Song lyrics generation within a specified genre

Melihedov Artem

May 2023

Abstract

From the very beginning of work in the field of text generation, one of the directions was the direction associated with the creation of works of art: poems, stories, songs. After the developments in the field of text generation for musical works achieved some success, researchers began to complicate the generation process with additional conditions, such as text subject matter, rhythmic pattern, generation based on a musical fragment, etc. This work explores the possibility of generating lyrics within a given genre. During the implementation of the project, 2 generative models (LSTM as a baseline and GPT-2) and a classifying model (BERT) were trained (finetuned) to assess belonging to the genre. Link to the project code repository: `https://github.com/art1yome/npl-fp-2023`.

1 Introduction

Lyrics text generation can be used in a variety of ways. The generated lyrics can be used as standalone pieces or be used as inspiration for artists and songwriters.

Simple text generation does not take into account the peculiarities of lyrics, vocabulary and structure used in individual genres. In this work, the possibility of generating song lyrics with belonging to a particular genre is explored.

1.1 Team

The project was done alone by **Melihedov Artem**.

2 Related Work

Using a computer to create literary texts is a very old idea, it was described in the middle of the last century [Wheatley, 1965]. However, due to the obvious technical limitations of the time, the really fruitful advances came along with the rapid development of language models in the last decade and the increased interest in this topic.

The improvement of lyrics generation models comes with the exploration of different approaches. One of these is conditional text generation on audio

clips [Vechtomova et al., 2020]. The authors of the article propose to use CNN Encoder to encode the spectrogram of an audio file into latent space and use it to generate text using LSTM. A similar approach is discussed in another article [Tian et al., 2023] using the GPT-2 model. Fig. 1

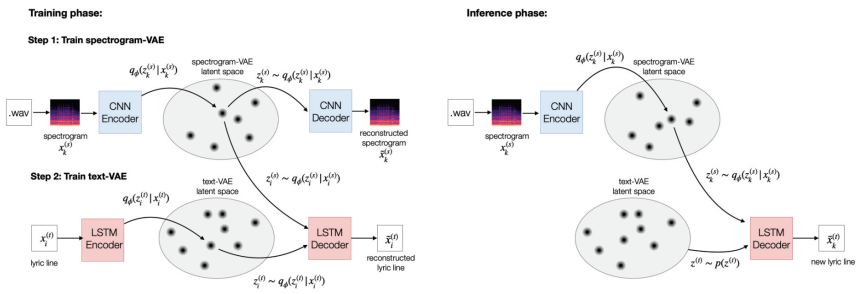

Figure 1: Music audio clip conditioned lyrics generation.

A more advanced approach is proposed by the authors of the article [Tikhonov and Yamshchikov, 2018]. They solve the problem of generating rap, so modeling rhyme and rhythm is important to them. To do this, they produce Beat Detection, Lyric and Beat Alignment. They get the timestamp of each word and each bit and align them according to their timestamps. But since words and bits do not always match, the authors of the article minimize the distance between the timestamp of a word and the timestamp of a bit.

An alternative approach to the generation of musical texts is considered in the article [Zhang et al., 2022]. The authors believe that it is not worth completely eliminating a person from the creative process and are considering an AI assistant who can generate lyrics and improve it based on feedback from a person. To do this, they use GPT-2 and context. Fig. 2

The generation of stylized poems is studied in the article [Tikhonov and Yamshchikov, 2018]. The authors use the LSTM model and dataset from English and Russian poetry to generate poems in the style of one of the writers. As a result, they manage to generate texts that are twice as likely to refer to a given author than simple generation. Fig. 3

3 Model Description

3.1 LSTM

The compact forms of the equations for the forward pass of an LSTM cell with a forget gate are:

