

Text Generation using Neural Models

Khushboo Lathia, Mahesh Maurya

Abstract: *The use of automatically generated summaries for long/short texts is commonly used in digital services. In this Paper, a successful approach at text generation using generative adversarial networks (GAN) has been studied. In this paper, we have studied various neural models for text generation. Our main focus was on generating text using Recurrent Neural Network (RNN) and its variants and analyze its result. We have generated and translated text varying number of epochs and temperature to improve the confidence of the model as well as by varying the size of input file. We were amazed to see how the Long-Short Term Memory (LSTM) model responded to these varying parameters. The performance of LSTMs was better when the appropriate size of dataset was given to the model for training. The resulting model is tested on different datasets originating of varying sizes. The evaluations show that the output generated by the model do not correlate with the corresponding datasets which means that the generated output is different from the dataset.*

Keywords: *Text generation, recurrent neural networks, LSTM, GRU, Adversarial training Machine translation*

I. INTRODUCTION

Sequence prediction problems were around for a long time. They are considered as one of the hardest issues to resolve in the data technology enterprise. These encompass an extensive range of troubles; from predicting income to locating patterns in inventory markets' data, from information movie plots to recognizing your way of speech, from language translations to predicting your next phrase in your iPhone's keyboard, from short testimonies to writing 50,000 phrase novels, machines are churning out phrases like by no means before. There are tons of examples available on the net in which builders have used machine gaining knowledge of to write portions of textual content, and the consequences range from the absurd to delightfully humorous. Thanks to essential advancements in the discipline of Natural Language Processing (NLP), machines are able to apprehend the context and spin up tales all with the aid of themselves.

Natural language generation is a crucial building block in lots of applications, including device translation, talk generation, and picture captioning. While those packages display the realistic rate of producing coherent and sizeable sentences in a supervised setup, unsupervised textual content generation, which objectives to estimate the distribution of real text from a corpus, remains hard. Previous strategies that given in advance determined words, typically be bothered by exposure bias, i.e., the discrepancy between schooling and inference tiers.

Revised Manuscript Received on December 12, 2019.

* Correspondence Author

Khushboo Lathia, Computer Engineering Department, MPSTME, NMIMS University, Mumbai, India. Email: lathiakl.nmims.edu

Mahesh Maurya, Computer Engineering Department, MPSTME, NMIMS University, Mumbai, India. Email: mahesh.mourya@nmims.edu

During inference, each word is generated in series based on formerly generated phrases, while for the duration of education floor-reality words are used for every time step.

Recently, adversarial training has emerged as a powerful paradigm to address the aforementioned issues. The generative adversarial network (GAN) matches the distribution of synthetic and real data by introducing a two-player adversarial game between a generator and a discriminator. The generator is trained to learn a nonlinear function that maps samples from a given (simple) prior distribution to synthetic data that appear realistic, while the discriminator aims to distinguish the fake data from real samples. GAN can be trained efficiently via back-propagation through the nonlinear function of the generator, which typically requires the data to be continuous (e.g., images). However, the discrete nature of text renders the model non-differentiable, hindering use of GAN in natural language processing tasks.

Recurrent neural network language models (RNNLMs, Mikolov et al., 2011) represent the state of the art in unsupervised generative modeling for natural language sentences.

The paper is divided into 5 sections. Section II describes various use cases for text generation, section III describes the architecture of text generating model, section IV describes the proposed Architecture and section V describes work done and results using different datasets and section V describes the conclusion and future work.

USE CASES OF TEXT GENERATION

A. NEURAL MACHINE TRANSLATION (NMT)

As the name says, NMT is automatically translating the input sentences from one language to another language. Neural Machine Translation makes use of a single neural network consisting of two RNNs:

Encoder RNN: Extracts all of the pertinent information from the source sentence to produce an encoding

Decoder RNN: A language model that generates the target sentence conditioned with the encoding created by the encoder. Its architecture uses a simple seq2seq model and is trained on a sample of pairs of source and translating sentences.

