5-gram.

We showed in **Figure 1**, the dataset cleaning format, also the creation of 5 new clean datasets. The n-gram language model allocates possibilities of the next word sequence and this model follows an (n − 1) order of n items. Here we created 5 models which will give us different outputs for different inputs. Usually, to forecast the possible next word or words, the necessary number of input text can be varied. When we input a single word and predict the next probable words. This is called unigram or 1-gram [14]. Unigram does not use the history of words. When input will be two words and predict the next words using the

Table 1. Data collection summary.

Source of Words	Total Words	Individual Words
Bangla Newspaper	55,000	7500
Social Media (Facebook, YouTube)	40,000	5000
Bangla Academic Books	19,852	4900

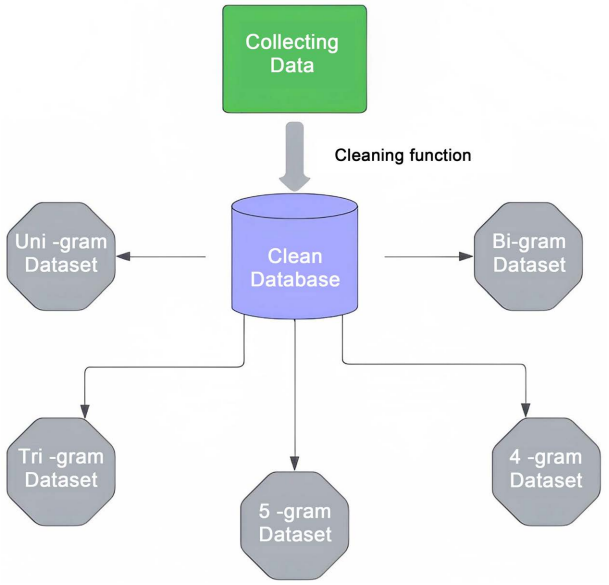


Figure 1. Workflow Diagram of dataset cleaning.

history of one word from two words. It is called the Bi-gram model. Similarly, the input will be three words to predict the next words using the history of the last two words from three words. It is called the tri-gram or 3-gram model. Again 4-gram and 5-gram also input four and five words and it also takes the history of the last 3 and 4 words to predict the next words. Generally, the previous 4 or 5 words are enough to understand the sequence sufficiently. For a better understanding of this model, we use a Bengali sentence as an example—We exhibit our models using this sentence in Table 2.

আমার সোনার বাংলা আমি তোমায় ভালোবাসি

3.2. Implementation

To train the proposed model, we made a corpus of 114,873 words. The data of the corpus has been taken from popular Bangla newspapers entitled the daily "Prothom Alo", popular social media Facebook, YouTube, and Bangla academic books. The corpus contains 17,400 unique words. The N-gram model measures the possibility of feasible next words and N-gram models cannot deal with incoming problems for zero possibility due to the dataset's lack of expected next terms [15]. It elects the maximal probable words from all the probabilities words and sets the new one or more words. Usually, language models calculate the

Table 2. Exhibit models using a sentence.

Model Name	Input X	Output Y
Uni-gram model	আমার	সোনার বাংলা আমি
Bi-gram model	আমার সোনার	বাংলা আমি তোমায়
Tri-gram model	আমার সোনার বাংলা	আমি তোমায় ভালোবাসি

Likewise, the 5-gram model uses five inputs and provides three output values like other models.

possibility of the presence of a word counts in a certain sequence. For Example, when the model uses bi-grams, the frequency of each bi-gram is computed by combining every word with its previous work, which the frequency of the related uni-gram would split.

The relationships between the bigram and trigram models are shown in Equations (2) and (3) below [16].

$$p\left(w_i/w_{i-1}\right)=\text{count}\left(w_{i-1},w_i\right)/\text{count}\left(w_i\right) \tag{1}$$

$$p\left(w_i/w_{i-2},w_{i-1}\right)=\text{count}\left(w_{i-2},w_{i-1},w_i\right)/\text{count}\left(w_{i-2},w_{i-1}\right) \tag{2}$$

We know n-gram is a good contribution to word prediction but there are some limitations here. For this reason, it cannot advise the next most possible words. It also failed to predict the proper standard. N-gram does not work efficiently where the dataset is a big and too lengthy sequence. Some methods like Back-off and Katz Back-off are used in n-gram for the probability distribution with the small count and a sigmoid activation function in Equation (4) is employed to compress the result between 0 and 1 once the two outcomes are concurrently added [17]. Here, we predict the next proper words using Bengali language corpus data and also suggest two full sentences. Our research also used RNN (Recurrent Neural Network) because this performs nicely with sequential data. RNN uses Equation (3) for permanent state by using output as input where the input is x_t having a weight W and h_{t-1} having a weight value of U.

$$h_t=\tanh\left(Wx_t+r_t\Theta Uh_{t-1}\right) \tag{3}$$

Figure 2 shows the GRU-based RNN models training structure.

