Bengali Text generation Using Bi-directional RNN

Sheikh Abujar
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
sheikh.cse@diu.edu.bd

Abu Kaisar Mohammad Masum
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
mohammad15-6759@diu.edu.bd

S. M. Mazharul Hoque Chowdhury
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
mazharul2213@diu.edu.bd

Mahmudul Hasan
Dept. of CSE
Comilla University
Cumilla, Bangladesh
mhasanraju@gmail.com

Syed Akhter Hossain
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
aktarhossain@daffodilvarsity.edu.bd

Abstract— **Current world is growing so fast and communication between nation and different type of people with different language became part of our life. Even from buying product to our social life everything is dependent on communication. Therefore language is the most important part of human life. Though still now there is a language barrier for communication between people. But very soon language will be universal and everyone will be able to communicate in any language worldwide using the NLP technology. For that it is necessary to understand each language individually. This research proposes a new type of text generation of Bangla language using the bi-directional RNN. This technique is used to predict the next possible word in a Bangla text.**

Keywords— Bangla language, Bi-directioanl RNN, Corpus, NLP.

I. INTRODUCTION

Modern technology made our life so easy that things can be done in minutes. All those tasks that was considered as impossible are now possible for the technology. Even without knowing language people are traveling around the world depending on their smartphones. The reason behind this is Natural Language Processing. Using NLP it is possible to analysis text any find necessary information. Not only English or any particular language, NLP can be applied to any language if it has its own Unicode or computerized form. Bangla is the fifth language according to the number of user. Therefore it is important to focus on development of tool and technique using NLP to process Bangla language. In this research a very important topic was discussed which is word prediction. A corpus is built using daily life data to predict the next word. During this process word frequency was used to determine the next word.

Generally in a next possible word prediction algorithm some key topics are considered first. One of them is determination of the topic. Algorithm tries to figure out the topic user is writing. So that the algorithm will have a short listed word that is most relevant to the topic. Exception may occur, but that will be limited and based on the corpus used in the analysis. Next the algorithm finds the frequency distance of the current word and the other relevant words on the list. Even it is possible in the word level analysis where after every letter algorithm will be able to determine the next possible word. The most perfect algorithm and corpus will provide higher accuracy. As NLP is progressing rapidly it has become essential to focus on Bangla text analysis to make this language valuable worldwide.

II. LITERATURE REVIEW

As throughout all these years a lot of work has been done on this field more is needed to make it useful. Different researchers are developing new algorithms and techniques as well as new model to improve current result of text generations. Some of them will be discussed in this section.

Partha Pratim Barman et al. (2018) worked on a next word prediction model and in this research they used RNN based approach [1]. Basically they built a model of LSTM which is a special kind of RNN. They applied this work in a live chatting application where the application is able to predict the next possible word. In this model their target language was Assamese language. Hyeonwoo Noh et al. (2016) worked on a question answering system and used neural network in their system to automatically predict the possible reply for the question provided by the user [2]. In their conventional neural network (CNN) they used a parameter prediction network to create an adaptive question answering model. They applied the hashing technique to reduce the complexity of large number of parameters. Therefore answer selection will be much easier for the system.

Researcher Martin Sundermeyer and his team worked on language modeling using the LSTM neural network in 2012 [3]. The main purpose of their research was to build a neural network that is able to increase the accuracy of the analysis. They used their model on English language and as well as a large set of French language modeling task. Tomas Mikolov and his team from Microsoft Research worked on linguistic regularities [4]. In this research they tried to build a model that can predict the space word representation for a question answering system. This system includes vector representation and can automatically learn the relationship of words. With about 40% accuracy on symmetric question answering their work was the best model back then.

Andriy Mnih et al. (2013) was researching on word embeddings efficiently and in this research they used noise constrictive estimation [5]. Their main focus was to improve the result with a much simpler analysis technique. They made a comparison with Mikolov et al. (2013a) model where state of art method was used and this research uses training log-bilinear models with the help of noise-contrastive estimation of word. They also done some comparison between several other techniques on the same field. A very long time ago researcher Helmut Schmid worked on parts of speech tagging and he used neural network in this process [6]. This research proposed a new type of model on POS tagging using neural network to provide better result compared to some other most popular techniques as HMM-tagger by Cutting et al. in 1992 and a trigram-based tagger by Kempe in 1993.

