Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 152 (2019) 51–58

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Pervasive Computing Advances and Applications - PerCAA 2019

Sequence-to-sequence Bangla Sentence Generation with LSTM Recurrent Neural Networks

Md. Sanzidul Islam[a,*], Sadia Sultana Sharmin Mousumi[a], Sheikh Abujar[b], Syed Akhter Hossain[c]

[a]*Student, Dept. of CSE, Daffodil International University, Dhaka-1207, Bangladesh*
[b]*Lecturer, Dept. of CSE, Daffodil International University, Dhaka-1207, Bangladesh*
[c]*Dept. Head, Dept. of CSE, Daffodil International University, Dhaka-1207, Bangladesh*

Abstract

Sequence to sequence text generation is the most efficient approach for automatically converting the script of a word from a source sequence to a target sequence. Text generation is the application of natural language generation which is useful in sequence modeling like the machine translation, speech recognition, image captioning, language identification, video captioning and much more. In this paper we have discussed about Bangla text generation, using deep learning approach, Long Short-term Memory (LSTM), a special kind of RNN (Recurrent Neural Network). LSTM networks are suitable for analyzing sequences of text data and predicting the next word. LSTM could be a respectable solution if you want to predict the very next point of a given time sequence. In this article we proposed a artificial Bangla Text Generator with LSTM, which is very early for this language and also this model is validated with satisfactory accuracy rate.

Keywords:
Language Modeling; Text Generation; NLP; Bangla Text; Sequence-to-sequence; RNN; LSTM, Deep Learning, Machine Learning

1. Introduction

Recurrent neural networks are types of neural network designed for capturing information from sequences or time series data. It is extension of feed forward neural network and different from other in general neural network architectures. It can handle the variable length. In earlier Schmidhuber with Hochreiter, they proposed Long Short Term Memory (LSTM) technique in 1997 [19]. It solves the hiding gradient problem by constructing some extra instruction and very efficient and better then RNN. It was like a revolution over Recurrent Network Networks (RNN). It works well on sequence based task and on any type of sequential data.

* Corresponding author. Tel.: +880 1736752047
E-mail address: sanzidul15-5223@diu.edu.bd

1877-0509
10.1016/j.procs.2019.05.026

RNN can not handle backdrop very well but LSTM can. RNN have limitation of memory but LSTM don't have any limitation of memory problen in long going dependency. RNN suffers from the same vanishing (or less notorious exploding) gradient problem as fully connected networks but LSTM can vanish gradient properly. LSTM is better than RNN because LSTMs are unequivocally intended to stay away from the long haul reliance issue. Recollecting data for significant lots of time is for all intents and purposes their default conduct, not something they battle to learn.

1.1. Dataset Properties

The neural network we made was trained with Bangla newspaper corpus. We collected a newspaper corpus of 917 days newspaper text from Prothom Alo online. The web scraping with Python was helped a lot for doing this work automatically. The training dataset contains the properties like-

- Total 917 days newspaper text.
- The daily newspaper text contains average 4500 sentences.
- 4500 sentences contains 12,500 words.
- 12,500 words contains about 15,5000 characters in average.

2. Literature Review

We are proposing a model which can generate sequence-to-sequrnce Bangla Text. There are many research and development works in this field. But hardly we can find text generation related works with LSTM for Bangla language. That's why we determined to make our own dataset and our own prediction model.

Naveen Sankaran et al. proposed a formulation, where they recognized a task which makes model as training of a sequential translation method [1]. They worked for converting words from a document into Unicode sequence directly.

Praveen Krishnan and et al. introduced an OCR system which pursues a combined architecture in seven different languages of India and a segmentation free method [2]. Their system was proposed to assist the continuous learning in the time of being it usable, like continuous user input. They worked with BLSTM method, another form of general LSTM.

A character-based encoder-decoder model which is acquired to transliterate sequence to sequence consists by Amir H. Jadidinejad [3]. The proposed an encoder built with Bidirectional Recurrent Neural Network that encodes a sequence of symbols into vector reprsentation with fixed length.

