Lyrics & Tune Style Transfer

Qixuan Xiao, Haozhe Liu, Honghu Luo, Ming Wang, Yanhao Shen
Viterbi School of Engineering, University of Southern California
{qixuanxi, haozheli, honghulu, mwang283, yanhaosh}@usc.edu

1 Introduction

While the interest in style transfer is getting popular in recent years, many NLP-related research and applications show the success of transforming formal English to informal (Jin et al., 2020). It inspired us to apply for style transfer in music. In this project, we are going to select two artists, each from rang song and pop music areas and transfer one artist's style to another's.

Text style transfer has grown very fast during the past few years, and lyrics style transfer is a subset of it. The major challenge of lyrics style transfer is the lack of parallel corpora. Therefore, we applied unsupervised text style transfer to change the style of lyrics, since golden labels aren't common to find. We adopt the sequence-to-sequence approach and use the Encoder-Decoder architecture to achieve style transfer. The baseline model is adapted from (Santos et al., 2018), it combines an autoencoder with a style classifier.

More recently, especially in the context of the language, the term 'style transfer' came to be used more generally to refer to any form of style transformation. An extension of this concept to music offers composers and music lyricists a creative tool for repurposing an existing material in an innovative way. Our work represents one of the first few attempts of tune style transfer for symbolic music. We proposed a supervised end-to-end learning model, which produced any given combination of tracks (Cífka et al., 2020).

In this present world, we adhere to the traditional definition of style transfer and apply it within the domain of symbolic music. Our project consists of two parts, lyrics style transfer and tune style transfer. And our task can be more precisely formulated as follows: transfer the content of a song X in the style of a reference song Y and obtain a synthetic song Z, which keeps content from X and style from Y.

2 Related Work

Interest in style transfer has rapidly grown in recent years, we found out that (Jin et al., 2020) summarizes text style transfer and many popular approaches. We explored that it uses an encoding decoding architecture to transform text styles. There are also some papers considering similar approaches but using different encoders(Hu et al., 2017), back-translation(Prabhumoye et al., 2018), and adversarial training(Romanov et al., 2018). We also read some papers that considered challenging tasks: detoxifying hate speech (Santos et al., 2018), changing political slant (Prabhumoye et al., 2018). In this project, we begin with an approach from (Santos et al., 2018) but a simplified version.

Music style transformations can take many different forms depending on the definition of style. Moreover, the present work is most closely to what we refer to as style translation, where a piece of music is converted to a given style. This work has a variety of subfield, instrumentation(Hung et al., 2019), general arrangement style(Brunner et al., 2018) and covering melodies(Nakamura et al., 2019). However, the above methods generally allow for a limited set of target styles or a single one.

Another type of tune style transfer is the harmonization of given melodies, what could be referred to as arrangement completion, where a new track is generated to complement a set of tracks given as inputs. Compared with the first type, this kind of transformation suffers fewer problems of imbalance music corpus. For example, the models proposed by (Hadjeres et al., 2017) enable harmonizing a given melody of Bach chorale.

3 Method

3.1 Lyric Style Transfer

For the Lyric Style transfer, we hope to develop a model that learns the structure in different music

genres and transfers original lyrics to a target style, while preserving the original lyrics' content. This is a style transfer task in NLP. However, the main challenge is that different music genres may not have too much overlap, they can be quite different in terms of structure, theme, and vocabulary. Therefore, this style transfer task is in the absence of parallel corpora. In this section, we will describe the model we use in detail and introduce the improvement we make to enhance the performance.

3.1.1 Design of the Model

For the baseline model, we adopt an Encoder-Decoeder approach to unsupervised text style transfer. Since it is hard to find matched pairs of original and target lyrics, we have to use the lyric line and its label as the input to do unsupervised text style transfer.

Given an input lyric line x_i^j, where j indicates the style of the lyric, we first need to extract its context feature which is separated from its style. Therefore, we adopted an autoencoder model. We feed the x_i and its label j to the encoder E, the context feature we hope to capture is z_i the output of the final hidden state of encoder E. The decoder D of the autoencoder maps z_i to the reconstruction $x_i^{j\to j}$ of the same shape as x_i. To make sure z_i can capture the content information, we train the autoencoder to minimise reconstruction errors $\mathcal{L}(x_i^{j\to j}, x_i)$.