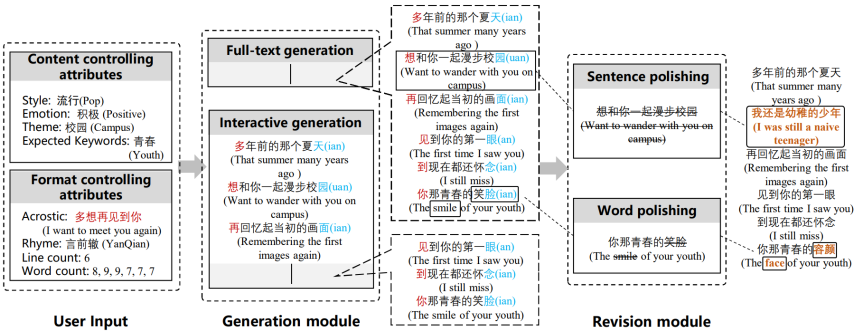

Figure 2: The system supports multifaceted controlling attributes in user input to control the content and format of lyrics. The generation module provides two modes for draft lyrics creation: full-text generation and interactive generation. The former generates a full lyrics while the latter generates following sentences conditioned on the preceding context. Besides, a revision module is introduced to polish undesirable sentences or words.

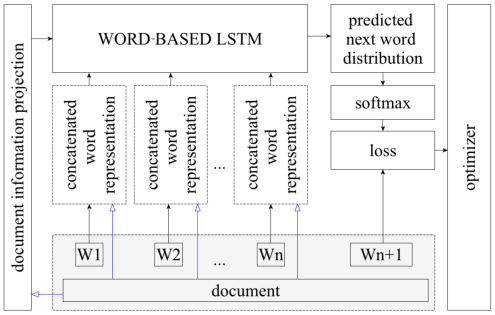

Figure 3: The scheme of the language model used. Document information projections are highlighted with blue and white arrows. The projections on a state space of the corresponding dimension is achieved with simple matrix multiplication of document embeddings.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$
$$h_t = o_t \circ \sigma_h(c_t) \tag{5}$$

where the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\odot$ denotes the Hadamard product (element-wise product). The subscript t indexes the

time step.

Variables:

- $x_t \in \mathbb{R}^d$: input vector to the LSTM unit
- $f_t \in (0,1)^h$: forget gate's activation vector
- $i_t \in (0,1)^h$: input/update gate's activation vector
- $o_t \in (0,1)^h$: output gate's activation vector
- $h_t \in (-1,1)^h$: hidden state vector also known as output vector of the LSTM unit
- $\tilde{c}_t \in (-1,1)^h$: cell input activation vector
- $c_t \in \mathbb{R}^h$: cell state vector
- $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h}, b \in \mathbb{R}^h$: weight matrices and bias vector parameters which need to be learned during training

3.2 GPT-2

GPT-2 has a generative pre-trained transformer architecture which implements a deep neural network, specifically a transformer model, which uses attention in place of previous recurrence- and convolution-based architectures. Attention mechanisms allow the model to selectively focus on segments of input text it predicts to be the most relevant. This model allows for greatly increased parallelization, and outperforms previous benchmarks for RNN/CNN/LSTM-based models. Fig. 4

3.3 BERT

BERT is based on the transformer architecture. Specifically, BERT is composed of Transformer encoder layers.

BERT uses WordPiece to convert each English word into an integer code. Its vocabulary has size 30,000. Any token not appearing in its vocabulary is replaced by [UNK] for "unknown".

BERT was pre-trained simultaneously on two tasks:

language modeling: 15 of tokens were selected for prediction, and the training objective was to predict the selected token given its context.

After processing the input text, the model's 4-th output vector is passed to a separate neural network, which outputs a probability distribution over its 30,000-large vocabulary.

next sentence prediction: Given two spans of text, the model predicts if these two spans appeared sequentially in the training corpus, outputting either [IsNext] or [NotNext]. The first span starts with a special token [CLS] (for "classify"). The two spans are separated by a special token [SEP] (for "separate").