The effects of the SIGMORPHON 2016 combined task specified that the attentional sequence-to-sequence model of Bahdanauet is proper for this task [4] [5].

Robert Ostling and Johannes Bjervas proposed a model which was constructed with sequence-to-sequence artificial neural network and LSTM architecture that was a big attention to enthusiasts[6].

Yasuhisa Fujii et al. considered line-level script documentation papers in the context of multilingual OCR. They considered some alternatives of an encoder-summarizer method in the framework of an up-to-date multilingual OCR structure and they used an estimate set of several-domain streak photos from 232 languages in 30 scripts [7].

A DNN based SPSS system was made by Sivanand Achanta and et al. which is representing the audio parametric classifications of things with a single vector by sequence-to-sequence auto-encoders [8].

Mikolov et al. have established the importance of allocated images and the competence to model randomly extensive needs using Current RNN based language models[9] [10].

Sutskever et al. produce significant sentences by modifying a RNN as well through acquiring from a character-level corpus. [11]. They introduced a newly made RNN model that one works as multiplicative connectors.

Karpathy and et al. have ensured that an RNNLM is more effective of making image explanations on the pre-trained model by training the neural network model with RNN[12]. They tried to construct a model architecture of multimodal RNN.

Zhang and Lapata are also explains remarkable work using RNNs to create Chinese poetry [13]. It was a good initiative in that time which could able to generate some lines of chinese poem autometically.

Mairesse and Young suggested a phrase-based NLG method was proposed on factored LMs that can realize after a semantically united corpus [14]. They focused their crowd sourced data and shown how to work with that.

Even though active learning was similarly recommended to accept absorbing online directly from operators, the necessity for human interpreted alignments boundaries the scalability of the scheme by Mairesse et al. [15].

One more related approach throws NLG as a pattern extraction and matching problem by Angeli et al. [16].

Kondadadi et al. display that the outputs can be more improved by an SVM ranker creating them equivalent to human-authored texts [17]. They proposed a approach by end-to-end generation technique with some local decisions.

Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney suggest a novel sequence-to-sequence model to generate captions for videos. They made explanations with a sequence-to-sequence model, where frames are first read sequentially and then words are made serially [18].

3. Method Discussion

3.1. RNN Structure

The LSTM network is a special type of RNN. The RNN is neural network which attempts to model sequence or time dependent in regular behavior. This is done by output feeding back of a neural net layer in time t to the input layer at time-

$$t + 1 \tag{1}$$

It looks like this [20]-

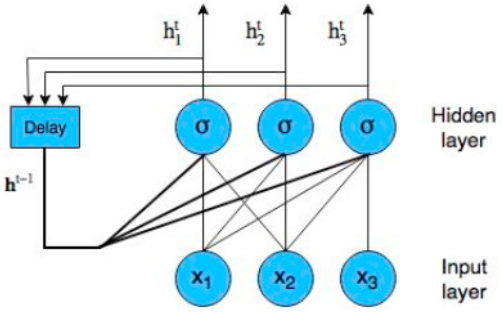

Fig. 1. Sequential nodes of Recurrent Neural Network.

Recurrent Neural Networks could be described as Unrolled programmatically at the time of training and testing. So, we can see something like [20]-

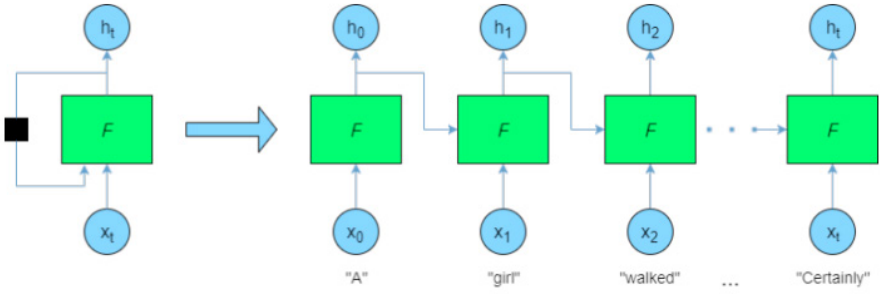

Fig. 2. Unrolled Recurrent Neural Network.