After extract the context feature z_i, we need to use it to generate a lyric in the target style. We can use the same decoder which the autodecoder used to reconstruction. We feed the context feature z_i through the decoder D and generate lyric $x_i^{j\to 1-j}$ in the transferred style. Here, one problem is that we need to ensure the lyrics $x_i^{j\to j}$ and $x_i^{j\to 1-j}$ we generated are in the correct styles. Therefore, we need a classifier to predict the style of the lyrics, this binary classifier is trained along with the encoder and decoder. At each training epoch, we can use the original lyrics, the reconstructed lyrics, and the generated lyrics as the input data to train the classifier. The classification loss can be written as follow:

$$\begin{aligned}\mathcal{L}_{class}(x_i^j) =& \mathcal{L}_{class}(C(x_i^j), j) + \\ & \mathcal{L}_{class}(C(x_i^{j\to j}), j) + \\ & \mathcal{L}_{class}(C(x_i^{j\to 1-j}), 1-j)\end{aligned} \tag{1}$$

To train this two model simultaneously, we can combine the two loss mentioned before togather and obtain the final loss of out baseline model:

$$\mathcal{L}(x_i^j) = \mathcal{L}_{class}(x_i^j) + \mathcal{L}(x_i^{j\to j}, x_i) \tag{2}$$

The architecture of our basic model is shown in Figure 1.Next, we will briefly describe the encoder E, the decoder D and the classifier C we used in our model.

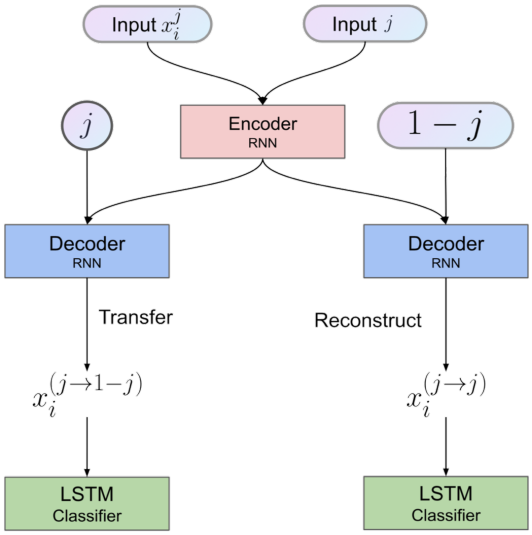

Figure 1: Architecture of model

Encoder For the encoder E, we use an RNN with GRU cells and add dropout for regularization. For lyrics with different length, we padded all them to the same length.

Decoder For the decoder D, we also use an RNN with GRU cells. Here, we have 2 decoder, one is for reconstruction, another is for text generation.

Classifier For the classifier C, we use the Bi-LSTM to predict whether lyric is the original or the target style.

3.1.2 Improvement on the Baseline Model

To further improve the performance of our model, we proposed one strategy. Since we use the encoder-decoder to do this sequence to sequence translation task, the qaulity of the generated text is depend on the context feature z_i. A potential issue here is that the encoder needs to compress all context feature of a source sentence into a fixed-length vector, which may make it difficult for the encoder to cope with long sentences, especially those that are longer than the sentences in the training corpus. This can hinder accurate reconstruction. To relieve this problem, we adopt the attention mechanism. We add an additive Attention on top of the decoder

D used in the basic model.(Bahdanau et al., 2014) By adding attention to the decoder, we can use the relevant information to select the appropriate output.

3.2 Tune Style Transfer

Style transfer is the process of changing the style of an image, video, audio clip or musical piece. For this project, our team would like to transfer the style of a song from a singer to another song with a different style and finally generate a new song. This task has great potential in practical applications within the music industry. We may have to overcome two main difficulties. First, the paired data is limited, so we have to find a efficient way to label the data. Second, music is hard to be understood by machines, so we need to tackle the problem of representing the music files in a trainable format. In this section, we will describe the solution architecture and the model in detail.