Tomáš Mikolov et al. (2011) worked on modification of RNN for language modeling [7]. This research outperformed several other techniques significantly. The major complexity remained in this system was computational complexity. Their presented technique can speed up the data training and testing up to 15 times of present techniques. In this analysis they used back propagation algorithm and this technique provides better accuracy than the basic algorithms. Yoshua Bengio et al. (2013) worked on a probabilistic model for language analysis using neural network [8]. In this research they worked on such technique that will be able to find and inform the model relation between sentences and the model learns simultaneously using distributed representation of words.

Text generation is important for the sequence to sequence word classification. This paper we tried to explain a method of how to generate Bengali word next sequence using Bidirectional LSTM and RNN. Actual use of the text generation is a machine translation. It is difficult to translate the Bengali language for machine translation in NLP purpose. Because of any machine translation problem, Bengali sentence structure is not correctly working. So accurately text generation for the Bengali language is our main intention and correct Bengali sentence sequence generation.

Real utilization of the Text generation is a machine interpretation. It is hard to decipher the Bengali language for machine interpretation in NLP reason. In view of any machine interpretation issue, Bengali sentence structure isn't effectively working. So precisely message age for the Bengali language is our principle goal and right Bengali sentence sequence genetation.

III. Methodology

Language Modelling is the most significant piece of present-day NLP. There is some piece of the assignment, for example, Text Summarization, Machine Interpretation, Text Generation, Speech to Text Generation and so forth. Text generation is a noteworthy piece of Language Modelling. A well-prepared language model acquires information of the likelihood of the occasion of a word dependent on the past arrangement of words. In this paper, we talked about the n-gram modelling with a pre-trained Bengali word embedding for Text generation and make a Bi-directional Recurrent Neural Network for preparing the model. In figure1 has been given our work methodology stream.

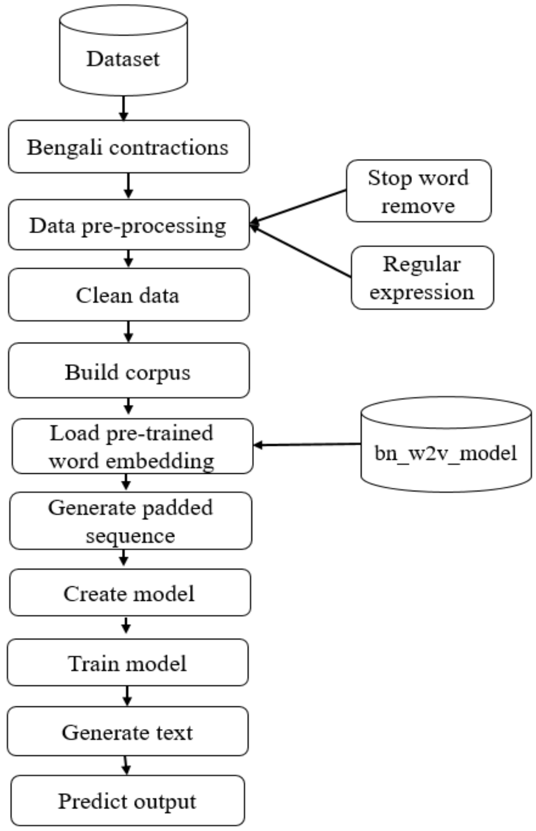

Figure1: Working flow for Text Generation

A. Data collect & pre-processing

Bengali content needs a decent dataset. We utilize our very own dataset which was gathered from online life. Our dataset contains a few sorts of Bengali post such gathering post, individual post, page post and so forth. There is some snag to gather Bengali information, for example, the structure of Bengali content. In any case, in our dataset, we endeavour to lessen the majority of that deterrent to keep a clean Bengali content. Our dataset contains text information with their sort and content outline or summary. For our working reason, we utilize just content and their outline to create a sequence of next Bengali word which is almost generate a sentence. Before getting a ready dataset for content age, we have to include Bengali compressions. Since contraction contain a short type of a word, for example "রেজি:"=" রেজিস্ট্রেশন", "ডা."=" ডাক্তার". In the wake of gathering dataset, we have to a clean dataset to create content. So for clean information, we expel whitespace, digits, accentuation from Bengali content and expel Bengali prevent words from a Bengali stop word content record. At long last, we clean the content and make a rundown which contains content with their summary. At that point, we make a corpus for text generation.

