3-2022

Generating music with emotions

Chunhui BAO

Qianru SUN
Singapore Management University, qianrusun@smu.edu.sg

Generating Music with Emotions

Chunhui Bao and Qianru Sun

***Abstract*—We focus on the music generation conditional on human emotions, specifically the positive and negative emotions. There is no existing large-scale music datasets with the annotation of human emotion labels. It is thus not intuitive how to generate music conditioned on emotion labels. In this paper, we propose an annotation-free method to build a new dataset where each sample is a triplet of lyric, melody and emotion label (without requiring any labours). Specifically, we first train the automated emotion recognition model using the BERT (pre-trained on GoEmotions dataset) on Edmonds Dance dataset. We use it to automatically "label" the music with the emotion labels recognized from the lyrics. We then train the encoder-decoder based model to generate emotional music on that dataset, and call our overall method as Emotional Lyric and Melody Generator (ELMG). The framework of ELMG is consisted of three modules: 1) an encoder-decoder model trained end-to-end to generate lyric and melody; 2) a music emotion classifier trained on labeled data (our proposed dataset); and 3) a modified beam search algorithm that guides the music generation process by incorporating the music emotion classifier. We conduct objective and subjective evaluations on the generated music pieces, and our results show that ELMG is capable of generating tuneful lyric and melody with specified human emotions.**

***Index Terms*—Conditional Music Generation; Seq2Seq; Beam Search; Transformer**

I. Introduction

MUSIC is the art of sounds. In human life, the contemporary pop music is often used to express and share emotions. It is composed of lyrics, melody, accompaniment, chords, etc. Melody is a temporal sequence consisting of musical notes, and lyric is natural language representing music themes. Melody and lyric provide complementary information in understanding human emotions in songs. Conversely, it is interesting to see if the labels of human emotions can be used to composite tuneful music. Musicians composite music according to professional knowledge, such as harmonious relationships between pitch, duration, velocity, and tempo. We aim to achieve the composition automatically, e.g., by machine learning techniques simply with limited emotion labels.

In recent years, deep learning has made great progress for generating sequential data, such as natural language [1], audio [2], as well as music [3], [4]. Music generation aims to facilitate advanced automation in Smart City Life as well as Advanced Mechanical Engineering. Given the advent of large-scale music datasets, e.g., LMD-full MIDI Dataset [5] and reddit MIDI dataset [6], deep learning based generation models are now capable to "composite" high-quality music [7]–[9]. However, the mainstream is just focused generating music that can not be distinguished from those created by human composers [9]–[11]. It has not yet been widely studied if the generation can be conditional on human emotion labels.

As said by Carroll [12], music is the art of sounds in the topic of mood. While training a deep model with the sense of "mood" categories (i.e., human emotions) is challenging, due to the fact that most music datasets do not have emotion labels. Though in the literature there are small datasets, they are limited to learn the mapping (from emotions to music). In 2019, Ferreira *et al.* [13] built a music dataset VGMIDI composed of 95 labelled piano pieces and 728 unlabelled pieces, and trained a deep generative network to generate music with a given emotion. In 2020, they expanded this VGMIDI dataset from 95 to 200 labelled pieces and presented a model called Bardo Composer based on GPT-2 [14] to generate melodies with emotions specially for role-playing games. More recently, Hung *et al.* [15] proposed a symbolic music dataset EMOPIA that includes 1,078 music clips from 387 songs with Valence-Arousal emotion labels. However, as the generation model is notoriously hard to train and data-hungry, the number of labelled data in neither VGMIDI nor EMOPIA is great enough. Besides, these related works generate only melodies. We aim for the large-scale manual-labour-free dataset and the music generator for not only melodies but also lyrics with specific emotions.

To this end, we first build the lyric-melody dataset with emotion labels. We use the songs with English lyrics selected from the LMD-full MIDI dataset [5] and reddit MIDI dataset [6]. For labeling the clips with emotions labels, we do not use human labour but leverage a well-trained machine recognizer. The pipeline includes a few steps: 1) cutting each song into lyric segments with fixed length; 2) fine-tuning a Bert [16] model on Edmonds Dance dataset [17]; and 3) using the result model to annotate the segments. We elaborate the dataset construction in Section III. This pipeline has the advantage that it is automatic way to process a large amount of segments, e.g., in our case there are more than 170k segments. Its disadvantage is that it is difficult to take the melody as a cue for annotation.

Beside of building the dataset, we design the Emotional Lyric and Melody Generator (ELMG) system, which to our best knowledge, is the first attempt to automatically and simultaneously generate lyric and melody with a specific given emotion using deep learning. This is inspired by the great success of sole lyric or melody generation based on deep learning, and the variations of beam search algorithms for guiding the generation process [18], [19]. Specifically, our ELMG consists of the following three parts: 1) Lyric and melody generator: a novel encoder-decoder architecture that can generate lyric and melody by accepting a small piece

Chunhui Bao and Qianru Sun* are with the School of Computing and Information Systems, Singapore Management University. * indicates corresponding author and the email is qianrusun@smu.edu.sg.

