Course Project Proposal - Rock Lyrics Generation

Junzhe Wang
jwang169@stevens.edu
Stevens Institution of Technology
Hoboken, NJ

ABSTRACT

In this project, I introduce a method and discuss results of text-based LSTM language model for automatic rock lyrics composition. The proposed model is designed to learn relationships within text documents, and creates lyrics with meaningful, interesting story and lyrical patterns for rock music. First, I perform analysis of different lyrics genre. Results show word frequency and the length of sentence vary from different music genre. Second, I implement web scraping to collect lyrics of selected artists as the training data set. Finally, I build the text generation model which can be used for producing rock lyrics, and evaluate the model by computing BLEU score.

KEYWORDS

LSTM, lyrics, RNN, web scraping

1 INTRODUCTION

Rock music has embodied and served as the vehicle for cultural and social movements. My objective is to study the problem of computational creation of rock lyrics. Unlike many earlier styles of popular music, themes of rock lyrics are rather diverse in nature ranging from love, betrayal, sex, social and political themes such as rebelling against an establishment. Unlike rap lyrics, which often have a similar structure (similar word length per line and similar rhyming schemes), rock lyrics have less restrictions on the rhyming part, as Twenty one pilots says: "I wish I didn't have to rhyme every time I sang".

My interest in this problem is motivated by two different perspectives. First, I am interested in analyzing the formal topics of rock lyrics and in developing a model that can lead to generating artistic work. Second, in June 2016, Google published Dope Learning and released the DeepBeat program which uses machine learning techniques to generate rap lyrics by combining lines from existing rap songs. The rap lyrics generator has been deployed as an online tool called DeepBeat. Different from their approach, I approach the lyrics-generation problem by a word-by-word construction instead of line by line, so as to increase novelty rather than rhyming. This revolves around creating system that produce text that makes sense in content, grammar, lexical choice, and overall flow. The system also need to produce output that is non-repetitive, so it need to combine short sentences with the same subject. Assuming uses have a certain concept in their mind, formulated as a topic as a word, and their information need is to find the missing content, composing a song.

The project is organized as follows. In Section 2, I start with a brief discussion of lyrics genre analysis. Next, I introduce the data set that I have used, and describe the preparation of the training data set. I start in Section 3 by defining the framework of the proposed neural language generator. I introduce the word-based RNN and semantically controlled LSTM cell in section 3.1, then discuss how to extend it to a deep structure in Section 3.2. Section 3.3 proposes the GANs framework to improve the generation model. Training details are described in Section 3.4. Section 4 presents a potential method for evaluation of the model.

2 RELATED WORK

The study of human-generated lyrics is relevant for various subfields of computers science. Relevant literature can be found under domains of computational creativity, information extraction and natural language processing. Additionally, methods can be found under the domain of machine learning, for instance, in the emerging field of deep learning. Recent advances in recurrent neural network-based language models(RNNLM)(Mikolov et al., 2010[2];Mikolov et al., 2011[3] have demonstrated the value of distributed representations and the ability to model arbitrarily long dependencies. Zhang and Lapata(2014)[4] also describes an interesting work using RNNs to generate Chinese poetry, while DopeLearning[1] recombines lines from rap songs.

3 LYRICS ANALYSIS

In this section, I first compare the difference of lyrics among different music genres. This information will be the basis for extracting useful features for generating lyrics.

I use 380,000 lyrics in different genre, analyze them by word frequencies and sentence length, and visualize the most frequent word by word cloud.

This information will explain the major features of generating the lyrics words and sentences. For further information, the data is available from https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics.

3.1 Lyrics by genre

Since words like 'i', 'you', and 'the' are commonly used throughout all genres, all the consecutive statistics exclude them(together with some other common words). In order to see what are the differences between genres for each of these words, I plot the genre specific statistics with exact counts for each word.

'Love' is the most used of the uncommon words in most of the genres, while each genre has its own unique word, such as 'shit' and 'fuck' in Hip-pop, 'once' in R&B, and 'away' in Rock.