There is a limitation of RNN is that RNN has a problem remembering long sequences. Vanishing gradients are an issue with RNNs that makes learning from lengthy data sequences difficult. The gradients contain the data used to update the RNN parameters, and when the gradient shrinks, the parameter updates lose significance, which implies no true learning is done [18].

We are able to solve the vanishing gradient problem using GRU (Gated Recurrent Unit) and LSTM (Long Short-Term Memory). The update gate and reset gate are the two gates that GRU utilizes, hence we used it in our research [18]. However, LSTM employs three gates—input, forget, and output—to address the gradient problem [18]. GRU lack both internal memory and the output gates found in LSTM. GRU performs quicker than LSTM and consumes less training parameters and memory [19].

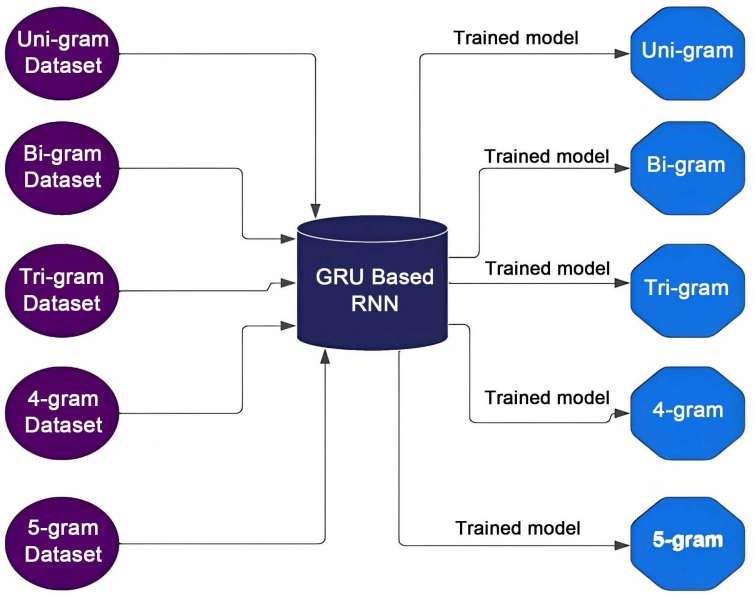


Figure 2. GRU-based RNN models training structure.

The update gate and the reset gate are two vectors that select the information to pass to the output in the GURU model. In Equation (4) counting the update gate z_t for time step t,

$$Z_t = \sigma W\left({}^{(z)}x_t + U^{(z)}h_{t-1}\right) \tag{4}$$

When x_t is connected to the network unit, its weight W is multiplied (z). It's the same with $h_{(t-1)}$, which contains information about previous $t-1$ units and is multiplied by its own weight $U^{(z)}$. Basically, the Reset gate is utilized from the model to determine how much past information must be forgotten. Reset gate use Equation (5) for calculating:

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right) \tag{5}$$

Inputs h_{t-1} and x_t are added, their corresponding weights are multiplied, the results are added, and the sigmoid function is used. In this research, we trained all datasets using GRU-based RNN and created five models.

Figure 3 displays the trained model structure. Embedding (Embedding), gru_4 (GRU), gru_5, dense_4 (Dense) and dense_5 are five hidden layers and entire params of 2, 719, 389, trained params 2, 719, 389, non-trained params 0 in trained models.

✓ **Word Prediction**

We got five trained models from completing the training of five datasets. These Five models take different length inputs and determine the next predicted words as output. So, when the input word length is one, it will go Uni-gram trained model because Uni-gram trained model is prepared for one input word and predicts three next words.

Similarly, the input word is two, then it will go to the bi-gram trained model because it takes two words as input and predicts the next three words.