B. Add Bengali Word Embedding

Word2vec is utilized to create word embedding. There are a few reasons why we utilized word embedding, for example, the idea of a word isn't comprehended by a machine. A machine can see just paired or numerical esteem. In this way, process a language and working with normal language preparing word2vector must be required. When applying word implanting each machine can change over tokenize word to a vector where every vector speaks to the vocabulary of content archives. There are several Word embedding pre-trained models in different kinds of language but in the Bengali language few numbers of word embedding file present most of is not enough for research. One good and usable pre-trained model found which is used in our research purposes. Which name is “bn_w2v_model”.

C. N-gram Tokens Sequence

Text generation language model required an arrangement of the token and which can anticipate the likelihood next word or grouping. So need to tokenize the words. We use keras work in tokenize model which concentrate word with their record number from the corpus. After this, all content changes the arrangement of the token. In n-gram, the arrangement contains whole number token which was produced using the info content corpus. Each whole number speak to the record of the word which is in the content vocabulary.

We used word embedding which represents the word vocabulary number. Each vector number present a word. So when generating n-gram sequences each word represents by a vector number in the embedded file.

D. Pad Sequence

Every progression has a substitute length. So we need to pad sequences for making arrangement length proportionate. For this point, we use keras pad sequences function. The commitment of the learning model we use n-gram gathering as given word and the foreseen word as the accompanying word. The model is given in table 1. Finally, we can do get the data X and the accompanying word Y which is used for setting up the model.

GIVEN WORD	***NEXT WORD***
বাস্তবতা	বাস্তবতা কে
বাস্তবতা কে	বাস্তবতা কে ঘৃনা
বাস্তবতা কে ঘৃনা	বাস্তবতা কে ঘৃনা করি

Table1: Example of pad sequences

E. Proposed Model

An RNN system works incredibly products for consecutive information. Since it's can recall it yields reason for outside memory. It can foresee up and coming next succession utilizing memory and furthermore profound comprehension with its arrangement contrasted with different calculations. When it can consider the present state likewise can recollect what it gains from the past state. RNN has a long short term memory (LSTM) that recalls the past grouping content.

This paper we work with Bidirectional RNN which have two directions one is forward and another is Backward, both are the opposite direction. Output dense get the information from forward and backwards. Past information provides by backwards direction and next or predicting sequence provides forward direction.

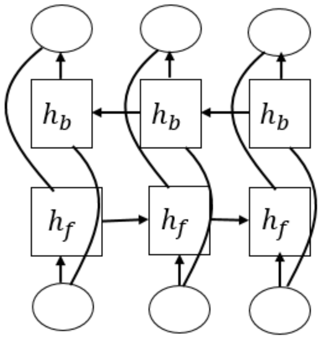

Figure2: Bi-directional RNN

The formula will be,

$$h_f = \sigma\left(W_f * \mathrm{X} + h_f + b_f\right) \quad (1)$$

$$h_b = \sigma(W_b * \mathrm{X} + h_b + b_b) \quad (2)$$

$$y = (h_f W_f + h_b W_b + b) \quad (3)$$

Here, σ = Activation Function
h_f = Forward hidden layer
h_b = Backward hidden layer
W = Weight, b=Bias

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. When working in the current state in can remember previous then using activation function it can predict the next word or sequence. Since here we use Bidirectional RNN the previous input was remember by backwards direction then for the future word or sequence prediction forward direction will help for prediction .For train our model we define keras sequential model. Add Bidirectional model with LSTM cell. For our research purpose we use 256 units and use ‘relu’ activation for LSTM cell. Set the value of Dropout function is 0.5 which helps to reduce the overfitting. Add Dense which is equal of the total word and use softmax activate function. For loss function calculation we use 'sparse categorical crossentropy' since numeric value and use 'Adam' optimization function. Finally fit the define model and set input and output sequence with verbose.

