# A Bengali Text Generation Approach in Context of Abstractive Text Summarization Using RNN

**Chapter** · March 2020

DOI: 10.1007/978-981-15-2043-3_55



**5 authors**, including:

Sheikh Abujar
Daffodil International University
**52** PUBLICATIONS **83** CITATIONS
SEE PROFILE

Abu Kaisar Mohammad Masum
Daffodil International University
**8** PUBLICATIONS **1** CITATION
SEE PROFILE

Md. Sanzidul Islam
Daffodil International University
**18** PUBLICATIONS **41** CITATIONS
SEE PROFILE

Fahad Faisal
Universidade de Évora
**14** PUBLICATIONS **9** CITATIONS
SEE PROFILE

# A Bengali Text Generation Approach in Context of Abstractive Text Summarization using RNN

Sheikh Abujar [1], Abu Kaisar Mohammad Masum [1], Md. Sanzidul Islam [1], Fahad Faisal [1], Syed Akhter Hossain [1]
Dept. of CSE , Daffodil International University, Dhaka, Bangladesh
{sheikh.cse, mohammad15-6759, sanzidul15-5223, fahad.cse}@diu.edu.bd, aktarhossain@daffodilvarsity.edu.bd

**Abstract.** Automatic text summarization is one of the mentionable research area of natural language processing. The amount of data is increasing rapidly as well the necessity of understanding the gist of any text is just a mandatory tools, now a days. The area of text summarization has been developing since many years. Mentionable research has been already done through extractive summarization approach, in other side- abstractive summarization approach, is the way to summarize any text as like – human. Machine will be able to provide a new type of summarization, where the understanding of given summary may found as like as human generated summary. Several research development has already been done for Abstractive summarization in English language. This paper shows, a necessary method – "text generation" in context of Bengali abstractive text summarization development. Text generation, helps the machine to understand the pattern of human written text and then produce the output as is human written text. A basis Recurrent Neural Network (RNN) has been applied for this text generations approach. The most applicable and successful RNN - long shortterm memory (LSTM) has been applied. Contextual tokens has been used for the better sequence prediction. The proposed method has been developed in context of making it useable for further development of abstractive text summarization.

**Keywords: Natural Language Processing, Deep Learning, Text Pre-processing, Text Generation, Abstractive text summarization. Bengali text summarization.**

## 1 Introduction

Machine learning and Data mining algorithms performs better after using a large number of labeled dataset. The importance of using a large dataset is- it helps the machine to understand the pattern based on any specific or general requirements, form that dataset and be able to produce the better output. Text summarization is one the most necessary branch of Natural language processing research. Extractive text summarization relies on the frequency, word and/or sentence repetitive nature, several word/sentence scoring methods and some other lexical analysis [1]. Text summarization in both English and Bangla language has already been successfully developed. But this type of summarization may not useful in this days. The necessity

of doing research in now a days, is to develop machines to understand the context of any information given and be able to produce the summary based on its understanding. This type of summarization are in a stage of making comparison with human generated summaries, and it's called abstractive summarization. Now a days, information in every language is available in internet or offline. Major research on this domain has done for English language and a few in Bengali language. The Bengali language has several limitations in data preprocessing. The best way of overcoming several problems is converting those text into Unicode [2]. Dataset is the major contribution towards successful research outcome, large scale of labeled dataset is must for this purpose.RNN process the sequence data very well because of its recurrent structure. Its hidden units are updated all the time and it has no limitation in sequence length. Both forward and backward computation helps the neurons to understand the sequence [3]. Majority of classifiers cannot provide expected result if the dataset is very small.For, Abstractive summarization, machine requires the understanding of human written text structure. From where, machine can understand the pattern of human written text. Based on this understanding – machine will provide summary by its own. To write a new sentence, machine needs to use its previous learning patterns of human wrote sentence. In this way, those predicted summary contains - incomplete sentences, may get the complete and corrected form. This paper represent the research implementation of text generation. The entire preprocessing, dataset structures and results has been discussed.