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 3, 50)	861950
gru_4 (GRU)	(None, 3, 100)	45600
gru_5 (GRU)	(None, 100)	60600
dense_4 (Dense)	(None, 100)	10100
dense_5 (Dense)	(None, 17239)	1741139

Total params: 2,719,389
Trainable params: 2,719,389
Non-trainable params: 0

Figure 3. Type of layer with the respective where n = 3.

Correspondingly, the input words are three then it will go to the trained Tri-gram model and predict the following three words using the last two inputted sequences. Again, the input word is four and five, then it will be sent into a 4-gram and 5-gram trained model then it will predict the next three words. Something is different when the input number length is more than 5. In this case, it would use only the last 3-word sequence and then send it to the tri-gram model to predict the next three words.

Figure 4 shows the input that is taken from the keyboard and sent to the trained model and displays the prediction of the next word from the trained model.

✓ **Sentence Completion**

Firstly, we said, in this research, will predict the next probable words and further propose a whole sentence using these predicted words. We used our earlier proposed structure of the N-gram model which is trained by GRU base RNN. We add our output (predicted next words) with the previous input values. We can use our new current input values to predict furthermore words which ultimately creates a full sentence. This method will end after completing a full sentence. In this work, we give the highest number of words length 15 for a sentence. Therefore, the whole output should be the proposed possible sentence. Now In **Table 3**, we have a Bengali sentence as an example to understand it better Predicted words:

4. Results

This time we need to experiment and analyze our proposed method. Because to ratify the proposed method, we must execute tests and analyze our results sincerely.

Consequently, evaluate our offered method on a corpus dataset. We train all models with similar structures up to 2500 epochs (**Figure 5**). Among the five trained models, the average exactness of the unigram model is 81.22% and the average Loss of this model is 33.73%. The average accuracy in the Bi-gram model is 89.31% and 19.56% is the average Loss in this model. The Tri-gram model

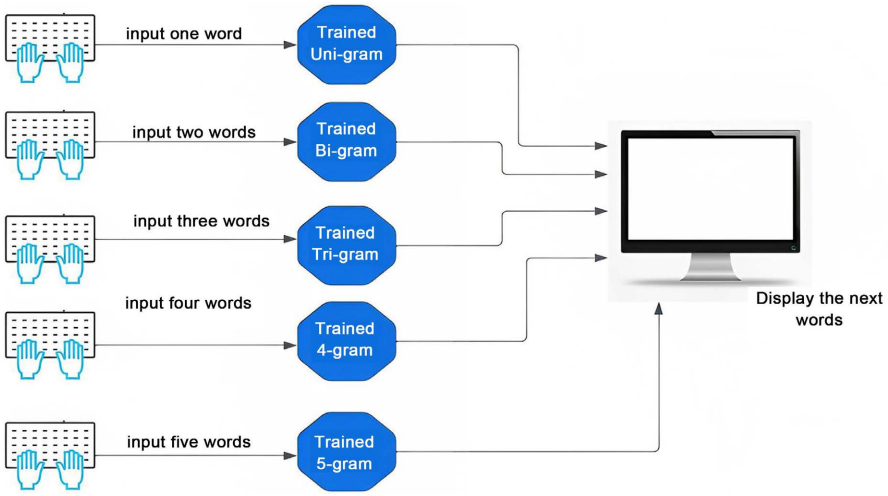


Figure 4. Processing the next probable word using the trained model.

Table 3. Example of probable next word.

Input X	Output Y
আন্তর্জাতিক ক্রিকেট	সংস্থা ঘোষিত বিশ্বকাপের

Completing sentence: আন্তর্জাতিক ক্রিকেট সংস্থা ঘোষিত বিশ্বকাপের সেরা একাদশে সাকিব থাকবেন তা ধারণাই ছিল

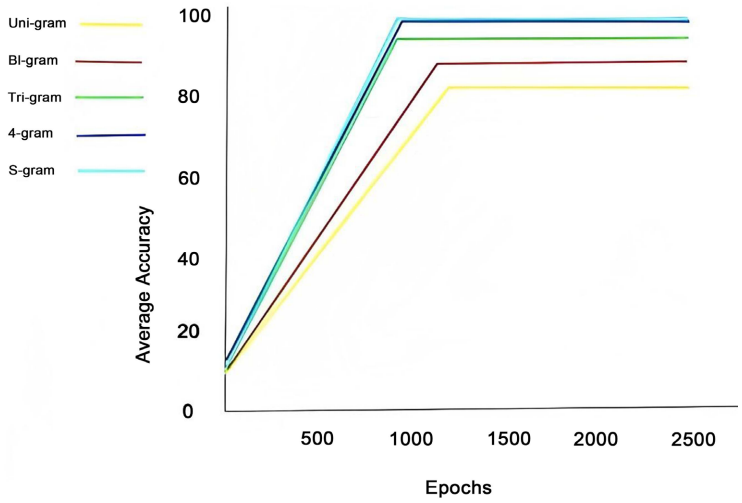


Figure 5. The average exactness of the trained model in a Graphical Presentation.