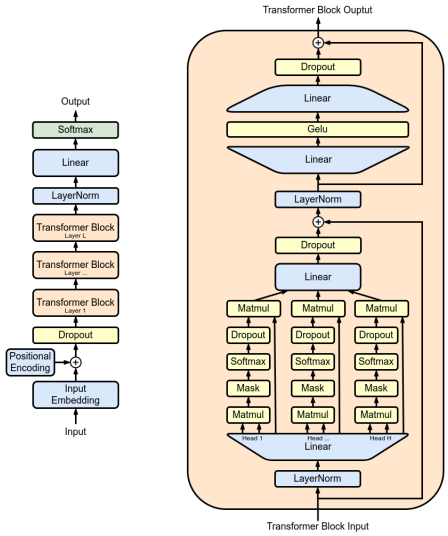

Figure 4: GPT architecture.

After processing the two spans, the 1-st output vector (the vector coding for [CLS]) is passed to a separate neural network for the binary classification into [IsNext] and [NotNext].

As a result of this training process, BERT learns latent representations of words and sentences in context. After pre-training, BERT can be fine-tuned with fewer resources on smaller datasets to optimize its performance on specific tasks such as NLP tasks (language inference, text classification) and sequence-to-sequence based language generation tasks (question-answering, conversational response generation). The pre-training stage is significantly more computationally expensive than fine-tuning. Fig. 5

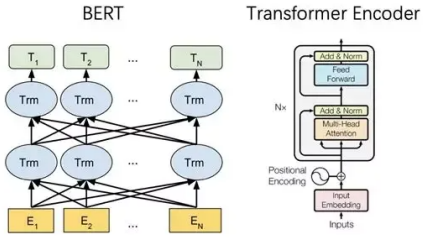

Figure 5: BERT.

4 Dataset

To build dataset scraped lyrics from Vagalume was used [Neisse, 2021].

	Train	Valid	Test
Songs	5000	1000	1000
Tokens	2,088,628	217,646	245,569
Total songs		371182	
Artists		4168	

Table 1: Statistics of the dataset. No full dataset (371182) was used because of technical limitations.

This dataset was made by scrapping data from Vagalume website. It contains 371182 song lyrics from 4168 artist in 79 different musical genres and in different languages. For this work we used only english songs with 5 most popular genres: Pop, Rock, Hip Hop, R&B, Rap.

We also removed song that was to short or to long.

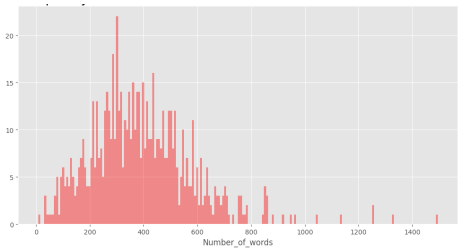

Figure 6: Song length.

5 Experiments

5.1 Metrics

In this project BLUE-metric was used to evaluete the quality of text generation.

BLEU's output is always a number between 0 and 1. This value indicates how similar the candidate text is to the reference texts, with values closer to 1 representing more similar texts. Few human translations will attain a score of 1, since this would indicate that the candidate is identical to one of the reference translations. For this reason, it is not necessary to attain a score of 1. Because there are more opportunities to match, adding additional reference translations will increase the BLEU score.

$$BLEU_w(\hat{S}; S) := BP(\hat{S}; S) \cdot \exp\left(\sum_{n=1}^{\infty} w_n \ln p_n(\hat{S}; S)\right)$$

Other metric that was used is GD (Genre Distance) - distance between specified genre and the one predicted by classification model.

$$GD(\hat{y}; y) := \sqrt{(\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \cdots + (\hat{y}_n - y_n)^2}$$

5.2 Experiment Setup

5.2.1 Generation model LSTM

LSTM model was train to predict next word.

LSTM generator consists of:

1. Biderectional LSTM (Long Short-Term Memory) size of 250 with dropout coefficient 0.1.
2. Linear layer with dropout coefficient 0.1.