## 2 Literature Review

LSTM is most widely used RNN model in current days [4], here based on contextual token, every neuron gates helps the model in the process of predicting the next pattern more accurately. With this consequence, bidirectional LSTM model has been built [5]. When there are so many variety in data sequence, this type of LSTM helps to generate sequence data as well made the entire model more easy and useful. Several direct or embedded sentence generation has been done using LSTM [6], Sequence Generative adversarial nets (SeqGan) used Monte Carlo method to identify the next predicted token. This method has also applied using neural network decoder with domain based knowledge for dialogue generation [7]. Sentence generation is entirely a decision making system, and a computational representation any information, which requires to understand the sequence of data in many forms. It follows a goal oriented method, Several Reinforcement learning models has applied for sentence generation [8], such as - actor-critic algorithm. Ho et al. [9] explained the relation between GAN and inverse Reinforcement learning. Hu and Yang et al. [10], has done several research to reduce the loss between input and target output data using encoderdecoder problem, recently VAE achieved outstanding result. Abstractive methods required deep investigation of the given text as input and extract the knowledge purpose of generating new sentence. Tanaka et al. [11] has explained several content selection technique as well rewriting methods. With improvement consequence of sentence generation, abstractive methods will be more accurate and machines will be able to predict and be able to

complete writing the whole sentence considering the predicted contextual tokens. Text generation is significant for the arrangement to grouping word order. This paper we attempted to clarify a technique for how to create Bengali word next succession utilizing LSTM and RNN. Genuine utilization of the Text generations a machine interpretation for the Bengali language.

## 3 Methodology

Language modelling is the most important part of modern NLP. There is some part of the task such as text summarization, machine translation, text generation, speech to text generation etc. Text generation is a significant part of Language Modelling. A well-trained language model does acquire knowledge of the probability of the event of a word based on the previous series of words. In this paper, we discussed n-gram language modelling for text generation and create a Recurrent Neural Network for training model. In figure1 has been given our work methodology flow.

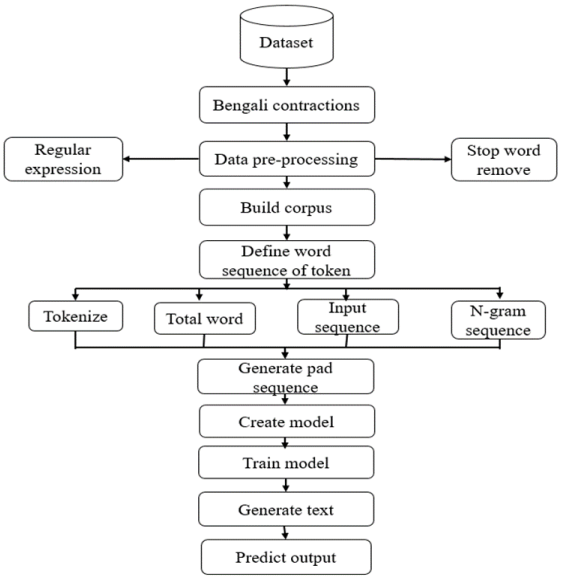


Figure1: Working flow for Text Generation

A. Data collect & pre-processing

Since we are working with Bengali text so need a good dataset. We use our own dataset which was collect from social media. Our dataset contains several types of Bengali post such group post, personal post, page post etc. There is some obstacle to collect Bengali data such as the structure of Bengali text. But in our dataset, we try to reduce all of that obstacle to keep a pure Bengali text. Our dataset contains

text data with their type and text summary. For our working purpose, we use only text and their summary to generate a sequence of next Bengali word.

Before prepare dataset for text generation, we need to add Bengali contractions. Because contractions contain a short form of a word such as "বি.দ্র"="বিশেষ দ্রষ্টব্য", "ড."="ডক্টর". After collecting dataset we need to a clean dataset to generate text. So for clean data, we remove whitespace, digits, punctuation from Bengali text and remove Bengali stop words from a Bengali stop word text file. Finally, we clean the text and create a list which contains text with their summary. Then we create a corpus for text generation.

B. *N-gram Tokens Sequence*

For text generation language model required a sequence of the token and which can predict the probability next word or sequence. So need to tokenize the words. We use keras build in tokenize model which extract word with their index number from the corpus. After this, all text transforms the sequence of the token. In n-gram, the sequence contains integer number token which was made from the input text corpus. Every integer number represent the index of the word which is in the text vocabulary. Example given in table 1.

| N-GRAM TEXT | TOKEN SEQUENCE |
|---|---|
| হাইটেক পার্ক | [103,45] |
| হাইটেক পার্ক নির্মাণ | [103,45,10] |
| হাইটেক পার্ক নির্মাণ কাজ | [103,45,10,24] |
| হাইটেক পার্ক নির্মাণ কাজ হাতে | [103,45,10,24,33] |
| হাইটেক পার্ক নির্মাণ কাজ হাতে নিয়েছে | [103,45,10,24,33,67] |
| হাইটেক পার্ক নির্মাণ কাজ হাতে নিয়েছে সরকার | [103,45,10,24,33,67,89] |

