# *Bangla Language Model Using RNN*

*Anik Alvi*

*Department of Computer Science and Engineering*

*Independent University, Bangladesh*

*Dhaka, Bangladesh*

*anikalviarifeen@gmail.com*

*Minhajuddin Ahmed*

*Department of Computer Science and Engineering*

*Independent University, Bangladesh*

*Dhaka, Bangladesh*

*kamranjaus@gmail.com*

## I. INTRODUCTION

Language Modeling is the task of predicting what characters, words or sentences comes next. It is one of the fundamental tasks of Natural Language Processing (NLP) and has many applications. Our model functions for predicting next words in the sentence for a given input of a set of words. The process will repeat itself until we generate a sentence of our desired length. This predictive model can learn the sequences of a problem and then generate entirely new plausible sequences for the problem domain. The model is created to run on Bangla language and the neural network that has been implemented in this model is Recurrent Neural Network (RNN)

## *II. MOTIVATION*

Bangla, one of the most important Indo-Iranian languages, is the sixth-most popular in the world. Most of the research and development works in this field has been done on English language. Due to lack of labelled data and the complexity of Bangla language, we can hardly find text generation related works with RNN for Bangla.

## *III. NEURAL NETWORK MODEL*

The neural network that we have used is Recurrent Neural Network (RNN) as because RNN have been the answer to most problems dealing with sequential data and Natural Language Processing(NLP) problems for many years. Recurrent is used to refer to repeating things. And hence an RNN is a neural network, which repeats itself. In an RNN, the value of hidden layer neurons is dependent on the present input as well as the input given to hidden layer neuron values in the past. As past hidden layer neuron values are obtained from previous inputs, we can say that an RNN takes into consideration all the previous inputs given to the network in the past to calculate the output. The type of RNN model we implemented is many-to-many vanilla RNN model with Pytorch library.

## IV. EXPERIMENT

### A. Data Collection and Description

We collected the data from one of our senior student in the department. Our corpus comprises of public dataset and web scraped data. We scraped blog posts, Bangla Wikipedia, e-books on Bangla novels, stories, etc. and used a public dataset available in Kaggle. The text corpus consisted of 700,000 articles with a total number of 39,500,468 sentences in the corpus. After preprocessing the corpus contains a total of 3,577,910 words. Among these 2,240,897 words are discarded due to single occurrences and our model uses the rest of the 1,337,032 words to create the model. Since we couldn't train the previous dataset we collected another dataset consisting 128 lines from a blog with 1,576 words .

### B. Data Preprocessing

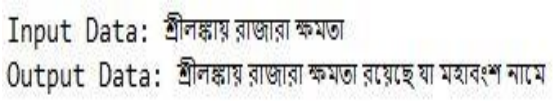
- We replace each word of the corpus with its one hot encoding.
- While scrapping data from blog post there were numerous English words that came along, we also deleted those words.
- We also rebuilt the dictionary, from the corpus we extracted distinct words and than from those distinct words we have assigned indexes as integers.
- We removed all bangla numbers from the corpus.
- We have introduced Padding for sentence length less than 6. The length of our input sequence = target sequence, the length is set to 5, which means a sentence for both input and target will contain 5 words. If

the length is less than 6 then the <word> tag will be used to fill the empty spaces of the sentence.

C. Model Training

We have designed our RNN model with the input dimension = 733, output dimension = 733 and hidden dimension = 180. We also used 150 epochs to design our model. After 150 epochs the loss was 0.1855.

## V. RESULT




We have given three inputs and wanted to see four more outputs. And the accuracy we got 81.45%.

## VI. PROBLEMS FACED

- First time pogramming in python
- No first hand experience with the pytorch library
- For this character "।" their exists 4 types of Unicode so it becomes difficult to detect
- For Bangla signature (যুক্তাক্ষর) their exists no unicode
- In the case of 3 sentences, after the padding, we have kept the input_seq and target_seq in the list but in the case of using corpus creating such large input_seq and target_seq caused cpu ram to be full and chrome to crash after processing for a long time. So python generator was used, which helped to process faster since it does not hold the entire data inside memory but gives one result at a time.
- While generating sentences during preprocessing, it considers "।" as a single sentence
- Number_filter cannot detect single character Bangla number and place it <NUM> tag But it detect for multiple digit Bangla number and replace it with <NUM> tag

## VII. CONCLUSION

We have overcome all the problem except the last error while training the 700,000. We have also generated large output data for three random inputs, which is in the text file. The output data gives somewhat meaningful sentence.

## VIII. SCOPE OF IMPROVEMENTS

Use of Long short-term memory (LSTM), Bidirectional Long Short-Term Memory (BI-LSTM) and Attention-based models.