It's is well explained by Zhen yang et.al.in his paper [12]

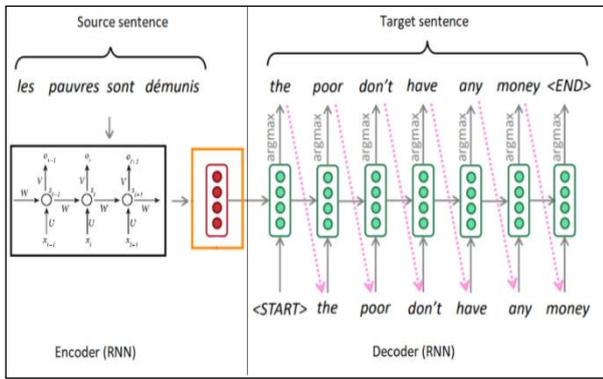


Fig.1: Seq2seq architecture for machine translation [20]

B. Image Captioning

Image Captioning is used when we want to generate description of the image. Given an image as an input, the model generates its corresponding output as the description related to the objects in the image. As our input is an image a CNN architecture can be used as a feature extractor along with an RNN for the text generation.

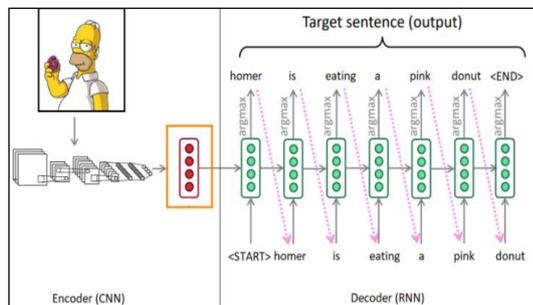


Fig.2: CNN based architecture for image captioning

C. Adversarial text generation

Apart from using auto-encoders for text generation task, an approach of supervised classification for training can be used. In this approach, classifier will be trained to determine whether the generated text has a desired property or not. In this approach a classifier (also called a discriminator) is trained on top of generated text. They encode text using a variational auto-encoder and add an additional label (feature) in the encoding that is used by the decoder when decoding. Then, the discriminator on the other hand of the decoder determines whether the decoded text has the proper label. The dotted line indicates learning signal feeding backward from the classifier.

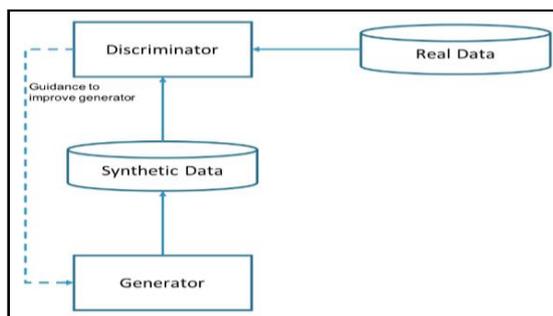


Fig.3: Basic GAN architecture for generating synthetic data

Gan's consist of a generator and a discriminator (classifier). The generator's input is a random seed which creates synthetic data out of it such as images or sentences. The discriminator's task is to tell the difference between the synthetic data and real examples of that data. The results of the discriminator can serve as the error value for the generator, so that if the discriminator can easily tell the real data from the synthetic data, the error for the generator is high.

II. BACKGROUND

RECURRENT MODELS

Recurrent neural networks suffers from the issue called short term memory, i.e., they are unable to memorize the long sequence pattern.so, of we want to generate text from a large corpus then there is a possibility that they might skip an important information. While using back propagation, RNN suffers from vanishing gradients. Gradients are used to update the weights of a neural network. The vanishing gradient problem arises when the gradient shrinks as it back propagates over time. If a gradient value becomes extremely small, it doesn't contribute too much learning.

In order to overcome this issue LSTM and GRU based RNN neural network can be used. They have internal mechanisms called gates that can regulate the flow of information. These gates help the network to manage which information to keep and which to forget.