Algorithm1 for Bengali text generation using Bi-directional LSTM

```
Set function model  create(max sequence length, total words):
    input length= max sequence length-1
    declare Sequential()
    add(Bidirectional(LSTM(units, activation),input shape)
    add(Dropout(0.5))
    add(Dense(total words, activation))
    compile(loss function, optimizer)
     return model
create model(max sequence length, total words)
```

This section we show our model graphical view. Here unique id is the input of the process will continue to the Dense or output layer.

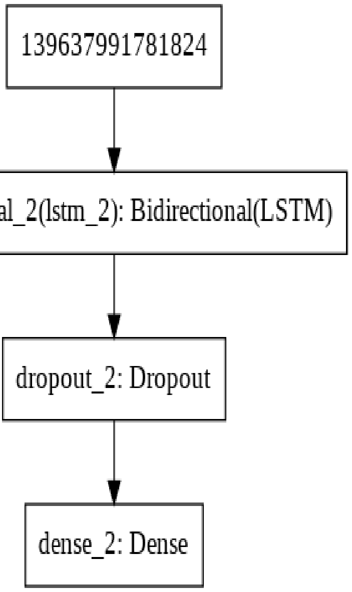

Figure3: Visualizing Bidirectional Model structure

i. *Long Short Term Memory:*

Long Short Term Memory is a piece of the Recurrent Neural Network. It's utilized to the vanishing of inclination and cancels angle. Each LSTM cell has three entryways, for example, Input Gate, Forget Gate, Output Gate and a cell state which included data by means of the doors.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (4)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (5)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (6)$$
$$c_t = f_t * c_{t-1} + i_t * \sigma(w_c[h_{t-1}, x_c] + b_c) \quad (7)$$
$$h_t = o_t * \sigma(c_t) \quad (8)$$

Here, i_t = input gate's, f_t = forget gate's ,
o_t = output gate, c_t = cell state,
h_t = hidden state, σ = activation function

ii. *Activation function:*

In model we use two activation function such as ReLu and softmax. Rectified Linear Unit utilizes for actuating the LSTM cell in Bi-directional RNN. It always put the value zero to maximum. The equation will be,

$$f(x) = \max(0, x) \quad (9)$$

Here, x= input of neuron.
The softmax activation is the calculated initiation work or logistic activation, which is utilized to manage order issues. It keeps up the yield somewhere in the range of 0 and 1 counts likelihood or probability. The recipe for softmax activation is,

$$\sigma(Z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \quad (10)$$

Here, z is the contributions to the output layer and j records the output.

IV. EXPERIMENT AND OUTPUT

Subsequent to making the model capacity we have to prepare our model. We fit the model with the present and next word. We used 80 per cent data for train and 20 per cent data for the test. For dataset set the epochs size 100 and set verbose 2 then using the fit function for train model. Train model right around 3 hours it gives better accuracy of 98.766% with loss 0.0430.We Figure4 shows model train precision graph and figure5 show loss graph of the model.