Table1: Example of n-gram sequence token

C. *Pad Sequence*

Every sequence has a different length. So we need to pad sequence for making sequence length equal. For this intention, we use keras pad sequence function. The input of the learning model we use n-gram sequence as given word and the predicted word as next word. Example given in table 2. Finally, we can do acquire the input X and the next word Y which is used for training model

| *Given word* | *Next Word* |
|---|---|
| *জ্যামের* | *জ্যামের জন্য* |
| *জ্যামের জন্য* | *জ্যামের জন্য এক্সাম* |
| *জ্যামের জন্য এক্সাম* | *জ্যামের জন্য এক্সাম মিস* |

Table2: Example of pad sequence

D. Proposed Model

A recurrent neural network works extremely goods for sequential data. Because it's can remember it output cause of exterior memory. It can predict upcoming next sequence using memory and also deep understanding with its sequence compared to other algorithms. When it can consider the current state also can remember what it learns from the previous state. RNN has the long short term memory (LSTM) that helps to remember the previous sequence. Generally, Recurrent Neural Network has two input one is its present input and another is recent previous. Because remember the sequence current input and previous input both help to generate a complete text. RNN apply weights of the sequence as input with time and produce weights of next sequence as output.

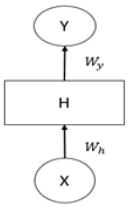


Figure2: Recurrent Neural Network

The formula will be,

$$\mathrm{H} = \sigma(W_h * \mathrm{X}) \quad (1)$$

$$\mathrm{Y} = \mathrm{Softmax}(W_y * \mathrm{H}) \quad (2)$$

Here, $\sigma = \mathrm{Activation\ Function}$

$\mathrm{X} = \mathrm{Input}$ ,$\mathrm{Y} = \mathrm{Output}$, $\mathrm{H} = \mathrm{Hidden\ State}$, $\mathrm{W} = \mathrm{Weight}$

In our proposed model, we use the weight (w) of text sequence as input with the time (t).LSTM cell can store previous input state and then working with the current state. In figure3 shows, the input is a previous state and is the current state. When working in the current state in can remember previous then using activation function it can predict the next word or sequence. For train our model we define keras sequential model and embedding the total word with input sequence. Define LSTM with 256 units and 0.5 dropouts. Add Dense which is equal of the total word and use softmax activate function. For loss function calculation we use 'categorical crossentropy' and use 'Adam' optimization function.

***Algorithm1*** *for Bengali text generation*

```
Set function model  create(max sequence length, total word):
    declare Sequential()
  add(Embedding(total word, number of word, input word length))
    add(LSTM(size))
    add(Dropout(value))
    add(Dense(total words, activation function))
    compile(loss, optimization)
    return model
create model(max sequence length, total words)
```

This segment we demonstrate our model graphical view. Here remarkable id is the contribution of the procedure will proceed to the Dense or yield layer.

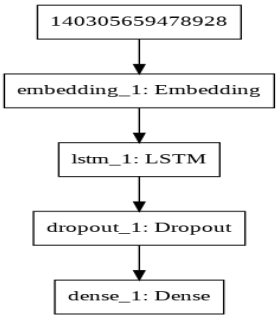


Figure3: Visualizing LSTM Model structure

In figure 4 shows, a short view of the working model, here lstm can store the previous sequence. When working with the current state and find the next sequence its use the activation function. Softmax activation calculate the probability and keep only the correct next sequence.

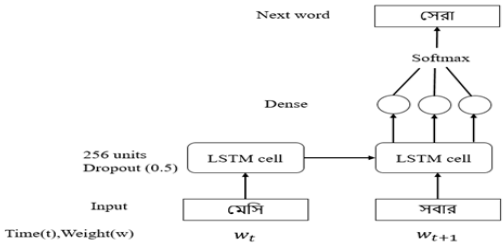


Figure4: View of Proposed Model.

i. *Long Short Term Memory:*

Long Short Term Memory is a part of the Recurrent Neural Network. It's used to disappearance of gradient and abolishes gradient. Every LSTM cell has

three gates such as Input Gate, Forget Gate, Output Gate and a cell state which added information via the gates.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (3)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (4)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (5)$$
$$c_t = f_t * c_{t-1} + i_t * \sigma(w_c[h_{t-1}, x_c] + b_c) \quad (6)$$
$$h_t = o_t * \sigma(c_t) \quad (7)$$

Here, $i_t$ = input gate′s, $f_t$ = forget gate's ,
$o_t$ = output gate, $c_t$ = cell state,
$h_t$ = hidden state, $\sigma$ = activation function

*ii. Activation function:*

The softmax function is the logistic activation function, which is used to deal with classification problems. It maintains the output between 0 and 1 calculations probability. The formula for softmax function is,

$$\sigma(Z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \quad (8)$$

Here, z is the inputs to the output layer and j indexes the output.