3.2 Rock Lyrics

This project will focus on the lyrics generation on Rock, and a typical rock song follows a pattern of alternating verses and choruses. The length of sentences and the word frequency also varies from

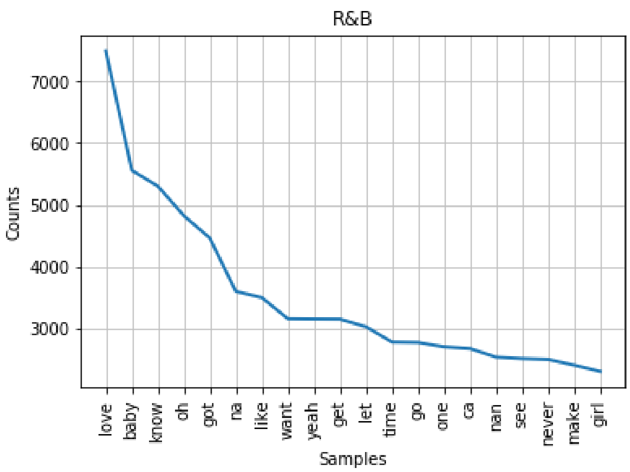

Figure 1: Most frequently used words in R&B.

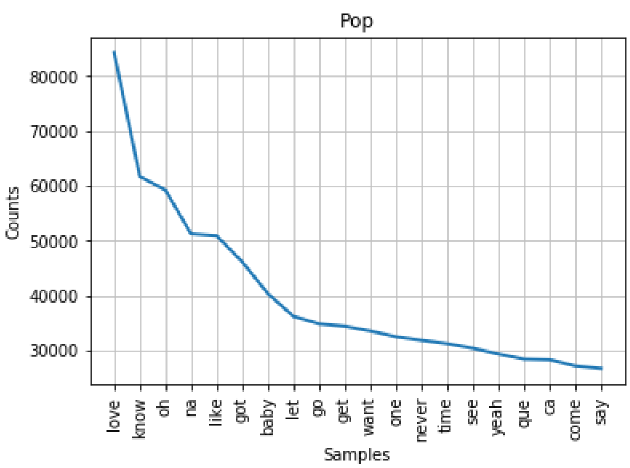

Figure 2: Most frequently used words in Pop.

different topics. The training data only covers lyrics of rock music, and the length of output text will be restricted accordingly.

3.3 Data

I compiled a list of English-speaking rock artists based on my personal preference, and scraped all their songs available from a lyrics sitehttps://www.lyricsfreak.com/. In total, I have around 300000 lines from 8500 songs. I then process a subset of those lyrics by removing the headers, removing unnecessary punctuation and white space, and lowercasing all the alphabet characters. Finally, I split the content of the lyrics into chorus verse sets.

4 THE NEURAL LANGUAGE GENERATOR

The generation model proposed in this project is based on a recurrent neural network architecture(Mikolov et al., 2012) in which a

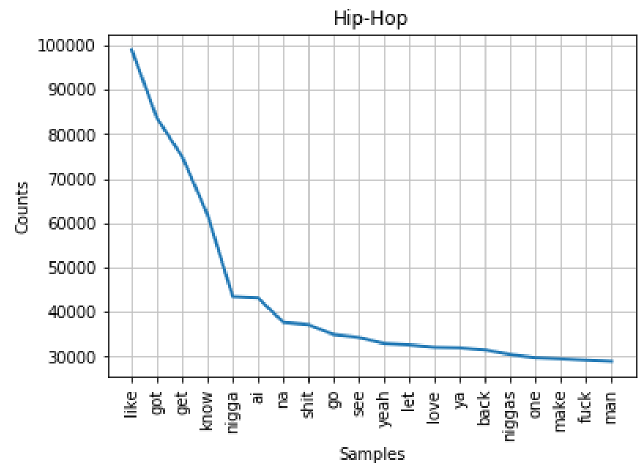

Figure 3: Most frequently used words in Hippop.

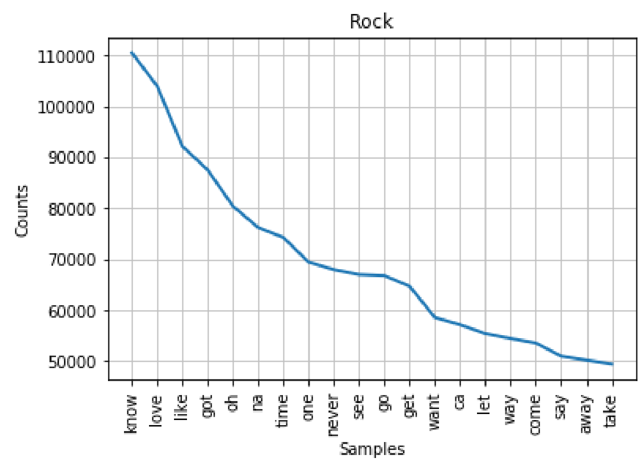

Figure 4: Most frequently used words in Rock.