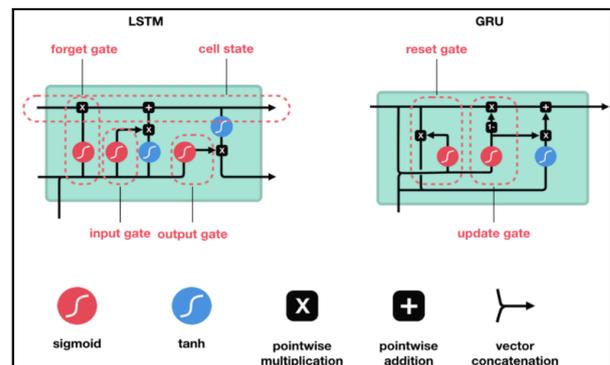


Fig .4: Architecture of RNN and GRU [22]

III. PROPOSED WORK

Possible Approaches for proposed work

- Use of text Generating Algorithms: Widely used text generating algorithms are RNN and its variants. We have used RNN, LSTMs and GRU for text generation task.
- Text Classifier: A CNN (Convolution Neural Network) architecture has been used for text classification task.
- Data Acquisition: corpus of Wikipedia articles, Shakespeare's sonnets, and machine translation datasets for translation of English to French, Hindi and Gujarati has been acquired.

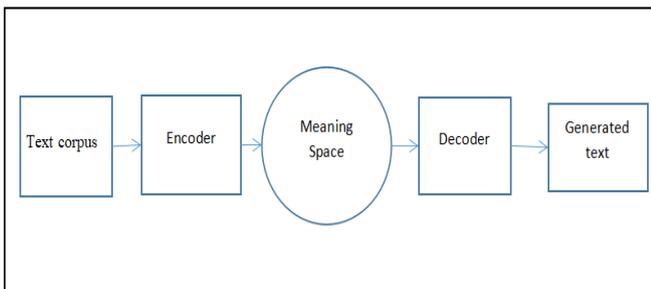


Fig.5: Basic flow for Text Generation

Fig.5 shows the flow for text generation and Fig.6 show the flow of the proposed model.

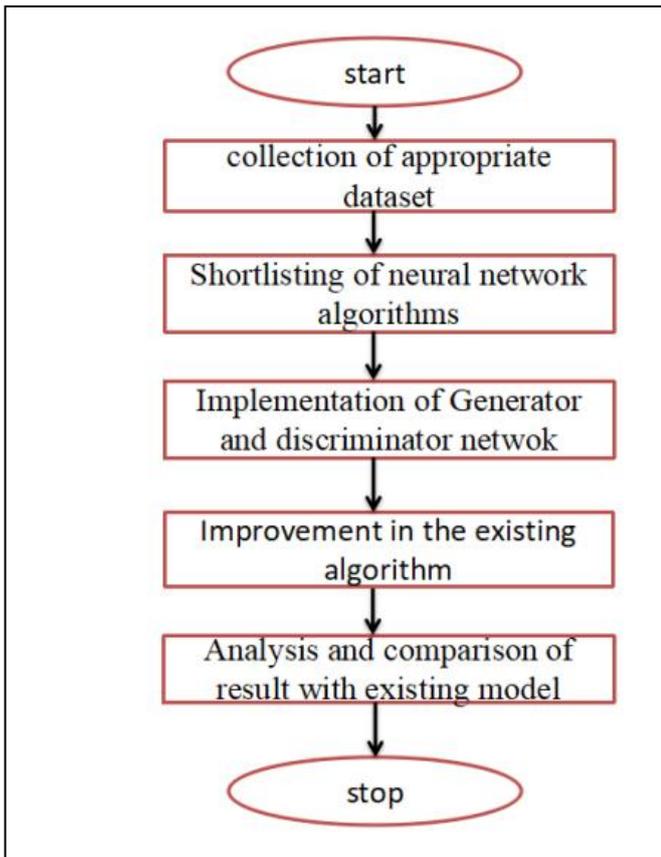


Fig.6: Flow of proposed work

IV. WORK DONE AND RESULTS

In our experiments we have used LSTM, GRU and adversarial training for text generation. There has been great result using LSTM and GRU. They performed really well given the corpus. We implemented it for neural machine translation and text generation given an input string. We observed its performance varying no. Of epochs, adding temperature to improve the confidence of the generated output and giving different size of datasets. On comparing the results, we came to know that for a neural network to learn efficiently it has to be provided with larger dataset so that it can learn to generate text grammatically correct. Even if the number of epochs are very less then also the network fails to learn and generate grammatically correct sentences.