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received on Aug 14, 2021; revised on Oct 04 2021 and Jan 17 2022; accepted on Mar 18 2022.

of initial seed lyric as input. 2) Music emotion classifier: a classifier for lyric and melody segments. 3) Emotional beam search (EBS) algorithm: a modified beam search algorithm for controlling the music generation process with a given emotion.

Our contributions are thus four-fold.

- We build large-scale paired lyric-melody dataset with automatic emotion labels consisting of 11,528 MIDI songs.
- We train the encoder-decoder networks based on GRU and Transformer, to generate lyric and melody, simultaneously.
- We propose a modified beam search algorithm EBS to bias the music generation process to match a specific emotion.
- We evaluate the proposed ELMG system with both objective and subjective methods.

II. Related Work

Symbolic music composition. With the advent of large music datasets, deep learning models have recently achieved high-quality results in music composition tasks. DeepBach [20] is proposed by Hadjeres *et al.*, which used a dependency neural network and a Gibbs-like sampling procedure to generate Bach's four parts chorales. Roberts et at. proposed a recurrent variational auto-encoder (VAEs) [21] model to reproduce polyphonic music sequences. Generative adversarial network (GAN) [22] has also been successfully applied to the field of music generation. MuseGAN [7] is a convolutional neural network (CNN) based GAN to compose polyphonic music with 5 sound-tracks. Similarly, recurrent neural network (RNN) based GAN is proposed in C-RNN-GAN [23], which can generate polyphonic continuous music sequence. However, these models mainly trained for generating human-like music, the emotion expression of generated music was ignored. In this work, we focus on how to generate music with a specific given emotion.

Generate music with a given emotion. Music is a way for humans to express their emotions. However, it is too expensive to manually annotate emotion labels for music datasets, which causes great difficulties for music generation tasks conditioned on emotions. In 2019, Ferreira *et al.* [13] proposed a mLSTM based deep generative network, which was the first work to explore deep learning models for symbolic music generation conditioned on emotions. They also built a new music dataset with manually emotion labels called VGMIDI, which consists of 95 labelled piano pieces and 728 unlabelled pieces. In 2020, a GPT-2 model was proposed by Ferreira *et al.* [24] to generate music with a specific emotion and the VGMIDI dataset was extended to 200 labelled data. In [25], a model called CVAE-GAN was proposed for emotion-conditioned symbolic music generation, which synthesized Conditional-VAE and Conditional-GAN [26]. More recently, Hung *et al.* built an emotion-labeled symbolic music dataset called EMOPIA [15], which consists of 1,078 music clips from 387 songs. They also verified that the proposed dataset can be used for generating music conditioned on emotions. Nevertheless, existing music datasets with emotion labels are both small in size. Therefore, we create a new large-scale paired lyric-melody dataset with emotion labels for generating harmonious music that can evoke emotions.

Generate music with lyrics. In recent years, with the advent of music datasets with lyrics, deep learning was also researched for mining musical knowledge between lyrics and melodies. Bao *et al.* [27] proposed Songwriter, which focused on lyric-conditioned music generation. They use a seq2seq network to generate melody from the input lyrics, and then merge the generated melody segments into a complete melody. Yu *et al.* [28], [29] utilized conditional-GAN to generate melody conditioned on the given lyric, in which the generator and discriminator were both LSTM networks with lyric as condition. AutoNLMC [30] proposed by Madhumani *et al.* can create songs with both lyrics and melodies automatically. It was an encoder-decoder LSTM network where the encoder was designed to generate lyric and three decoders are trained to generate pitch, duration and rest of melody respectively. Jukebox [9] trained on raw audio data can also generate music with lyrics. In this work, we propose a novel encoder-decoder architecture for lyric and melody generation. The melody is represented to a sequence of tokens and only one decoder is trained to generate melody, which can be easily controlled to match a particular emotion.

III. Dataset Construction

There is no large-scale music dataset with emotion labels publicly available for emotion-conditioned music generation. In this work, we build a paired lyric-melody music dataset, the details of the new dataset used to generate lyric and melody with emotions are introduced in this section. There are many different ways to represent music for deep learning, the form of music representation in this work is introduced in Section III-A. The basic information of the paired lyric-melody English songs dataset is introduced in Section III-B. The method that we used to annotate music is introduced in Section III-C. The detailed analysis of the annotated dataset is given in Section III-D.