The figure showing here a new word is being supplied in every step with the previous output (i.e. ht-1) and that one also being supplied at next.

Basically, RNNs are amazingly able to handle the long-term dependencies. The issue was noticed in details by Hochreiter (1991) and Bengio, et al. (1994), who showed some pretty basic causes why it might be difficult. Thats why we will use LSTM, a better form of RNN.

3.2. LSTM Networks

The LSTMs are called Long Short Term Memory (LSTM) which are a special type of RNN, capable of learning long-term dependency problem. Thhis one was discovered by Schmidhuber and Hochreiter in 1997, and were updated and spreaded by many people in that work.
LSTMs are actually made for avoiding the long-term dependency issue. Keeping information in long periods of time is their actual default behavior, nothing what they struggle to be trained! The graphical representation of LSTM cell could be shown as below [20]-

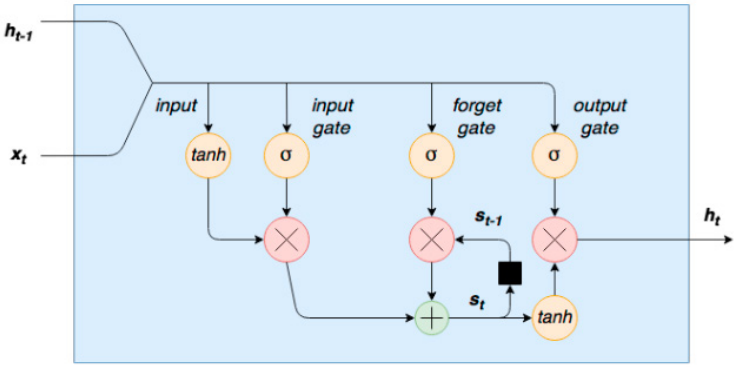

Fig. 3. LSTM Cell Diagram.

4. Proposed Methodology

4.1. Dataset Preprocessing

Working with Bangla is too much difficult still now as there has no much resource and R&D works in this field. So, processing Bengali text data a difficult task as these are too noisy and also not suitable for working with machine learning or deep learning approaches. We did some preprocessing work for making our dataset noise-free and performing its best in neural network, like-

- Removed all Bengali punctuation marks.
- Removed extra spaces and new lines
- Converted the text into utf-8 format.

4.2. Proposed Method

In general an LSTM network is complex comparative to other methods. It consume a much power in hardware ans machines capability. The whole interior activities and logic flow could be presented as below-
1) Input: Firstly, The input is squashed with the tanh activation function between-1 and 1. This could be expressed by-

$$g = tanh(b^g + t_t U^g + h_{t-1} V^g) \tag{2}$$

Where UgUg and VgVg are the previous weights of cell output and inputs. In other side bgbg is performing as an input bias. Remember, the exponents (g) is only considering as input weights.

$$i = \sigma(b^i + x_t U^i + h_{t-1} V^i) \tag{3}$$

The equation 4 is considered as output of LSTM input section-

$$g \circ i \tag{4}$$

Here the $\circ$ is elements-wise multiplication.
2) Forget state loop and gate: the output forgotten gate expression is-

$$f = \sigma(b^f + x_t U^f + h_{t-1} V^f) \tag{5}$$

The product output shows the position of previous state and forgotten gate. The equation for this calculation is-

$$s_{t-1} \circ f \tag{6}$$

The output of forgotten loop is calculated in another strategy. For different time frame-

$$s_t = s_{t-1} \circ f + g \circ i \tag{7}$$

3) Output gate: Necessary output gate is evolved as-

$$\circ = \sigma(b^\circ + x_t U^\circ + h_{t-1} V^\circ) \tag{8}$$

So that the cell final output, with tanh squashing, can be expressed as-

$$h_t = tanh(s_t) \circ O \tag{9}$$

Finally, a very common form of LSTM networks equation can be written from Colahs famous blog post [21]-

$$z_t = \sigma(W_z.[h_{(t-1),x_t}])$$
$$z_t = \sigma(W_r.[h_{(t-1),x_t}])$$

$$h_{t1} = \tanh\left(W.[(r_t * h_{(t-1),x_t}]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * h_{t1}$$

Fig. 4. LSTM networks equation.