has an average accuracy of 97.69% and 4.13% average loss for this dataset. Similarly, 4-gram and 5-gram models get an average accuracy of 99.43% and 99.78%, whereas these two models have an average loss of 2.07% and 1.15%.

Figure 5 shows the average accuracy of our five models. Where every model started 0 and completed 2500 epochs, and after 700+ epochs, the accuracy is stable. The following Table 4 shows the average accuracy and average Loss for five trained models. Where we see when the input number is increased then average accuracy is increased, and average losses are decreased.

Table 4. The average accuracy and average Loss.

Model	Average Accuracy	Average Loss
Uni-gram	81.22%	33.73%
Bi-gram	89.31%	19.56%
Tri-gram	97.69%	4.13%
4-gram	99.43%	2.07%
5-gram	99.78%	1.15%

We used GRU in the N-gram language model, where we compared the results of our suggested model with the models in other research papers.

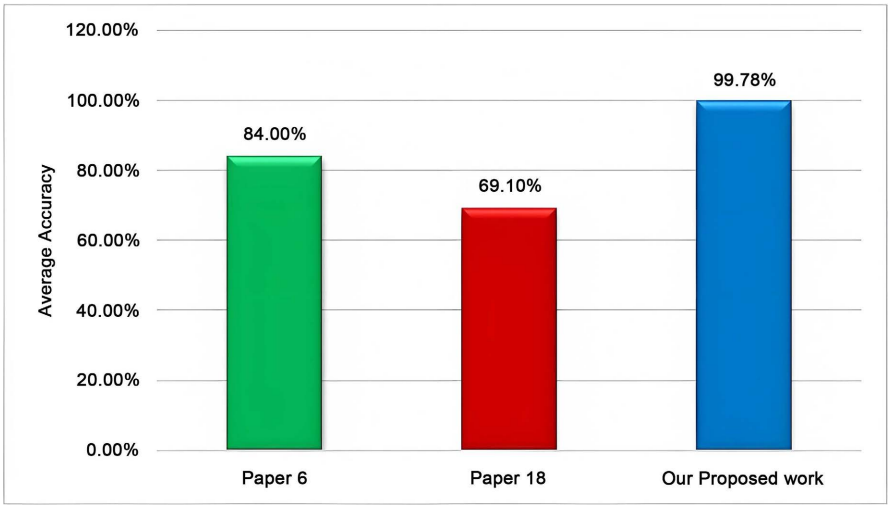


Figure 6. Comparison with other research papers.

Figure 6 shows the comparison among paper 6, paper 18 and our proposed system. In paper 6, they used Tri and a Hybrid Approach of Sequential LSTM and N-gram and their accuracy was 84%. On the other hand, we have a maximum efficiency of 99.78% for high-order sequences. In paper 18 [20], they got an accuracy on average 69.1% for their proposed method whereas our paper has 99.78%.

5. Conclusion

We used a larger data corpus than other researchers in the Bangla language. GRU-based RNN has displayed a noteworthy performance in this work to predict the next most probable Bangla words and complete sentences. As we can see, our research work has better results than other research works in the Bengali language. Our proposed method in higher-order like Tri-gram, 4-gram, and 5-gram, exactness rate is very good (respectively 97.69%, 99.43%, 99.78%). All in all, our proposed method is impressive because we use a larger dataset here. In this study, the Bangla dataset used is very challenging because there is no ready-made dataset for the Bengali language. So, we manage datasets from various origins.