A. Data Representation

Inspired by Yu *et al.* [28], [31] and Madhumani *et al.* [30]. We represent music as a sequence of syllable-note pairs. As shown in Figure 1, lyric as natural language sentences are made up of words. English words are made up of one or more syllables, for example, "do" is made up of one syllable "do" and "doing" is made up of two syllables "do" and "ing". Melody can be defined as a sequence of musical notes. Each note of the melody is represented as a three-dimensional tuple $n = (n_{pitch}, n_{dur}, n_{rest})$:

- n_{pitch}: in music, the pitch represents the frequency of the played note, it can take any integer from 0 to 127.
- n_{dur}: how long a note is played. The standard unit is one beat, if the duration of a note is one beat, denote its duration as 1.0.
- n_{rest}: the duration of the rest before the note. 0.0 means no rest before the note.

TABLE I
EXAMPLES FROM GOEMOTIONS DATASET.

Sample Text	Label(s)
You know the answer man, you are programmed to capture those codes they send you, don't avoid them!	annoyance, surprise
I've never been this sad in my life!	remorse
I don't necessarily hate them, but then again, I dislike it when people breed while knowing how harsh life is.	disappointment, anger
You're right. Sorry for the poor reply.	relief
Absolutely. I'd love it. No matter how much I like the guy, if he just goes for it that's not cool.	embarrassment, joy

TABLE II
EXAMPLES FROM EDMONDS DANCE DATASET.

Sample Text	Label(s)
Just one day in the life. So I can understand. Fighting just to survive. But you taught me I can. We are the lucky ones. We are...	surprise, trust, joy
Hypnotized, this love out of me. Without your air I can't even breathe. Lead my way...	trust, joy
You ruined my life. What you said in your message that night. left me broken and bruised but now i know that you were wrong...	sadness, disgust, anger

TABLE III
CLASSIFICATION RESULTS (%) OF BERT MODELS TRAINED ON GOEMOTIONS DATASET AND EDMONDS DANCE DATASET, TESTED ON EDMONDS DANCE DATASET. "BOTH" MEANS FIRST TRAINED ON GOEMOTIONS DATASET AND THE FINE-TUNED ON EDMONDS DANCE DATASET.

Train dataset	Acc	Precision	Recall	F1 score
GoEmotions	52.44	45.50	55.26	49.93
Edmonds Dance	77.90	81.82	80.67	81.23
Both	**79.02**	**82.85**	**81.88**	**82.31**

Therefore, music segments with length N can be defined as $M = \{m_1, m_2, ..., m_N\}$, where each m_i is a (syllable, pitch, duration, rest) four-dimensional tuple. For simplicity, we do not consider the velocity and tempo of the music. And suppose that the lyrics and melodies can be paired as one-syllable-to-one-note.

B. Data Collection

The dataset used in our work comes from two large-scale MIDI music datasets: LMD-full MIDI dataset [5] and reddit MIDI dataset [6]. MIDI is the abbreviation of musical instrument digital interface, which is an industry standard that describes the interoperability protocol of digital music representation. The MIDI file records all the information of the music and saves it on the computer. There are 176,581 different MIDI files in the LMD-full dataset, but most of them do not contain lyrics. In this work we only use the music data with English lyrics, so only 7,497 MIDI files are selected from the LMD-full dataset. Similarly, the reddit MIDI dataset contains 130k different MIDI files but only 4,031 with English lyrics are selected. Altogether there are 11,528 MIDI files in our dataset. Paired lyric-melody sequences are obtained by parsing the MIDI files as follows:

- Open the file, find out the beginning of the lyric and its corresponding note.
- Store the information of note that has corresponding English syllable.
- If a syllable corresponds to multiple notes, only the information of the first note is recorded.

After parsing, there are 1,971,257 notes in total and the average length of music segments is 171 notes. The pitch distribution of these selected songs is shown in Figure 2a, from which we can see that the pitch distribution approximately obeys a normal distribution with a mean of 66.58 and a standard deviation of 9.96. Similarly, the duration distribution is shown in Figure 2b, we can observe that most of them fall in the interval [0.5, 2.0], and the mode is 1.0. Rest distribution is shown in Fighre 2c, we can observe that most of the rests are zero. For the lyrics, there are 20,934 unique syllables and 20,268 unique words in total.

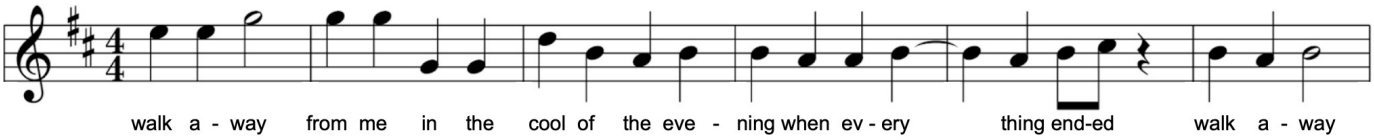
C. Data Annotation

For the above large-scale dataset, manually labelling emotions expressed in music by humen is expensive. Therefore, in this work we exploit the deep learning models to automatically annotate the paired lyric-melody dataset. There are many datasets that can be used to train the annotator, such as large-scale social media or dialog datasets with emotion labels [32], relatively small-scale lyric datasets for lyric emotion classification [17], [33], [34] and small-scale emotion-labelled music datasets without lyric [13], [15]. In this section, we explore the reliable method to train the annotator.