Params:

1. Loss: $categorical_crossentropy$.
2. Optimizer: $adam$.
3. Epoch number: 10.
4. Batch size: 64.
5. Max length 512.

5.2.2 Generation model GPT-2

Pretrained GPT-2 model was used with finetuning.

Params:

1. Loss: $categorical_crossentropy$.
2. Optimizer: $adamw$.
3. Sheduler: $get_linear_schedule_with_warmup$.
4. Epoch number: 10.
5. Batch size: 16.
6. Max length 400.

5.2.3 Classification model BERT

Pretrained Bert model was used with finetuning.

LSTM generator consists of:

1. BERT model.
2. Dropout with coefficient 0.3.
3. Linear layer.

Params:

1. Loss: $BCEWithLogitsLoss$.
2. Optimizer: $Adam$.
3. Epoch number: 10.
4. Batch size: 32.
5. Max length 16.

6 Results

Classification model metrics presented in Fig. 7 and Fig. 8.

	precision	recall	f1-score
0	0.79	0.81	0.80
1	0.88	0.78	0.83
2	0.80	0.68	0.74
3	0.69	0.69	0.69
4	0.60	0.19	0.29
micro avg	0.79	0.73	0.76
macro avg	0.76	0.58	0.63
weighted avg	0.79	0.73	0.75
samples avg	0.79	0.77	0.76

Figure 7: Classification metrics.

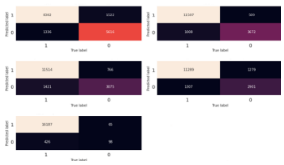

Figure 8: Classification confusion matrix.

Examples of generated texts presented in Tab. 2 for LSTM model and in Tab. 3 for GPT-2 model.

I am missing you with me you shame in you us it i time if it i for still i different been i different i what girl the come i what now girl the can it i of let's we cause so the yeah the have when hard so and don't i for life i could don't i turn girl things will i pull back you day i let's time mine don't i dirt can't me make go i of stay we're i let's

Table 2: LSTM sample.

I can remember the good old days When you and me used to hide away Where the stars were shining Or the sun was blinding our eyes Yeah you filled up my glass With promises that could never last And I still find pieces of you in the back of my mind And all of the things that we once said They're not in my heart

Table 3: GPT-2 sample.

	BLUE	GenreDistance
LSTM	0.07	0.64
GPT-2	0.45	0.51

Table 4: Generation models metrics.

7 Conclusion

This project proposes a solution to the problem of generating song lyrics within a specified genre. Train data was taken from scraping Vagalume. Tha problem was solved using two natural language processing approaches: LSTM-model and GPT-2 model. A comparative analysis of the results of this approaches showed that the second approach - GPT-2 model is superior in both metrics.

References

[Neisse, 2021] Neisse, A. (2021). Scraping lyrics from vagalume. In *https://aneisse.com/post/20190210-music-data-scraping/20190210-music-data-scraping*.

[Tian et al., 2023] Tian, Y., Narayan-Chen, A., Oraby, S., Cervone, A., Sigurdsson, G., Tao, C., Zhao, W., Chung, T., Huang, J., and Peng, N. (2023). Unsupervised melody-guided lyrics generation.

[Tikhonov and Yamshchikov, 2018] Tikhonov, A. and Yamshchikov, I. P. (2018). Guess who? multilingual approach for the automated generation of author-stylized poetry. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

[Vechtomova et al., 2020] Vechtomova, O., Sahu, G., and Kumar, D. (2020). Generation of lyrics lines conditioned on music audio clips.

[Wheatley, 1965] Wheatley, J. (1965). The computer as poet. In *Journal of Mathematics and the Arts, vol. 72, no. 1, pp. 105.*

[Zhang et al., 2022] Zhang, R., Mao, X., Li, L., Jiang, L., Chen, L., Hu, Z., Xi, Y., Fan, C., and Huang, M. (2022). Youling: an ai-assisted lyrics creation system.