## Experiment and Output

After creating the model function we need to train our model. We fit the model with the current and next word. Set the epochs size 150 and set verbose= 2.Train model almost 3 hours it gives a better accuracy 97% with loss 0.0132.Figure4 shows model train accuracy graph and figure5 show loss graph of the model.

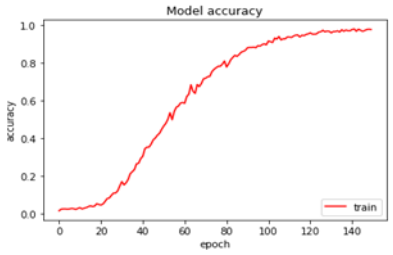


Fig 4: Model Accuracy graph

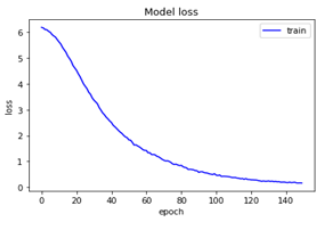


Fig5: Model loss graph

Previously several research work completed for English text generation with one direction RNN or LSTM. But in the Bengali language, very few research work completed using LSTM for Text generation. This paper we applied a method and process Bengali text for text generation and provide a better output. Algorithms perform given below in the table3

| ***Approach*** | ***Accuracy*** | ***Loss*** |
|---|---|---|
| ***General LSTM*** | 93% | *0.01793* |
| ***Using LSTM*** | 97% | 0.0132 |

Table3: Comparison with using LSTM and general LSTM

This experiment our main goal to create the next sequence of words. For output, we have created a function where we set a token list, seed text for showing output. We have fixed the seed word and set the length of predictor next word, call the model with maximum sequence length.Table4 shows our experiment result.

| Given Text | Output |
|---|---|
| হাইটেক পার্ক | হাইটেক পার্ক নির্মাণ কাজ হাতে নিয়েছে সরকার |
| উজ্জ্বল অর্থের | উজ্জ্বল অর্থের প্রয়োজনে মিথ্যা সংবাদ প্রচার করে |

Table4: Bengali Text Generation

## Conclusion and Future work

We have proposed a good method for generating an automatic Bengali text generation. Since no model gives accurate result but our model provides better output and maximum output is accurate. Using our proposed model we have easily generated a fixed length and meaning full Bengali text.

There are some limitations this paper such as can not generate text without given the length of the text and n-gram sequence defined needed which is a lengthy process. Sometimes the order of the sentence is not correct in giving output.

There are some defects in our proposed methodology such as can not generate random length text. We need to define the generating text length. Another defect is we need to define pad token for predict next words. In our future work, we will make an automatic text generator which provides a random length Bengali text without using any token or sequence.

## Acknowledgment

We would like to give thanks to our DIU-NLP and Machine Learning Research Lab for providing all research facility and guidance. We would also give special thanks to our Computer science and engineering department to support in completing our research.

## *References*

[1] Abujar S et al (2017) A heuristic approach of text summarization for Bengali documentation. In: 8th IEEE ICCCNT 2017, IIT Delhi, Delhi, India, 3–5 July 2017

[2] Abujar S, Hasan M (2016) A comprehensive text analysis for Bengali TTS using Unicode. In: 5th IEEE international conference on informatics, electronics and vision (ICIEV), Dhaka, Bangladesh, 13–14 May 2016

[3] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681, 1997.

[4] Sepp Hochreiter and Ju¨rgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[5] Alex Graves, Santiago Fern´andez, and J¨urgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005, pages 753–753, 2005.

[6] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. arXiv preprint arXiv:1609.05473, 2016.

[7] Jiwei Li, Will Monroe, and Dan Jurafsky. Learning to decode for future success. arXiv preprint arXiv:1701.06549, 2017.

[8] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction, volume 1(1). MIT press Cambridge, 1998.

[9] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Advances in Neural Information Processing Systems, pages 4565–4573, 2016.

[10] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric PXing. Controllable text generation. arXiv preprint arXiv:1703.00955, 2017.

[11] Hideki Tanaka, Akinori Kinoshita, Takeshi Kobayakawa, Tadashi Kumano, and Naoto Kato. 2009. Syntaxdriven sentence revision for broadcast news summarization. In Proceedings of the 2009Workshop on Language Generation and Summarisation, UCNLG+Sum ’09, pages 39–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

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