Understanding emotions expressed in natural language has been widely researched in resent years. The largest human annotated dataset for text sentiment classification is GoEmotions [32], which consists of 58k carefully selected Reddit comments and labelled for 27 emotion categories or neutral. Table I shows illustrative samples of GoEmotions dataset, each sample text has one or more corresponding labels. For music sentiment analysis, we don't need so many emotional categories, so we group the 27 categories according to positive and negative binary classification [35], the labels are divided into 4 categories as shown in follows:

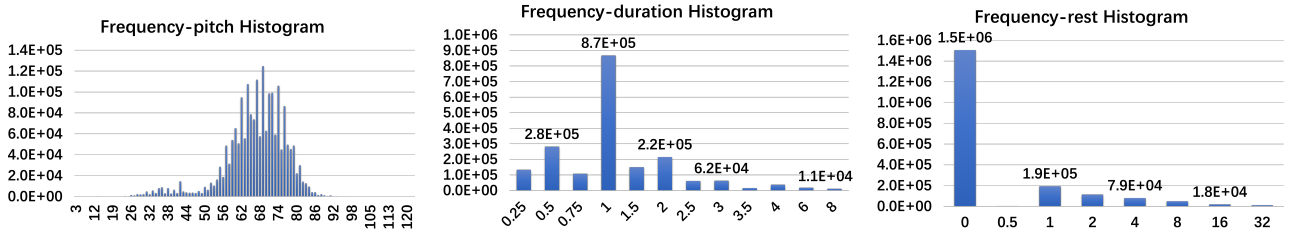
- **positive:** admiration, amusement, approval, caring, desire, excitement, gratitude, joy, love, optimism, pride, relief
- **negative:** anger, annoyance, disappointment, disapproval, disgust, embarrassment, fear, grief, nervousness, remorse, sadness
- **ambiguous:** confusion, curiosity, realization, surprise
- **neutral**

The advantage of GoEmotions dataset is its large scale, but the disadvantage is that there's a domain gap between Reddit comments and song lyrics.



Pitch	80	80	83	83	83	71	71	78	75	73	75	75	73	73	75	73	75	77	75	73	75
Duration	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	0.5	0.5	1.0	1.0	2.0
Rest	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

Fig. 1. An example of paired lyric-melody music data. Each note of the music is represented as a four-dimensional tuple $n = (n_{syllable}, n_{pitch}, n_{dur}, n_{rest})$.



(a) Pitch distribution of the whole dataset. (b) Duration distribution of the whole dataset. (c) Rest distribution of the whole dataset.

Fig. 2. Melody distribution of the collected dataset. (a), (b) and (c) show the distribution of pitch, duration and rest of the whole dataset respectively.

There's some relatively small-scale lyric datasets manually labelled according to human emotions. Recently, Edmonds *et al.* constructed Edmonds Dance dataset [17], which consists of lyrics retrieved from 524 English songs. As shown in Table II, there's 8 emotion categories in the Edmonds Dance Dataset and each song has one or more corresponding labels. Same as GoEmotions, the 8 categories are grouped into positive, negative or ambiguous:

- **positive:** anticipation, joy, trust
- **negative:** anger, disgust, fear, sadness
- **ambiguous:** surprise

In order to have a common model for emotion classification, we train Bert-base [16] models on GoEmotions and Edmonds Dance Dataset. Bert stands for Bidirectional Encoder Representations from Transformers [36], which has been pre-trained on large-scale natural language datasets and given state-of-the-art results on a wide variety of natural language processing tasks. When train the Bert model, the learning rate is set to 5e-5 with gradually decay. The model fine-tuned for 10 epochs with the warm-up proportion as 0.1 and batch size as 16. Because there's no domain gap between Edmonds Dance Dataset and our dataset, we randomly select 1/10 data from the Edmonds Dance Dataset as test data. The experimental results are shown in Table III, to our surprise, the Bert model trained on GoEmotions dataset has relatively worse performance for lyric emotion classification. It means that the emotion classifiers trained on large-scale out-of-domain data do not generalize well to song lyrics. However, the Bert model directly trained on Edmonds Dance Dataset achieves better performance, despite the in-domain dataset is magnitude smaller than out-of-domain dataset. In addition, pre-training the Bert model on GoEmotions dataset and then fine-tuning the model on Edmonds Dance Dataset can slightly improve the classification accuracy of song lyrics.