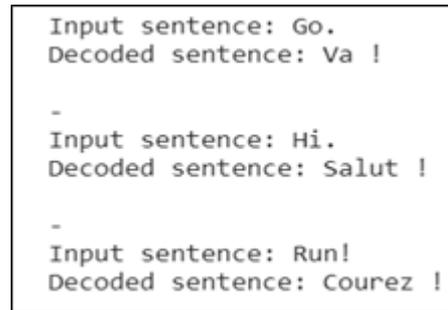


Fig 7: NMT with larger dataset (11mb)

When we trained our model on datasets of different sizes we were amazed to see how model reacted to training with varying size of corpus. In Fig 7, the model was trained over dataset of English-French of 11MB size. The results were quiet promising and translated the text correctly in French from English.

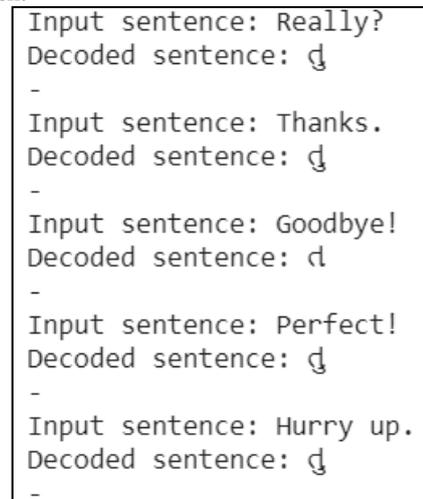


Fig 8: NMT with smaller dataset (7kb)

When we tested our architecture over the Gujarati dataset of size 7KB the model was not able to learn efficiently and did not translate to English words to Gujarati appropriately. Fig 8 shows the result how the model decoded to input sentences.

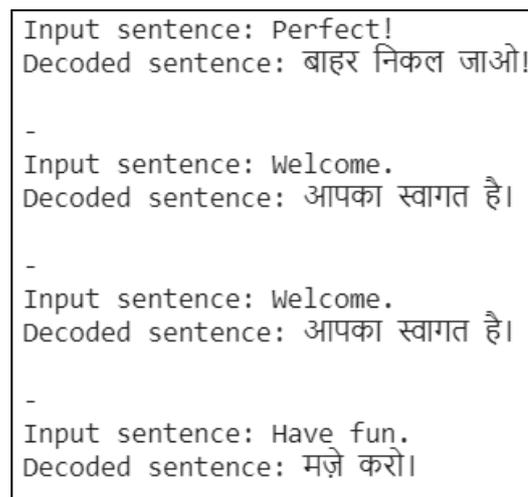


Fig 9:- NMT with small dataset (327kb)

Similarly, we again tested the model over a corpus of size greater than Gujarati corpus and smaller than French corpus i.e. of. The model was not able to decode the input sentences sometimes while for some input sentences it did decode it properly.

```

MIONE:
What say you to a father?

FLORIZEL:
Here comes a man; and therefore comes the boy he was a man--
As if the world as I live to see the crown,
And better 'twere to be a child, and do your honours
Which he did but be so still with her silver sound.

LADY ANNE:
And I will make a dead man in prison all the consul's worthiest:
The world is to be seen: I will not stay the seas
Which he did but the like success be remembered,
Shall with my cousin Buckingham, and the rest,
When the sea which you are not to be spoken with thee:
Have I not reason what with the consuls and tells his wife.

KATHARINA:
I know not what to say the truth of your company.
    
```

Fig 10-Generated text with Starting string ‘M’ and 20 epochs

No. of epochs plays an important role while training the model. No. of epochs is proportional to the confidence of a model saying how confident the model is with the generated output. When the model was trained to generate Shakespeare’s sonnets given the Shakespeare’s corpus the model was not able to generate grammatically correct sentences because the model was trained only with 20 epochs. Fig8 shows how the generated sonnets came out to be. When the same model was ran over 50 epochs the result came out amazingly nice. It was non-discriminative with the original Shakespeare’s sonnets. Here we intuitively told our model to generate sonnets given a particular starting key. Fig 10 shows the amazing result generated by the model.

```

MERLANIO:
The commons hath her own gentlemen.

SOMERSET:
But he, his own affections'd.

LUCIO:
I say the lark and loathed one.

POLIXENES:
Press me not, madam?

VOLUMNIA:
Thou art the case nor on my heart.

PRINCE EDWARD:
If that be mine, if you will take honourable.
When I have said, good aunt, go with him.
    
```

