Intelligent Lyric Generator of Different Genres

Yiding Wen
University of Virginia
Charlottesville, VA
yw2eq@virginia.edu

Jiefu Liang
University of Virginia
Charlottesville, VA
jl2dz@virginia.edu

Jiabao Zeng
University of Virginia
Charlottesville, VA
jz2av@virginia.edu

ABSTRACT

Text generation is an important and widely used technique in the text mining field. But when applied to generating lyrics, there are special concerns such as rhymes, maintaining the lyrical and topic consistency and differences among all kinds of genres. Therefore, in our project, we propose an intelligent lyric generator to apply this technique specifically to the lyric field, where the user inputs a sentence and our system generates a series of lyric sentences. The framework mainly contains two parts: keyword extraction and the seq2seq model. The former one ensures that the generated lyrics are consistent with the user's intent. And the seq2seq model automatically generates lyrics based on the extracted keywords. In order to make it more fun, we propose two lyric generators for pop and rock genres.

KEYWORDS

lyric generation, keyword extraction, seq2seq model, GRU, LSTM

ACM Reference Format:
Yiding Wen, Jiefu Liang, and Jiabao Zeng. 2018. Intelligent Lyric Generator of Different Genres. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Text generation has been applied to a variety of practical fields, such as auto-replying emails or auto-correcting misspelled and grammatical mistakes. Since texts are constructed by a series of words, Recurrent Neural Networks (RNNs), which is a family of neural networks designed specifically to catch the sequential characteristics of data, is widely used in text generation. Normally, the text generation model is given several leading words as the input and then generates a bunch of texts which are grammatically and semantically correct.

However, in the case of lyric generation, several problems should be addressed. First, when song writers write lyrics, they have a certain topic in mind or lyrics are created around several keywords, which is different from generating texts from several leading words. Then, lyrics need to fulfill a set of rhetoric and rhythmic rules in order to maintain a certain lyrical consistency as well as following the melody and beats. At last, even given the same keywords, different genres of music will have have different lyrics.

In order to solve these problems, we propose an intelligent lyric generator to apply text generation specifically to the lyrics field. Our work is mainly divided into two parts: extracting keywords and generating lyrics. Keywords are extracted from our training text set and lyrics are generated from keywords. To gain better text generation performance, we try GRU (Gated recurrent units) and Long Short Term Memory networks (LSTMs), special kinds of RNNs capable of learning long-term dependencies, to train the great amount of lyric data. After the experiment, we found that GRU outperforms LSTM in our project. Therefore, we choose to use GRU in our lyric generation part. Also, there is a potential improvement of our proposal. Since lyrics are highly related to song writers' sentiment and the social media is the best approach to express personal feelings, we can make full use of this in a way that keywords are generated from users' social media and lyrics are created according to their daily post.

2 RELATED WORK

Lyrics generation is a challenging task in NLP. To some extent, this task is similar to poem generation, which is also challenging. The basic idea of generation text is similar in this two applications, but lyrics have a stronger connection with melody and rhythm. At the same time, lyrics have a quite significant different words preference among different genres.

Lots of work have been done to explore this field from different aspects. In this paper[6], the researchers proposed a scheme for content based keyword generation of song lyrics. The researchers also implemented N-gram model to generate lyrics for different genre[1]. Jones, M. deployed a model based on Recurrent Neural Network to generate rap lyrics. Some work focused on prediction. A prediction model[3] was developed to identify the next line of existing lyrics from a set of candidate next lines. This model based on two machine-learning techniques: the RankSVM algorithm and a deep neural network model with a novel structure.

Some work about about poem generation is also can be used and applied into lyrics generation. The researchers identified the required syllable pattern and tried to automatically generate lyrics for a given melody[3]. Greene et al. applied statistical methods to analyze, generate and translate rhythmic poetry[2]. Researchers showed that models for generation can combine word and character level information to significantly outperform word level model or char level model[7]. Another breakthrough is to use language model and recurrent neural network to adjust the poem generate model to various kinds of inputs[5].

Our proposal is different from the previous work as follows. First, our work generates lyrics based on contents from users' social

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

network, which is highly related to each users' emotion and life. Second, we extract keywords from lyrics and train our model with keywords. The previous lyric generators accept existing lyrics or words need as input, but our model can generate lyrics based on input keywords, which is a more natural way to write lyrics. When considering the rhythm feature of lyrics, we also can add melody data into our RNN model easily to generate lyrics that can match to a certain melody. This makes it possible to not only generate lyrics but also generate a whole song in the future.