In addition to lyrics, is there any way that can utilize the melodies for annotation? In order to answer this question, we train deep learning models on the EMOPIA dataset [15] and evaluate if they can be used on our dataset. The EMOPIA dataset consists of 1,078 clips from 387 piano solo performances. They are labelled corresponding to the Russell's 2-dimensional model [37], which represents music emotion using a valence-arousal pair. Arousal indicates emotion intensity and valence indicates the positive or negative emotion. Thus, the clips with high valence label can be considered as positive data and the clips with low valence label are negative data. We train a bidirectional LSTM with self-attention to classify the music clips according to their valence, and achieves 83.3% test accuracy on EMOPIA dataset. Then, this model are used to classify the melodies of our dataset, the results are shown in Table IV, we can see that the classification results are catastrophically unbalanced, even though the training data in EMOPIA dataset is balanced. We also manually verify randomly selected data of the classification results, the unanimous ratio is less than 50%. Therefore, we think the deep learning model trained on EMOPIA cannot be used to annatate our dataset because of the following reasons: 1) There's a domain gap between piano solo performances and pop songs' melodies in our dataset. 2) EMOPIA is a small-scale dataset. 3) The unanimous ratio of automatic labelling and manual labelling is less than 50%.

D. Data Analysis

The 11,528 MIDI files are cut into small music segments with fixed length N (20, 50 or 100), and gets 103,540, 43,902,

TABLE IV
ANNOTATION RESULTS OF THE BI-LSTM TRAINED ON EMOPIA FOR OUR DATASET, ALL SEGMENTS ARE LABELLED TO HIGH-VALENCE OR LOW-VALENCE.

Length	Annotations		Total
	High-valence	Low-valence	
20	25,743	77,797	103,540
50	7,069	36,833	43,902
100	2,699	20,163	22,862

TABLE V
ANNOTATION RESULTS FOR OUR DATASET, ALL SEGMENTS ARE LABELLED TO POSITIVE, NEGATIVE OR UNLABELLED.

Length	Annotations			Total
	positive	negative	unlabelled	
20	48,659	17,019	37,862	103,540
50	18,557	9,341	16,004	43,902
100	8,968	5,712	8,182	22,862

22,862 segments respectively for N equals to 20, 50, or 100. Then, the Bert model trained on GoEmotions dataset and then fine-tuned on Edmonds Dance Dataset is used to annotate these music segments. Specifically, if the confidence is greater than 95%, mark the lyric as positive or negative, others are unlabelled. If more than one labels have confidence greater than 95%, then choose the higher one.

Table V shows the annotation results for our dataset, from which we can see that about 64% are labelled, and the number of positive segments is larger than the number of negative segments. Examples form the annotated dataset are shown in Table VI. Detailed quantitative comparison of melody distributions is shown in Table VII, it shows that the pitch, duration and rest distributions of positive and negative samples are pretty similar to the whole dataset. We also measure the major-minor tonality of the music segments by using Krumhansl-Kessler algorithm [38]. We can see that the major-minor tonality distributions of positive data and negative data are a little bit different, which indicates that when people create emotional-positive music, they prefer to use major keys, but when they create emotional-negative music, more minor keys are used.

TABLE VI
EXAMPLES FROM ANNOTATED DATASET.

Sample Text	Label(s)
When I look into your eyes your love is there for me And the more I go inside the more there is to see	positive
I believe in angels Something good in everything I see I believe in angels When I know the time is right for me	positive
Please forgive me I stop loving you deny me this pain going through Please forgive me I need you	negative
Please forgive me I know not what I do Please forgive me I stop loving you deny me this pain	negative
Quit playing games with my heart With my heart my heart I should have known from the start	unlabelled

TABLE VII
DETAILED QUANTITATIVE COMPARISON OF MELODY DISTRIBUTIONS, INCLUDE THE WHOLE DATASET (WD), POSITIVE AND NEGATIVE.

Items	WD	Positive	Negative
Mean value of pitch	66.58	66.34	66.63
Standard deviation of pitch	9.96	9.98	10.11
Number of pitch value	98	85	79
Mode of duration	1.0	1.0	1.0
Number of duration value	19	19	18
Percentage of 1.0	45.17	45.68	48.83
Mode of rest	0.0	0.0	0.0
Number of rest value	8	8	8
Percentage of 0.0	80.25	83.55	86.66
Percentage of major keys	**57.73**	**61.59**	**55.83**
Percentage of minor keys	**42.27**	**38.41**	**44.17**