3.2.1 Design of the Architecture

Basically, directly training a model to generate a new tune for the source tune and target tune is hard, and we can see that it may lead to a result that we may not recognize the elements in both tunes. So the solution is that we only generate an accompaniment with new style and then apply this accompaniment to the original tune, in that way, we can preserve most of the original content.

Formally, our task can then be described as follows: Generate a new accompaniment for A in the style of B. Note that even though we expect the output to follow the same chord chart as A, we do not assume this chart to be available. Also, we assume B to be a song fragment approximately 8 measures long. While such a fragment might not fully capture the style of the entire song, it should manifest enough of its key features to allow for meaningful extrapolation. Basically, employing a previously heard accompaniment pattern in a creative, improvised way is a skill possessed by many human musicians, and one that we aim to mimic here. The graph shown below is the detailed architecture of our approach to this task.

So we start from the chord(source A and target B) and we create synthetic accompaniments in different styles (S and T), and then the input of the training model is content input A_S(accompaniment for A in style S) and style input B_T(a single track of B in style T), the output is $target\_T$(the corresponding track of the target

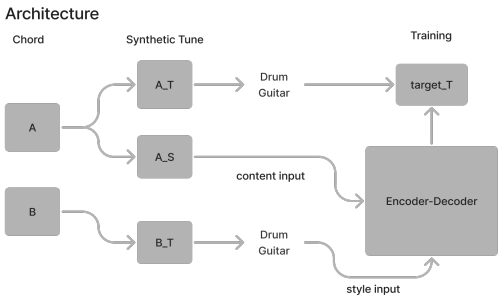

Figure 2: Architecture of workflow

accompaniment A_T(a single track of A in style T)), which will later be compared with the label for training.

3.2.2 Design of the Training Model

For the training model, we adopt a state-of-art Encoder-Decoder structure. To be specific, we use two different encoders to separately encode content input(source) and style input(target), and then we use some middle layers(Attention, Embedding) to better extract the features from the input, and in the end, we use a decoder to combine the representations computed by the two encoders to generate the corresponding output tune. The model structure is shown below.

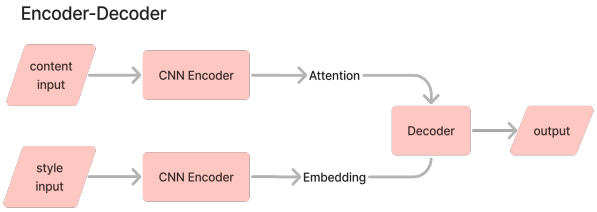

Figure 3: Architecture of training model

Encoder Apply CNN encoder to encode the content input and the style input respectively.

Mid-layer Use Attention for content input to calculate the attention weight, use embedding to represent different tracks in style input.

Decoder Combine the representations from these two encoders and use GRU to generate the corresponding output track.

4 Experiment

4.1 Experimental Setup

Datasets We obtained songs from Lyric.com[1] and cleaned the data as our dataset, which contains 3, 865 lines of lyrics from 86 songs of the original

[1]https://www.lyrics.com/

singer and 5, 540 lines from 77 songs of the target singer. And then, we tokenized each line with a word tokenizer, got rid of any lines fewer than 10 tokens or longer than 30 tokens, and transformed all words to lower case. The vocabulary extracted from this corpus yields 2, 226 unique tokens, including four special tokens: <pad>, <unk>, <s>, and </s>. In the experiment, our dataset is split into 70%, 10% and 20% for training, validation and testing separately.

Baseline Methods We used Encoder-Decoder (without attention) + BiLSTM as the baseline method.

Evaluation Protocols To measure the success of our lyric style transfer models, we evaluated the performance in three aspects: **fluency** (how natural the transferred lyrics sound), **content Preservation** (whether the output lyric remains in a similar meaning), and **style** (whether our pre-trained classifier can identify the output lyrics as the target style). In the fluency criteria, we used the pre-trained GPT-2 model to score the general perplexity (PPLg) for both the decoded sentences back to the original style and the target style(Lee, 2020), which shows how well the GPT-2 model can predict the lyrics generated by our encoder. As for the content preservation, we calculated the BLEU score(Liu et al., 2016) and generated contextualized word embeddings from BERT(Shimanaka et al., 2019) to quantify the semantic preservation extent. And for the style evaluation, we implemented another pre-trained BiLSTM classifier independent of the training one and computed the accuracy of correctly labelled lyrics.