3 DATASET

The dataset we use is from Kaggle [1] (a platform for predictive modelling and analytics competitions), which contains total 38,000,000 lyrics from different artists and various music genres arranged by year. This dataset has already annotated the lyrics with its corresponding music genres. In this dataset, every row contains the five attributes: Artist, Song name, Year, Genre, Lyrics. The number of documents for several music genres is more than 40,000, like pop music and country music. In our project, we choose pop and rock as our target music genres, which respectively contains 89280 and 167894 songs, in order to guarantee the training dataset is large enough.

4 APPROACH

4.1 Keyword extraction

There are various keyword extraction algorithms, such as TF-IDF, TextRank and RAKE (Rapid Automatic Keyword Extraction). In our project, we choose the RAKE algorithm. The reason is that the TF-IDF algorithm is frequently utilized in MPs and we'd like to learn new approaches after the project. Another consideration is that RAKE algorithm is easy to understand and implement using Python and NLTK library. Basically, the RAKE implementation contains the following steps: 1) remove all stopwords from the text data; 2) create an array of candidate keywords which are set of words separated by stopwords; 3) find the frequency of the words; 4) find the degree of the each word. (Degree of a word is the number of how many times a word is used by other candidate keywords); 5) for every candidate keyword, find the total frequency and degree by summing all word's scores; 6) finally degree/frequency gives the score for being keyword.

Keyword extraction is mainly used in two places in our project, the training stage and the generating stage. In the training stage, we will extract one keyword from each sentence of the lyrics, which will be added to the original dataset as our training data. In the generating stage, we will extract any number of keywords from the user input. Note that in the generating stage, the user could define the number of keywords, which result in the number of lyric sentences.

4.2 The lyrics generator

4.2.1 Seq2Seq. In our project, we used a similar model as the Chinese poetry generator[5]. The Seq2Seq model can be used for many text automatic generators, like machine translation and text style extraction. In our model, we will generate the lyric line by line. In order to generate the first sentence lyric, we only use the first keyword to generate the first sentence lyric because here we do not have input sentence for encoder. For all the other non-first sentence lyrics, we will rely on the keyword for this sentence lyric and the previous sentence lyric to generate them. Figure 1 shows the big picture of encoder-decode model.

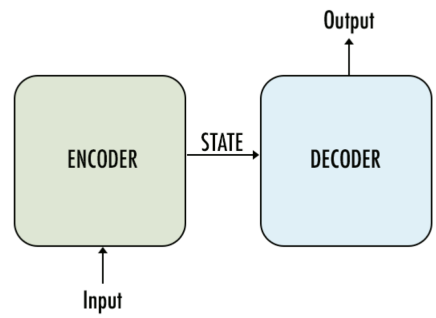

Figure 1: The basic structure for Seq2Seq model

4.2.2 Basic Structure. Firstly we will do a word embedding to convert our input sentence to a list of index and put the list into the encoder model. In the structure, the encoder model will memorize the important features of our input sentence and generate the encoder hidden state, which is also called context vector. Then the decoder model will rely on the context vector and its own output as the next input to generate the lyric word by word. Figure 5 shows the detail for encoder-decoder structure.

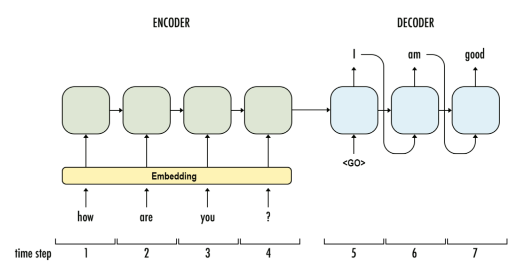

Figure 2: Concrete structure for Seq2Seq model

5 DATA PREPOSSESSING

Based on the previous training rules, we processed our training data and split it into a bunch of sentence pairs. For every sentence pair in our training data, the first item concatenate the keyword and the previous sentence lyric. The second item is the ground truth for the training period. Here we take the pop songs as example. After processing the data in this way, we totally read the around 660,000 sentence pairs and around 90,000 tokens in the corpus for pop songs.