IV. METHODOLOGY

The proposed ELMG system is designed to generate lyric and melody with a required specific emotion given a piece of seed lyric. A general overview is shown in Algorithm 1 and Fighre 3. It receives the labelled and unlabelled music segments built in Section III, a required emotion and a piece of seed lyric as input. Firstly, syllable-level and word-level skip-gram models are trained on the whole dataset, which aim at mapping each English word and syllable to a vector [39]. Then, an encoder-decoder model is trained end-to-end as the lyric and melody generator, in which the encoder is lyric generator and the decoder is melody generator, its structure is illustrated in Section IV-A. Next step, a music sentiment classifier is trained on the labelled data, which is demonstrated in Section IV-B. Finally, an emotional beam search (EBS) algorithm is proposed in Section IV-C, it takes the required emotion, lyric and melody generator, music sentiment classifier and a piece of seed lyric as input and output a music segment.

A. Lyric-melody Generator

The architecture of the proposed lyric-melody generator is shown in Figure 3, which is a sequential encoder-decoder model trained end-to-end to compose lyrics and melodies.

The encoder is designed as lyric generator and lyric encoder. It takes a sequence of English syllables as input, denoted as $S = \{s_1, s_2, ..., s_T\}$. The lyric embedding layers are skip-gram models [39] trained on the whole lyrics dataset, we keep most of the hyper-parameter settings in [28] for training the skip-gram models: tokens context window $c = 7$, negative sampling distribution parameter $\alpha = 0.75$, and the learning rate is set to 0.03 with a gradually decay. After training, we obtain word-level and syllable-level embedding models,

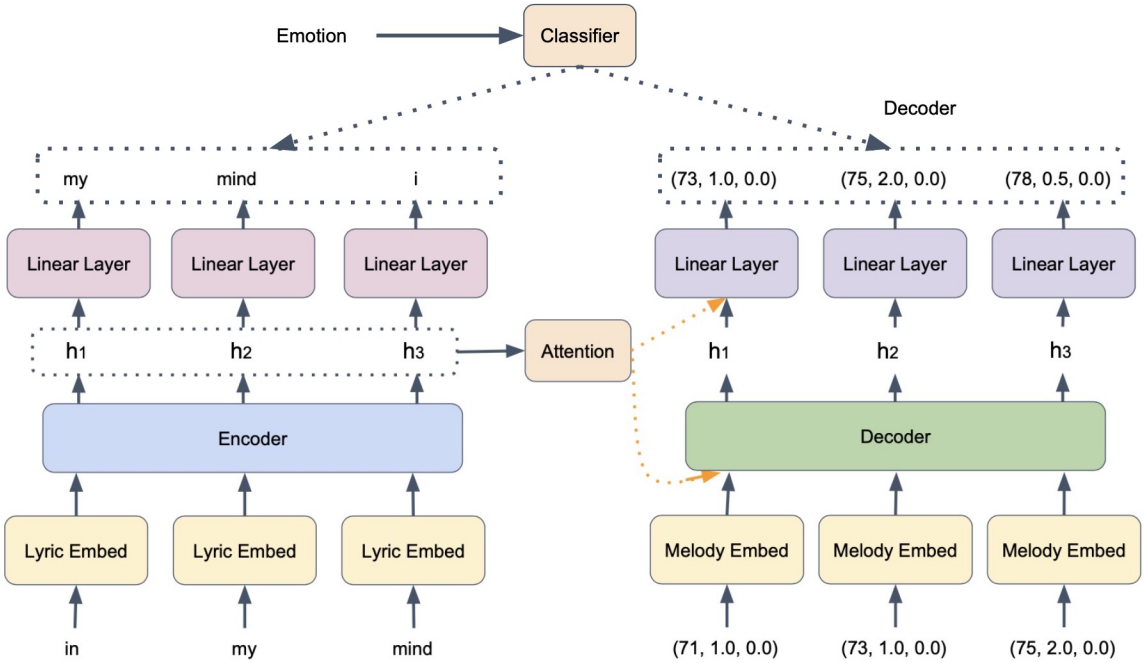


Fig. 3. The architecture of the proposed ELMG system. The encoder is designed as lyric generator and lyric encoder. It takes a sequence of lyric as input. Input the lyric into lyric embedding layer, vector representation of the input lyric will output. then, the encoder network takes the vector representation of the lyric as input and output a sequence of hidden states, $H = \{h_1, h_2, ..., h_T\}$. These output hidden states are the representations of the input lyric, input them into a fully connected layer, the predictions of next tokens of the input lyric will output. Similarly, the function of the decoder is to generate melody conditioned on the input lyric. It takes the melody sequence as input, $M = \{m_1, m_2, ..., m_T\}$, where each m_i is a (pitch, duration, rest) three-dimensional tuple. Then, input M into melody embedding layer and decoder network, a sequence of hidden states will output. These hidden states are the vector representations of the melody, input them into a fully connected layer, the predictions of next tokens of the input melody will output. In addition, attention mechanism is used to insure that lyric is taken into consideration during the decoder computing process [36], [40]. After training, the generation process is controlled by a classifier using the EBS algorithm.