4.2 Results and Discussion

Fluency We used GPT-2 model to compute the general perplexity score of the decoded lyrics. The PPLg score of the baseline is already high at 163.645, and our autoencoder with a attention layer achieves 166.976.

Content preservation BLEU and BERTscore shown in Table 1 were used to measure semantic preservation, and different approaches rank models differently.

Style Since some preserved contexts in the transferred lyrics are not present in the training set, the accuracy scores of the transferred lyrics are lower than the reconstructed sentences. Comparing classification accuracy shown in Table 2, the autoencoder model with an attention mechanism performs better.

Table 1: Evaluation of AutoEncoder

	Content		Fluency
Model	**BLEU**	**BERTscore**	PPL_{gpt}
Autoencoder	71.329	0.918	163.645
Autoencoder_Class	65.861	0.871	166.976

Table 2: Accuracy of AutoEncoder

	Style		
Model	**AUC real**	**AUC recons**	**AUC tsf**
Autoencoder	0.923	0.908	0.629
Autoencoder_Class	0.916	0.921	0.889

Lyric Transfer Results The aforementioned criteria objectively evaluate how our model's performance. However, it is better to ask humans to evaluate how well the style is transferred. Therefore, we give out some lyric transfer results in Table 3.

5 Conclusion and Future Work

Conclusion For the **lyric style transfer**, we use encoder-decoder approach to do unsupervised style transfer and add an attention layer on the decoder to improve the performance of the transformation. For the **tune style transfer**, we use bi-encoder to represent the tunes of source style and target style and apply a supervised learning approach to achieve a track-wise style transfer.

Future Work In the future, there are several aspects we intend to consider:

Transformer We hope to introduce the transformer architectures to further improve the performance. Compared with RNN, transformer do not process the data in order, therefore it can capture more diverse signals.

Human Evaluation We will also ask experts to evaluate the lyric transfer results, which is a better standard to evaluate the quality of transferred sentences.

Combination We plan to combine the lyrics transfer and the tune transfer to compose a complete song. Basically we have two approaches. One is to directly combine them, which requires handling the alignment problem. Another is to train the lyrics and tune simultaneously in one model.

Table 3: Results of Lyric transfer

Original	Transferred
I seen your cousin in the streets he sweet eying this booty.	I left your mark in the world whats this.
just the other day i had to shed a couple tears	i got put the greatest i had just a heavy.
and yes now im here without and see can i do	and now im blowing up and all you see is i

Division of labor

Lyrics Style Transfer

Qixuan Xiao : Implemented the basic model to achieve the lyric transfer results.

Haozhe Liu : Implemented the Web crawler to collect and clean the training data.

Honghu Luo : Implemented the evaluation methods to evaluate the model's performance.

Tune Style Transfer

Yanhao Shen : Combined accompaniments and source content, drew pitch graphs and built website for display.

Ming Wang : Trained the model to generate the transferred tune.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. 2018. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer.

Ondřej Cífka, Umut Şimşekli, and Gaël Richard. 2020. Groove2groove: One-shot music style transfer with supervision from synthetic data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2638–2650.

Gaëtan Hadjeres, François Pachet, and Frank Nielsen. 2017. Deepbach: a steerable model for bach chorales generation. In *International Conference on Machine Learning*, pages 1362–1371. PMLR.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Controllable text generation.

Yun-Ning Hung, I-Tung Chiang, Yi-An Chen, and Yi-Hsuan Yang. 2019. Musical composition style transfer via disentangled timbre representations.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2020. Deep learning for text style transfer: A survey.

Joosung Lee. 2020. Stable style transformer: Delete and generate approach with encoder-decoder for text style transfer. *arXiv preprint arXiv:2005.12086*.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Eita Nakamura, Kentaro Shibata, Ryo Nishikimi, and Kazuyoshi Yoshii. 2019. Unsupervised melody style conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 196–200. IEEE.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation.

Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2018. Adversarial decomposition of text representation.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. *arXiv preprint arXiv:1805.07685*.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine translation evaluation with bert regressor. *arXiv preprint arXiv:1907.12679*.