Fig.11-Generated text with Starting string ‘M’ and 50 epochs

Epoch 15 Batch 0 Loss 1.1989 Epoch 15 Loss 1.1958 Time taken for 1 epoch 23.18953013420195 sec	Epoch 45 Batch 0 Loss 0.4086 Epoch 45 Loss 0.5200 Time taken for 1 epoch 23.458430767059326 sec
Epoch 16 Batch 0 Loss 1.1391 Epoch 16 Loss 1.1500 Time taken for 1 epoch 23.132075548171997 sec	Epoch 46 Batch 0 Loss 0.4041 Epoch 46 Loss 0.5192 Time taken for 1 epoch 23.281251430511475 sec
Epoch 17 Batch 0 Loss 1.0954 Epoch 17 Loss 1.1607 Time taken for 1 epoch 23.174434661865234 sec	Epoch 47 Batch 0 Loss 0.3994 Epoch 47 Loss 0.4905 Time taken for 1 epoch 23.268502950668335 sec
Epoch 18 Batch 0 Loss 1.0709 Epoch 18 Loss 1.1401 Time taken for 1 epoch 23.168772220611572 sec	Epoch 48 Batch 0 Loss 0.3922 Epoch 48 Loss 0.5014 Time taken for 1 epoch 23.420222759246626 sec
Epoch 19 Batch 0 Loss 1.0504 Epoch 19 Loss 1.0788 Time taken for 1 epoch 23.222381830215454 sec	Epoch 49 Batch 0 Loss 0.3858 Epoch 49 Loss 0.4829 Time taken for 1 epoch 23.332348585128784 sec
Epoch 20 Batch 0 Loss 1.0271 Epoch 20 Loss 1.0852 Time taken for 1 epoch 23.008175134658813 sec	Epoch 50 Batch 0 Loss 0.3842 Epoch 50 Loss 0.4798 Time taken for 1 epoch 23.417008638381958 sec
Epoch 21 Batch 0 Loss 1.0087 Epoch 21 Loss 1.0631 Time taken for 1 epoch 23.0411593914032 sec	
Epoch 22 Batch 0 Loss 0.9796 Epoch 22 Loss 1.0190 Time taken for 1 epoch 22.920073747634888 sec	

Fig.12: Loss and time taken for 50 epochs

As the number of Epochs have increased the network learned better to form a sentence that is grammatically sound and logical. Increasing the no of epochs corresponding to the size of dataset helps model to learn better and gives a more accurate outcome in response to the text generation.

The Fig.10 and Fig.11 shows that the network trained with less epoch generated poor results whereas those with balanced epochs were grammatically correct. Similarly in NMT the model translated sentences correctly only when the size of dataset was big enough for the model to learn.

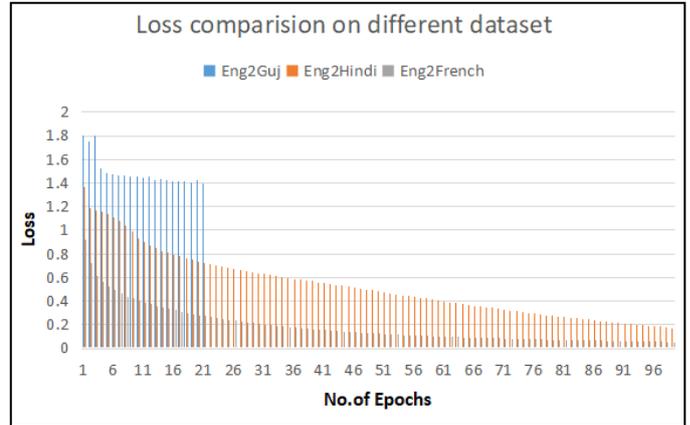


Fig 13: Comparative study of LSTM models on varying datasets

An LSTM based Machine Translation Model was trained and tested on 3 corpus of varying sizes. The comparison of their loss function same has been shown in the above figure while Table 1 shows the comparative parameters and they outcome.