Algorithm 1 Emotional Lyric and Melody Generator

Require: labelled and unlabelled dataset X_l and X_u, required emotion e, piece of seed lyric m

Initialize word embedding E_w
Initialize syllable embedding E_s
for $x \in X_l \cup X_u$ **do**
  Update E_w and E_s
end for
Initialize lyric and melody generator G
for $x \in X_l \cup X_u$ **do**
  Update G
end for
Initialize music sentiment classifier C
for $x \in X_l$ **do**
  Update C
end for
$y \leftarrow \mathrm{EBS}(G, C, m, e)$
return y, E_w, E_s, G, C

denoted as $E_w(\cdot)$ and $E_s(\cdot)$ respectively. For a syllable s comes from word w, it can be represented as the concatenation of $E_w(w)$ and $E_s(s)$, denoted as $E_w(w)||E_s(s)$. Then, the output vectors of lyric embedding layers are input into the encoder.

The encoder takes the whole syllable sequence S as its input and output a sequence of hidden states as the representation of the input lyric, $H = \{h_1, h_2, ..., h_T\}$. These output hidden states are used for lyric generation. Input H into a fully connected layer, for every unit, the lyric generator is modeled to predict the next syllable token conditioned on all the previous syllables in the input sequence. Thus, the goal of encoder can be represented as learning the following probability distribution:

$$p(S) = \prod_{t=1}^{T} p\left(s_t \mid s_1, s_2, \ldots, s_{t-1}\right). \tag{1}$$

Here, in order to overcome dull and repetitive outputs problem, we use unlikelihood training [41] to train the encoder

$$L_{lyric} = -\frac{\alpha}{t-1} \sum_{i=1}^{t-1} \log(1 - p_\theta\left(s_i\right)) - \log p_\theta\left(s_t \mid s_{<t}\right). \tag{2}$$

where α is a real value hyper-parameter. While increasing the probability of the true target token, the unlikelihood loss reduce the probability of the tokens that have appeared before the true target token in the sentence, forbid the model using high frequency tokens repeatedly in a sentence.

The decoder takes the corresponding melody sequence as input, $M = \{m_1, m_2, ..., m_T\}$, where each m_i is a (pitch, duration, rest) three-dimensional tuple. Firstly, each m_i is converted to a word form representation, for example, $m =$

$(70, 1.0, 0.0)$ are denoted as 'p_70 d_1.0 r_0.0', then the melody notes can be input into embedding layer as normal words. The output of decoder is also a sequence of hidden states, $\tilde{H} = \{\tilde{h}_1, \tilde{h}_2, ..., \tilde{h}_T\}$, which is the representation of the input melody. In addition, attention mechanism [36], [40] is used to insure that lyric is taken into consideration during the decoder computing process. The same as encoder, these output hidden states are input to a fully connected layer, for every unit, the melody generator is learned to predict the next melody note conditioned on previous melody notes and the corresponding lyric, which means that the melody generator is modeled to learn the following probability distribution

$$p(M|S) = \prod_{t=1}^{T} p\left(m_t \mid m_1, m_2, \ldots, m_{t-1}, S\right). \tag{3}$$

Similar to encoder, the decoder is trained to minimize the following unlikelihood loss function

$$L_{melody} = -\frac{\alpha}{t-1} \sum_{i=1}^{t-1} \log(1 - p_\theta\left(m_i\right)) - \log p_\theta\left(m_t \mid m_{<t}\right). \tag{4}$$

where α is a real value hyper-parameter.

Combine the loss function of the encoder and decoder, the lyric-melody generator is trained to minimize the total loss defined as

$$L = L_{lyric} + \lambda L_{melody}, \tag{5}$$

where λ is a real value hyper-parameter.

B. Music Emotion Classifier

In order to control the music generation process, we train a music emotion classifier by using the labelled data. It takes a sequence of music, $C = \{c_1, c_2, ..., c_T\}$, as input, each c_i is a (syllable, pitch, duration, rest) four-dimensional tuple. The syllable is converted to a vector and then the three-dimensional music note (pitch, duration, rest) is also embedded as a vector. These two vectors are concatenated to represent a music note c_i. Next step, bidirectional long short-term memory (LSTM) network and multi-head self-attention Transformer [36] encoder are trained to predict the label of the input music sequence C.

C. Music Generation with Emotions

In this section, we describe how to use the music emotion classifier to control the process of music generation to match a particular emotion. Beam search is a commonly used algorithm for text generation and neural machine translation [42], which selects the best and most likely words for the target sequence. In this work, the music generator is required to generate music not only harmonious but also perceived to have a specific emotion. For that we propose emotional beam search (EBS), a modified beam search algorithm guided by the music emotion classifier as illustrated in Section IV-B.