Acknowledgments

I would like to thank my all co-authors for their support and encouragement.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] (2023) What Is Word Prediction? https://www2.edc.org/ncip/library/wp/what_is.htm

[2] Yazdani, A., Safdari, R., Golkar, A. and Niakan Kalhori S.R., (2019) Words Prediction Based on N-Gram Model for Free-Text Entry in Electronic Health Records. *Health Information Science and Systems*, **7**, Article No. 6. https://doi.org/10.1007/s13755-019-0065-5

[3] Barman, P.P. and Boruah, A. (2018) A RNN Based Approach for Next Word Prediction in Assamese Phonetic Transcription. *Procedia Computer Science*, **143**, 117-123. https://doi.org/10.1016/j.procs.2018.10.359

[4] Bickel, S., Haider, P. and Scheffer, T. (2005) Predicting Sentences Using N-Gram Language Models. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, 6-8 October 2005, 193-200. https://doi.org/10.3115/1220575.1220600

[5] Haque, M.M., Habib, M.T. and Rahman, M.M. (2015) Automated Word Prediction in Bangla Language Using Stochastic Language Models. *International Journal in Foundations of Computer Science & Technology*, **5**, 67-75. https://doi.org/10.5121/ijfcst.2015.5607

[6] Hamarashid, H.K., Saeed, S.A. and Rashid, T.A. (2020) Next Word Prediction Based on the N-Gram Model for Kurdish Sorani and Kurmanji. *Neural Computing and Applications*, **33**, 4547-4566. https://doi.org/10.1007/s00521-020-05245-3

[7] Al-Mubaid, H. (2007) A Learning-Classification Based Approach for Word Prediction. *The International Arab Journal of Information Technology*, **4**, 264-271.

[8] Soam, M. and Thakur, S. (2022) Next Word Prediction Using Deep Learning: A Comparative Study. 2022 12*th International Conference on Cloud Computing, Data Science & Engineering* (*Confluence*), Noida, 27-28 January 2022, 653-658. https://doi.org/10.1109/Confluence52989.2022.9734151

[9] Ambulgekar, S., Malewadikar, S., Garande, R. and Joshi, B. (2021) Next Words Prediction Using Recurrent NeuralNetworks. *ITM Web of Conferences*, **40**, Article ID: 03034. https://doi.org/10.1051/itmconf/20214003034

[10] Rianti, A., Widodo, S., Ayuningtyas, A.D. and Hermawan, F.B. (2022) Next Word Prediction Using Lstm. *Journal of Information Technology and Its Utilization*, **5**, 10-13. https://doi.org/10.56873/jitu.5.1.4748

[11] Kumar, A., Kumar Mishra, P., Namgai, T. and Kumar, S. (2023) Next Word Prediction in Bodhi Language Using LSTM Based Approach. https://ssrn.com/abstract=4367666 https://doi.org/10.2139/ssrn.4367666

[12] Sharma, R., Goel, N., Aggarwal, N., Kaur, P. and Prakash, C. (2019) Next Word Prediction in Hindi Using Deep Learning Techniques. 2019 *International Conference on Data Science and Engineering* (*ICDSE*), Patna, 26-28 September 2019,

55-60. https://doi.org/10.1109/ICDSE47409.2019.8971796

[13] Endalie, D., Haile, G. and Taye, W. (2022) Bi-Directional Long Short-Term Memory-Gated Re-Current Unit Model for Amharic Next Word Prediction. *PLOS ONE*, **17**, e0273156. https://doi.org/10.1371/journal.pone.0273156

[14] Kapadia, S. (2019) Language Models: N-Gram. A Step into Statistical Language Modeling. https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9

[15] Rakib, O.F., Akter, S., Khan, M.A., Das, A.K. and Habibullah, K.M. (2019) 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*), Dhaka, 24-25 December 2019, 1-6. https://doi.org/10.1109/STI47673.2019.9068063

[16] Inan, H., Khosravi, K. and Socher, R. (2016). Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling. arXiv:1611.01462.

[17] Kostadinov, S. (2017) Understanding GRU Networks. https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

[18] Arbel, N. (2018) How LSTM Networks Solve the Problem of Vanishing Gradients. https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

[19] Mystery Vault (2021) LSTM Vs GRU in Recurrent Neural Network: A Comparative Study. https://analyticsindiamag.com/lstm-vs-gru-in-recurrent-neural-network-a-comparative-study/

[20] Habib, M.T., Al-Mamun, A., Rahman, M.S., Siddiquee, S.M.T. and Ahmed, F. (2018) An Exploratory Approach to Find a Novel Metric Based Optimum Language Model for Automatic Bangla Word Prediction. *International Journal of Intelligent Systems and Applications* (*IJISA*), **10**, 47-54. https://doi.org/10.5815/ijisa.2018.02.05