Table 1: Comparative study of Generative model

Model	Dataset	Size of Dataset	No.of Epochs	Outcome
LSTM+GRU	English-to-French	10.7 MB	100	successful
	English-to-hindi	324 KB	100	Not very accurate
	English-to-Gujarati	10 KB	20	Not successful

V. CONCLUSION AND FUTURE WORK

We show the generation of text using variants of RNN model and how change in various attributes affect the generation of text and translation task. Further we would like to generate text using Adversarial training (GANS) and generate varied types of text using the same.

REFERENCES

1. Kenki Nakamura and Qiang Ma, "Context aware image generation using generative adversarial networks", IEEE International Symposium on multimedia, pp. 516-523, 2017.
2. Clayton connors and Rangaraju Vatsavai, "Semi-supervised deep generative model for change detection in very high resolution imaginary", IEEE, pp. 1063-1066, 2017.

3. Paul D. Gader and Mohamed Ali Khabou, "Automatic Feature Generation for Handwritten Digit Recognition", IEEE Transactions on pattern analysis and machine intelligence, vol 18, no 12, december 1996 .
4. Asim Munawar, Phongtharin Vinayavekhin and Giovanni DeMagistris, "Limiting the reconstruction capability of generative neural network using negative learning",
5. International workshop on machine learning for signal processing Tokyo, japan, IEEE, pp. 25–28, SEPT 2017.
6. Li Chen et.al. "Beyond Human Recognition: A CNN-Based Framework for Handwritten Character Recognition", 3rd IAPR Asian Conference on Pattern Recognition, IEEE, pp. 695-699, 2015
7. Junsuk Choe et.al. "Face Generation for Low-shot Learning using Generative Adversarial Networks", IEEE International Conference on Computer Vision Workshops, pp. 1940-1948, 2017.
8. Subhajit Choudhury et.al. "Text to image generative model using constrained embedding space mapping", IEEE international workshop on Machine Learning for signal processing, IEEE, SEPT. 25–28, 2017.
9. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NIPS, 2014, pp. 2672–2680.
10. M. Arjovsky, S. Chintala, and L. Bottou. 2017. Wasserstein GAN. arXiv preprint .
11. Kirtipreet Kaur, Er. Deepinderjeet Kaur, "Word Summarization from a Paragraph Using Word Sense Disambiguation", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 8, pp. 123-127, August 2015.
12. ZhenYang et.al. "Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets", arXiv preprint, 2018.
13. Anusha Bagalkotkar et.al. "A Novel Technique for Efficient Text Document Summarization as a Service", Third International Conference on Advances in Computing and Communications 2013
14. Ofir Press et.al. "Language Generation with Recurrent Generative Adversarial Networks without Pre-training", arXiv preprint, 2017.
15. Samuel R. Bowman et.al. "GENERATING SENTENCES FROM A CONTINUOUS SPACE", pp. 1-10, ICLR 2016,
16. Adversarial Learning for Neural Dialogue Generation, <https://arxiv.org/pdf/1701.06547.pdf>
17. MaskGAN: Better Text Generation via Filling in the Blank, <https://arxiv.org/pdf/1801.07736.pdf>
18. Understanding LSTM Networks, <http://colah.github.io/posts/2015-08-Understanding-LSTM/>
19. Generating Sentences from a Continuous Space, <https://arxiv.org/abs/1511.06349>
20. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, <https://arxiv.org/pdf/1609.05473v3.pdf>
21. Stanford course Natural Language Processing with Deep Learning (CS224n)
22. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>

AUTHORS PROFILE



Khushboo Lathia, MTech Computer Science from MPSTME, NMIMS, Mumbai.



Mahesh Maurya, Ph.D., M. Tech, B.Tech. in Computer Engineering. He has published more than twenty papers in national, international conferences and journals. His research area is distributed computing, parallel computing, system security, Big data and analytics, Artificial Intelligence, machine learning and deep learning. He is a life member of ISTE.