[1]https://www.kaggle.com/karineh/40-year-lyrics-evolution/data

```
Read 661730 sentence pairs
Counting words...
Counted words:
lyric 89280
['shadow And the circle wont be broken', 'When you sail into the shadow of the storm']
['pleas Wont you please forgive', 'Please']
['night I said its alright yeah', 'We got all night whoa']
['utgemandeulgo DO and Baekhyun', 'Aichorom nol jaju utgemandeulgo']
['heavi Trying to smile but', 'The air is so heavy and dry']
['enough Yeah I dont know I dont know I dont know why', 'Cant get enough of your love babe']
['workmat I shall go to his office', 'Mix with his workmates']
['beauti Drop dead', 'Beautiful']
['bring And my falling tears are mingled with the wine', 'I would bring you happiness']
['nearli She looks fine', 'And she nearly blew my mind']
['break Youre breaking my heart again', 'Youre breaking my heart again']
['goodby When We Touch', 'I Hate Goodbyes']
['hater Now Im no more tied down by your laugh out to drown', 'Youre a hater such a hater']
['matter Lookin over n over again', 'Talkin no matter how late n laffin away and away n']
['world', 'When your world is cold and lonely']
```

Figure 3: Training sentence pairs for pop song lyric

Here in figure 3 we randomly choose several sentence pairs as the examples. Note that in all the sample sentence pairs, there are not the first sentence lyric. If the item is for the first sentence lyric, the first item for the pair should only be a single keyword.

6 TRAINING

Based on the results from the previous two section, we used the training sentence pairs to train the basic Seq2Seq model. For the encoder and decoder model selection, we tried GRU and LSTM as our encoder model and decoder model. Compared the generated lyric from the two models, we found in this task, GRU outperformed LSTM in the perspective of the meaning for lyric. In the following next sections we will only focus on the performance of GRU.
For the lyric for pop songs, we totally read around 660,000 sentence pairs and around 90,000 tokens in corpus. In the training period, we chose the first 100,000 sentence pairs and trained the model for at most 5 epochs until we found overfitted problem. In the next section we will show our training results for the model and use the term average loss in the figure. We defined the average loss as the number of wrong prediction words in every sentence in average. For instance, if the average loss equals to 5, it means that in every sentence 5 words will be wrong prediction in average.

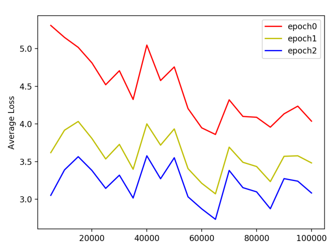

Figure 4: Loss of different epoch in training pop lyrics model

For the two figures 4 and 5, we showed the training results for the first three epochs. In the two figures, we calculated the average loss every 5000 pairs and 1000 pairs, respectively. In the horizontal view, every line in the two figures shows the trend of loss rate in one epoch. In the vertical view, we could see for the first 5000 pairs or 1000 pairs, the loss rate decreases as the number of epochs increases. Therefore, the overall trend for this training period is closer to convergence after every epoch.

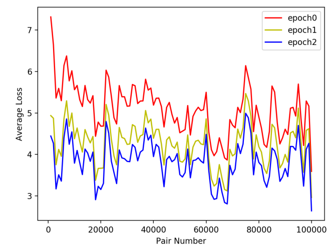

Figure 5: Loss of different epoch in training pop lyrics model

7 RESULT

7.1 Sample generated lyrics

Based on user's input sentence, our model can generate up to four lines of lyrics. Below are some examples. We can see that the generators can give appropriate and reasonable lyrics. At the same time, the lyrics generated by rock and pop are quite different.

Input:

I cant wait to meet you there, Saying goodbye and I dont even care

Lyrics generated by pop lyrics generator:

When we say goodbye,we say goodbye/ I dont know what i mean to be wrong/ but when I walk away/ I can be all the same and

Lyrics generated by rock lyrics generator:

Say goodbye/ You left a little bit/ I'm on the top of the angels/ I'm going all the way that I've been

Input:

I received a birthday gift from my best friends. I had a crush

Lyrics generated by pop lyrics generator:

but your birthday to your birthday/ and i was just friends and friends friends are friends/ but its just to stop/ you let me see the world

Lyrics generated by rock lyrics generator:

Im heading for the light/ You left a friend/ You left a way of the way/ You left a hole where my name

Input:

We had a fight last night, I am so upset

Lyrics generated by pop lyrics generator:

Into a night I can't stand/ Don't fight you dont know what you want to fight/ I just can't get myself to be for sure/ Lets give all your time

Lyrics generated by rock lyrics generator:

Im going all the night/ You gotta fight just to make it through/ You won't be the way of the sun/ You will take the blame

