

Lyrics Generator with Deep Neural Network (TensorFlow)

Nausheen Samiha - ID: 18101108
Maliha Tabassum Mithila - ID: 18101459
Shahriar Hossain - ID: 21141036

Abstract

Now a days, Music Industry is booming in many countries. There are a many young musicians who are just about to steeping into this industry but stumble upon one thing. Their original lyrics. However, Lyrics is basically generating texts. And this task of generating text is not new for Natural Language Processing. Natural Language Processing is a pretty handy tool in terms of analyzing human language. And Deep Neural Network is what it makes it more amazing. With the help of deep neural network, text generating by the system has become quite accurate. Deep Neural Network is a basic branch of Machine Learning. We are going to use RNN in this project which is also a type of Deep Neural Network.

1 Introduction

In the world of entertainment and social media like tik-tok and Instagram, music production is a popular topic among young generation especially among young singers. That's why we have come up with the idea of Lyrics Generator with tensor flow which is a part of natural language processing. Our Lyrics generator can be used in any application or website that produces music or hires new singers. With the help of Deep neural network and NLP next generating system has been way more precise than before. From the gigantic field of Deep Neural Network, we are going to use RNN to design our Lyrics generator with Tensor Flow.

2 Background/Literature-survey

The Neural network model is the fulcrum of the AI revolution. RNN is a very powerful sequence model but it does not have a spread wise use because it is very difficult to train them. It is a

pattern recognizing task and it has two main drawbacks when the training dataset is small, it is unable to capture information when there is a large class number. For applying neural network in the real world we use tensor flow. Tensor flow is used in RNN because it has a beginner friendly and flexible architecture to generate idea for code and publication it will be used in our project for various use.

3 Ideas/Plans

As our system is going to generate lyrics, it is basically text generating using Deep Neural Network. For this project we have to collect a dataset which will be some English songs. We will collect some songs of the famous Band "Imagine Dragons" and Famous Singer "Elvis Presley". To make a model, we will use word based Tokenizer and tokenize the words. In the data preprocessing phase, As the sequence of words is important for a meaningful sentence, we will use N grams Sequence for this model. N grams sequence basically predicts the next word based on the exact previous word. After that, we will need to pad the dataset into a single length. As we are using Deep Neural Network, if the input size is different, it will fail to compute for the neurons less accurately. We will pad every sequence into the size of the longest sequence. After padding, we will need to label every N gram sequence using hot encoded arrays for labeling. After that, we will need to build a model. We have to build a sequential model and then we will add some layers in the model. We will use 4 layers. 1. Embedding (input layer), 2. GlobalAveragePooling1D, 3. Bidirectional LSTM, 4. SoftMax (Classifier). Here, an embedding layer is needed as we are building a sequential model. Then we are using Bidirectional LSTM, which is a kind of RNN. Then we will use "SoftMax" as the classifier. We will use "Adam" Optimizer and

“Categorical Cross Entropy” the loss function. After training the model for 100 epochs, we will be able to predict words and by predicting words, we will generate lyrics.

4 Potential Challenges

Building up a project comes with many challenges. While developing the planned project, we may also face the below mentioned challenges. In the first place, our model is predicted to have higher loss function if the model is built upon having aforementioned features in the previous section. Loss function in this project is predicted to be close to 1.

Consequently, this leads us to losing more data while training our model. In order to resolve this issue, two solutions crossed our minds. It is also important to mention that we predicted to have more than 80% accuracy for our model. At first, based on our research we presumed that adding more layers can reduce loss function but everything comes at a cost. Hence, adding more layers will reduce loss function along with the accuracy. Therefore, we thought of another solution which will maintain accuracy and other factors as before and a better approach is to limit the number of words while performing prediction. By dealing with a least number of words at a time of prediction will reduce the loss function gradually.

Next, our assumption remarks our model to be highly biased if we implement all the features as per description. By the term highly biased, we refer that our model will always show the result similar to the dataset we feed to train our model. It is our notion that our model will provide the same keywords it will find from the dataset. With the aim of solving the problem we need to refine our dataset in such a way that our model will look for some other method to show the result. As we are unable to present a clear solution, therefore, it is still an area of investigation.

Later, as we planned to use hot encoded arrays for labeling, each label array will have the length of the vocabulary size, thus, it requires a huge amount of memory. If our dataset is a lot bigger, then it may take even 1, 2 or even more Terabytes of memory.

5 Conclusion

In this paper, we proposed a unique model which will generate lyrics. It can turn out to be a handy tool for the musicians most importantly for the beginners who may acquire ideas from it. The proposed model is expected to perform well in all cases. Our expectation is to have a good accuracy which we mentioned in previous sections that can be gained if other factors remain the same as discussion. Building up this project in the future may bring good results.

6 Bibliography

- [1] Ilya Sutskever, James Martens, Joffrey E. Hinton. 2011. Generating Text with Recurrent Neural Network. [Generating Text with Recurrent Neural Networks \(icml.cc\)](https://arxiv.org/abs/1308.0850)
- [2] Yang Li, Quan Pan, Suhan Wang, Tao Yang, Erik Cambria. 2018. A Generative Model for category text generation. <https://doi.org/10.1016/j.ins.2018.03.050>