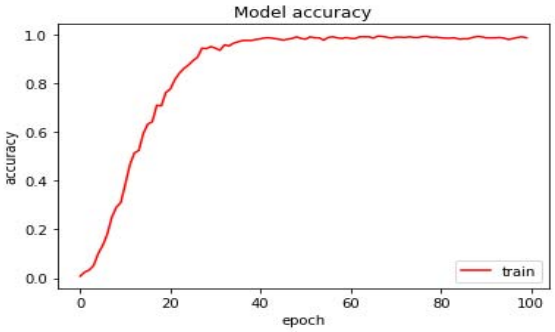

Figure 4: Model Accuracy graph for text generation

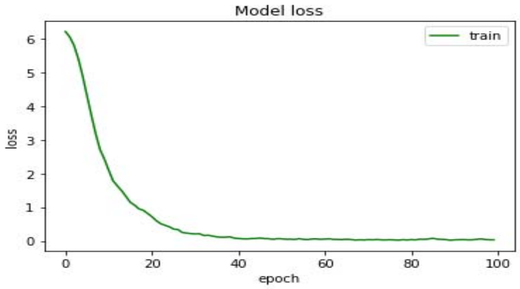

Figure5: Model loss graph for text generation

Previously we are working with one direction RNN or LSTM text sequence generation. Then use Bidirectional RNN for text sequence generation. Both perform differently in sequence generation they perform in train and loss are different. Both algorithms perform given below in the table2

APPROACH	ACCURACY	LOSS
LSTM	97%	0.04354
BI-DIRECTIONAL LSTM	98.766%	0.0430

Table2:Comparison with Bi-directional LSTM and LSTM

This test our fundamental objective to make the following arrangement of words. For output, we have made a capacity where we define Bengali pre-define word embedding for a set each word with a vector which helps to define the related word in a file with a numeric value and seed content for appearing.

We have generated a padded sequence with fixed seed word and set the length of indicator next word, call the model with the greatest arrangement length.Table3 demonstrates our test result.

Given Text	Output
শত	শত শত তরুন তরুণীর কর্মসংস্থ্যর ব্যবস্থা হবে
ভানুয়াতুতে	ভানুয়াতুতে শ্রমিক আটকা পড়েছে

Table3: Bengali Text Generation

V. Conclusion and Future work

We have proposed a decent technique for creating a programmed Bengali text generation using Bi-directional RNN. Since no model gives a precise outcome but yet our model gives better yield and maximum output is exact. Utilizing our proposed model we have effectively created a fixed length and importance full Bengali content.

There are a few imperfections in our proposed system, for example, cannot create arbitrary length content. We have to characterize the creating content length. Another deformity is we have to characterize cushion token for foreseeing next words. In our future work, we will make a programmed Bengali content generator which gives an arbitrary length Bengali content without utilizing any token or succession.

VI. ACKNOWLEDGMENT

We need to express gratefulness to our Computer Science and Engineering Department to give a superior facility for research. Extraordinarily thanks to our DIU-NLP and ML lab for supporting and helping to finish our research work.

References

1. P. P. Barmana, A. Boruaha, "A RNN based Approach for next word prediction in Assamese Phonetic Transcription," In Procedia Computer Science, Volume 143, Pages 117-123, 2018.
2. H. Noh, P. H. Seo, B. Han, "Image Question Answering Using Convolutional Neural Network With Dynamic Parameter Prediction," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 30-38, 2016.
3. M. Sundermeyer, R. Schlüter, H. Ney, "LSTM Neural Networks for Language Modeling," In 13th Annual Conference of the International Speech Communication Association, pp. 194-197, Portland, OR, USA, September 2012.
4. T. Mikolov, W. Yih, G. Zweig, "Linguistic Regularities in Continuous Space Word Representations," In Proceedings of NAACL-HLT 2013, pages 746–751, Atlanta, Georgia, 9–14 June 2013.
5. A. Mnih, K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," In Proceedings of the 26th International Conference on Neural Information Processing Systems, Volume 2, Pages 2265-2273, December 2013.
6. H. Schmid, "Part-of-speech tagging with neural networks," In Proceedings of the 15th conference on Computational linguistics, Volume 1, Pages 172-176, Kyoto, Japan, August 1994.
7. T. Mikolov, S. Kombrink, L. Burget, J. Černocký, S. Khudanpur, "Extensions of recurrent neural network language model," In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), DOI: 10.1109/ICASSP.2011.5947611, July 2011.
8. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, "A Neural Probabilistic Language Model," In The Journal of Machine Learning Research, Volume 3, Pages 1137-1155, February 2003.
9. Sanzidul Islam, et al. "Sequence-to-sequence Bangla Sentence Generation with LSTM Recurrent Neural Networks." Procedia Computer Science 152 (2019): 51-58.