That's how the Long-short-term-memory (LSTM) network do the operations sequentially. That's why it perform superior in any type of sequential data. The LSTM network activity flow could be presented as the figure given below. There we can notice some time evaluation term what's for LSTM is different.

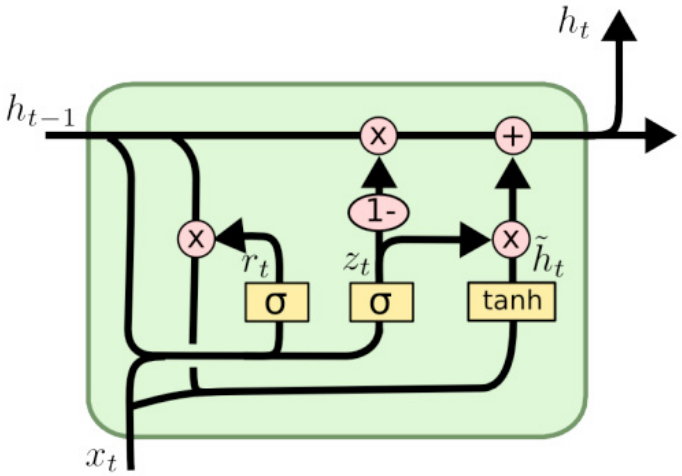

Fig. 5. LSTM networks activity flow.

4.3. Layer Description

Generally a neural network contains three layers for taking input, doing calculation and giving decision. A input embedding layer was taken as initial layer of neural network as input layer. Here a single line of text is being trained one after one and sequentially.

Then the hidden layer was taken place. It could be explain as the main LSTM layer and did it for 100 units.

The final and output layer is described now. An activation function is applied here named softmax. Softmax calculates the probability of event distribution over n events. This function generally calculates the probabilities of each target class across all possible target classes.

$$S(y_i) = \frac{e^{y_i}}{\sum_i e^{y_i}} \tag{10}$$

4.4. Model Validation

The LSTM model is little different in validation perspective. Performance determination with cross validation or train-test accuracy in general like CNN model [22] is not practical. It actually better to test the model with real data and its output. We trained only one weeks news paper corpus for having limitation of hardware limitation. And finally did test with different Bengali words, then the model generated some text according to previous text. Here are two generated Bangla sentences with our model-

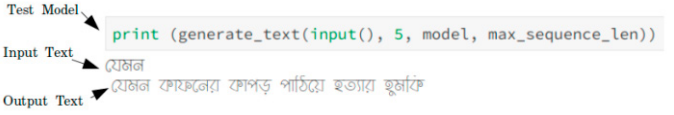

Fig. 6. Testing model (example-1).

Fig. 7. Testing model (example-2).

5. Future Work

In this paper we worked with less data, due to hardware limitations. Afterwards we will enhance our dataset. In future we will improve the model for achieving multi task sequence to sequence text generation and multi way translation like Bengali articles, caption generation. Furthermore we would aim to pursue the possibility of extending our model to Bangla regional languages. We also has plan to work with Bangla Sign Language [23] generation with sequential image data as like general people language.