Input:

Today is a sunny day and I am really happy to see my friends after final exams

Lyrics generated by pop lyrics generator:

but i really want to be all you've got to/ and i want to be the friend of the friends/ but, finally finally come around/ you want to be the only one

Lyrics generated by rock lyrics generator:

You really are/ You left a friend/ You left to finally let me/ You gotta get away and I feel all

7.2 User study

Since this is text generation task, there is no "correct answer" of the result. Thus, we can not use metrics like precision, recall or f1 to evaluate our work. It is well known that accurate evaluation of text generation system is difficult, such as the poetry generation and dialog response generation[8][4]. For given input keywords, there are nearly infinite ways to generate appropriate lyrics.

Thus, we design a user study to evaluate the quality of the lyrics our model generated. We selected 4 lyrics randomly from our dataset, Sample 1, 2, 3 and 4. Sample1 and Sample2 are from pop songs. Sample3 and Sample4 are from rock songs. We use the first line of Sample1 as input of our pop lyrics generator, then got the Result1. We also use the first line of Sample3 as input of our rock lyrics generator to got the Result3. We did the same operation on Sample2 and Sample4 correspondingly to got the Result2 and Result4.

After deleted the first line of Sample 1, 2, 3 and 4, we have GroundTruth 1, 2, 3 and 4 correspondingly. Then we randomly ordered GroundTruth 1, 2, 3 ,4 and Result 1, 2, 3, 4.

For each lyrics, participants were asked to choose an option of two questions:

- Who write these lyrics?
  Human or Machine.
- Which genre does these lyrics belong to?
  Pop or Rock.

In total, we have 15 participants for the turing test. 2 out of 4 machine generated lyrics are most likely to be evaluated as written by human beings, which proves that the quality of our project is close to the human lyrics. But as for the genre, participants cannot correctly tell the differences, even for the human lyrics.

GroundTruth1	You can leave with,me or you could have the blues Some call it arrogant, I call it,confident You decide when you find on what I'm,working with Damn I know I'm killing you with them,legs
Result1	You could feel the end You will be the end of me You left a little more
GroundTruth2	Alone in my place, my heart is away All that I can think of is, we should get married We should get married Let's stop holding back on this and let's get carried away
Result2	You left me under the sun You got a story You got a simple You gotta be the sun
GroundTruth3	Gonna leave behind things that won't decompose I'll just call this what it is My vanity gone wild with my crisis One day this all will repeat Now sure hope they make something useful, out of me
Result3	I'm heading for a breakdown I'm not one step behind I'm all right all these things You left a little more
GroundTruth4	No matter how tight the reigns love will find its own A time to reap, a time to sow And many a time to cry in vain But now the time to celebrate The glory of this imperfection
Result4	You left me to make it through You gotta be the right to the direction You left to make it through You take the way of the way of the way

Table 1: Lyrics used in user study

8 CONCLUSION

In our project, we propose an intelligent lyric generator for rock and pop genres. The framework contains two main parts: keyword extraction and the seq2seq model. Keyword extraction captures the user's intent and maintains the lyrical and topic consistency. The seq2seq model realizes generating a series of lyric sentences. From the result of our user study, we can conclude that the quality of our generator is close to the human lyrics where 2 out of our generated lyrics are evaluated as written by humans. However as for the genre differentiation, participants in general cannot tell which one is from the pop or rock genre.

9 FUTURE WORK

In the future, we would like to improve our project deeper and broader in two ways. The first one is that besides existing lyrics and extracted keywords, we can add melody features when training the seq2seq model, which will help our generator better capture the rhymes and beats. The second is to actually relate our generator with people's daily life, such as one song per day based on user's tweet data.

REFERENCES

[1] Sofie Christensen. 2016. *Applying text mining strategies to song lyrics.* Master's thesis. School of Computer Science and Informatics.

[2] Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 524–533.

[3] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 195–204.

[4] Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *6th SIGdial Workshop on DISCOURSE and DIALOGUE.*

[5] Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. *arXiv preprint arXiv:1610.09889* (2016).

[6] Bin Wei, U Rochester, Chengliang Zhang, and Mitsunori Ogihara. 2007. Keyword generation for lyrics. *marriage* 195, 47 (2007), 64.

[7] Stanley Xie, Ruchir Rastogi, and Max Chang. [n. d.]. Deep Poetry: Word-Level and Character-Level Language Models for Shakespearean Sonnet Generation. ([n. d.]).

[8] Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 670–680.