1-hot encoding w_t of a token is input at each teim step t conditioned on a recurrent hidden layer h_t and outputs the probability distribution of the next token.

4.1 Semantic Controlled LSTM cell

Long Short-term Memory is a recurrent neural network architecture which uses a vector of memory cell $c_t \in \mathrm{R}^n$, and a set of element-wise multiplication gates to control how information is stored, forgotten, and exploited inside the network. Of the various different connectivity designs for an LSTM cell(Graves, 2013; Zaremba et al., 2014), the architecture used in this paper is illustrated and defined by the following equations,

$$i_t = \sigma(W_{\omega i} w_t + W_{hi} h_{t-1}) \tag{1}$$

$$f_t = \sigma(W_{\omega f} w_t + W_{hf} h_{t-1}) \tag{2}$$

$$o_t = \sigma(W_{\omega o} w_t + W_{ho} h_{t-1}) \tag{3}$$

$$\hat{c}_t = \tanh(W_{\omega c} w_t + W_{hc} h_{t-1}) \tag{4}$$

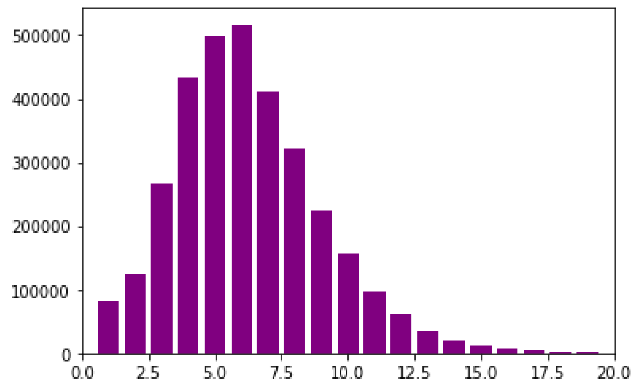

Figure 5: Sentence length of Rock

Figure 6: Most common words for AC DC

Figure 7: Most common words for Fall Out Boy

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t) \quad (5)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (6)$$

where σ is the sigmoid function, i_t, f_t, $o_t \in [0,1]^n$ are input, forget, and output gates respectively, and $\hat{c}_t$ and c_t are proposed cell value and true cell value at time t.

4.2 The Deep Structure

The neural language generator proposed in this project is extended to be deep by stacking multiple LSTM cells on top of the original structure.

4.3 GANs

In a generative adversarial networks(GANs) framework, there are two adversaries, a discriminator and a generator. The generator forges samples that are intended to be from the same distribution as the training data. The discriminator is a binary classifier striving to distinguish the generated fake samples from the real ones. In this project, different recurrent architectures of the generator and discriminator can be investigated. Considered GANs have been successfully applied in computer vision to generate high-resolution and realistic images, but not so successful in text generation models, this approach will be an attempt.

4.4 Training

The forward generator is trained by treating a group of sentence(verses or choruses) as a mini-batch. The objective function is the cross entropy error between the predicted word distribution p_t and the actual word distribution y_t in the training corpus. l2 regularisation will be added to the objective function to avoid over fitting. Besides, batch normalization, dropout and other regularization will be implemented accordingly.

5 EVALUATION OF GENERATED LYRICS

5.1 Bilingual Evaluation Understudy Score

The Bilingual Evaluation Understudy Score(BLEU), is a metric for evaluating a generated sentence to a reference sentence. Although developed for translation, it can be used to evaluate text generated by a language model. I plan to use the implementation of BLEU score provided by NLTK to evaluate the model.

REFERENCES

[1] malmi2016dopelearningMalmi, E., Takala, P., Toivonen, H., Raiko, T. Gionis, A. 2016. Dopelearning: A computational approach to rap lyrics generation Dopelearning: A computational approach to rap lyrics generation. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (195–204).

[2] mikolov2010recurrentMikolov, T., Karafiát, M., Burget, L., Černockỳ, J. Khudanpur, S. 2010. Recurrent neural network based language model Recurrent neural network based language model. Eleventh annual conference of the international speech communication association. Eleventh annual conference of the international speech communication association.

[3] mikolov2012contextMikolov, T. Zweig, G. 2012. Context dependent recurrent neural network language model Context dependent recurrent neural network language model. 2012 IEEE Spoken Language Technology Workshop (SLT) 2012 ieee spoken language technology workshop (slt) (234–239).

[4] zhang2014chineseZhang, X. Lapata, M. 2014. Chinese poetry generation with recurrent neural networks Chinese poetry generation with recurrent neural networks. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp) (670–680).