References

[1] Naveen Sankaran T, Aman Neelappa, C V Jawahar, Devanagari Text Recognition: A Transcription Based Formulation, 12th International Conference on Document Analysis and Recognition, 25-28 Aug. 2013, Washington DC, USA.
[2] Praveen Krishnan, Naveen Sankaran T, Ajeet Kumar Singh, C V Jawahar, Towards a Robust OCR System for Indic Scripts, International Workshop on Document Analysis Systems, Centre for Visual Information Technology, International Institute of Information Technology Hyderabad - 500 032, INDIA, April 2014.
[3] Amir H. Jadidinejad, Neural Machine Transliteration: Preliminary Results, arXiv:1609.04253v1 [cs.CL] 14 Sep 2016.
[4] Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. The sigmorphon 2016 shared task: Morphological reinflection. In Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology. Association for Computational Linguistics, Berlin, Germany, pages 1022, 2016.
[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, Neural machine translation by jointly learning to align and translate, CoRR abs/1409.0473, 2014.
[6] Robert Ostling and Johannes Bjerva, SU-RUG at the CoNLLSIGMORPHON 2017 shared task: Morphological Inflection with Attentional Sequence-to-Sequence Models, arXiv:1706.03499v1 [cs.CL] 12 Jun 2017.
[7] Yasuhisa Fujii, Karel Driesen, Jonathan Baccash, Ash Hurst and Ashok C. Popat, Sequence-to-Label Script Identification for Multilingual OCR, Google Research, Mountain View, CA 94043, USA, arXiv:1708.04671v2 [cs.CV] 17 Aug 2017.
[8] Sivanand Achanta, KNRK Raju Alluri and Suryakanth V Gangashetty, Statistical Parametric Speech Synthesis Using Bottleneck Representation From Sequence Auto-encoder, Speech and Vision Laboratory, IIIT Hyderabad, INDIA, arXiv:1606.05844v1 [cs.SD] 19 Jun 2016.
[9] Tomas Mikolov, Martin Karafit, Lukas Burget, JanC ernocky, and Sanjeev Khudanpur, Recurrent neural network based language model, In Proceedings on InterSpeech, 2010.
[10] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan H. Cernocky and Sanjeev Khudanpur, Extensions of recurrent neural network language model, In ICASSP, 2011 IEEE International Conference on, 2011.

[11] Ilya Sutskever, James Martens and Geoffrey E. Hinton, Generating text with recurrent neural networks, In Proceedings of the 28th International Conference on Machine Learning (ICML-11), ACM, 2011.
[12] Andrej Karpathy and Li Fei-Fei, Deep visual semantic alignments for generating image descriptions, CoRR, 2014.
[13] Xingxing Zhang and Mirella Lapata, Chinese poetry generation with recurrent neural networks, In Proceedings of the 2014 Conference on EMNLP, Association for Computational Linguistics, October, 2014.
[14] Francois Mairesse and Steve Young, Stochastic language generation in dialogue using factored language models, Computer Linguistics, 2014.
[15] Francois Mairesse, Milica Gasic, Filip Jurccek, Simon Keizer, Blaise Thomson, Kai Yu and Steve Young, Phrase-based statistical language generation using graphical models and active learning, In Proceedings of the 48th ACL, ACL 10, 2010.
[16] Gabor Angeli, Percy Liang, and Dan Klein, A simple domainindependent probabilistic approach to generation, In Proceedings of the 2010 Conference on EMNLP, EMNLP 10, Association for Computational Linguistics, 2010.
[17] Ravi Kondadadi, Blake Howald, and Frank Schilder, A statistical nlg framework for aggregated planning and realization In Proceedings of the 51st Annual Meeting of the ACL, Association for Computational Linguistics, 2013.
[18] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell and Kate Saenko, Sequence to Sequence Video to Text, arXiv:1505.00487 [cs.CV] or arXiv:1505.00487v3 [cs.CV] 19 Oct. 2015.
[19] Hochreiter, Sepp, and Jrgen Schmidhuber. Long short-term memory. Neural computation 9.8 (1997): 1735-1780.
[20] Adventuresinmachinelearning.com, Keras LSTM tutorial How to easily build a powerful deep learning language model, 2018. [Online]. Available: http://www.adventuresinmachinelearning.com/keras-lstm-tutorial/ . [Accessed: 14- Aug- 2018].
[21] Colah.github.io, Understanding LSTM Networks, 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ . [Accessed: 14- Aug- 2018].
[22] Islam, Sanzidul, et al. "A Potent Model to Recognize Bangla Sign Language Digits Using Convolutional Neural Network." Procedia computer science 143 (2018): 611-618.
[23] Islam, Md Sanzidul, et al. "Ishara-Lipi: The First Complete MultipurposeOpen Access Dataset of Isolated Characters for Bangla Sign Language." 2018 International Conference on Bangla Speech and Language Processing (ICBSLP). IEEE, 2018.