The EBS algorithm takes an initial seed lyric with length n, lyric and melody generator G, music emotion classifier C, beam size $b_1 \& b_2 \& b_3$ as input, output a piece of music with required emotion e of length N, where $n < N$.

As shown in Figure 4, assuming that a piece of music with length t has been generated, which consists of a piece of lyric $S = \{s_1, s_2, ..., s_t\}$ and a piece of melody $M = \{m_1, m_2, ..., m_t\}$. The probability of x_i being the next lyric token can be calculated by using softmax function to the output of encoder at position t

$$p(s_{t+1} = x_i|S) = \frac{\exp\left(e_{ti}\right)}{\sum_{k=0}^{|V_s|} \exp\left(e_{tk}\right)}, \tag{6}$$

where e_{ti} represents the i-th element of the output of encoder at position t, $|V_s|$ is the number of syllables in the vocabulary. The higher the probability, the more fluent lyrics are generated.

Similarly, the probability of y_i being the next melody note can be calculated by

$$p(m_{t+1} = y_i|S, M) = \frac{\exp\left(d_{ti}\right)}{\sum_{k=0}^{|V_m|} \exp\left(d_{tk}\right)}, \tag{7}$$

where d_{ti} represents the i-th element of the output of decoder at position t, $|V_m|$ is the length of melody vocabulary. Music note with high probability means the generated melody sound harmonious.

After calculating the probabilities of all tokens by using equation 6 and equation 7, b_1 lyric tokens and b_2 melody tokens with highest probabilities are selected, therefore, $b_1 * b_2$ candidate lyric-melody pairs are chosen in total, $\{(x_i, y_j)|i = 1, ..., b_1; j = 1, ..., b_2\}$.

Adding every candidate lyric-melody pair (x_i, y_j) to the original music piece (S, M), the probability that the new music piece is perceived to have a specific emotion e can be computed by the music emotion classifier

$$p(e|(S, M)||(x_i, y_j)) = \frac{\exp\left(e\right)}{\sum_{j=1}^{E} \exp\left(e_j\right)}, \tag{8}$$

where E is the number of emotions in the dataset and $||$ represents the concatenation operation. After calculating the probabilities of all candidate lyric-melody pairs, b_3 music segments with length $t + 1$ that have highest probabilities to represent the required emotion e are generated.

Therefore, there's b_3 segments of each length, and for every segment, $b_1 * b_2$ candidate lyric-melody pairs should be evaluated. So the computational complexity of EBS is $O(N * b_1 * b_2 * b_3)$ where N is the required length.

V. Experiments and Evaluation

Experimental setup, evaluation methods and experimental results are introduced in this section. The empirical evaluation of the proposed ELMG system is divided into three parts. First, we evaluate the accuracy of the music emotion classifier in Section V-A. Then, the experimental setup and objective evaluation of the lyric-melody generator are demonstrated in Section V-B. Finally, the subjective evaluation of the generated music is shown in Section V-C. The code of this work can be downloaded at https://github.com/BaoChunhui/Generate-Emotional-Music.

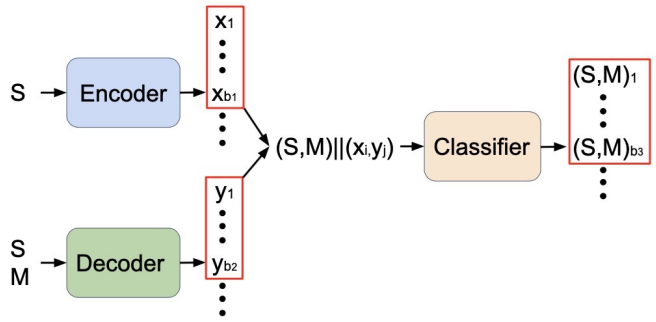


Fig. 4. The schematic diagram of EBS algorithm. Assuming that a piece of music with length t has been generated, which consists of a piece of lyric $S = \{s_1, s_2, ..., s_t\}$ and a piece of melody $M = \{m_1, m_2, ..., m_t\}$. Input S to encoder, b_1 syllables are selected; input S and M to decoder, b_2 melody notes are selected. Then concatenate each candidate lyric-melody pair (x_i, y_j) to the original music piece (S, M) and input to the classifier, b_3 music segments with length $t + 1$ that have highest probabilities to represent the required emotion are generated.

TABLE VIII
EMOTION CLASSIFICATION ACCURACY (%) OF LSTM AND TRANSFORMER ON DIFFERENT DATASETS WITH DIFFERENT LENGTH.

datasets	Length		
	20	50	100
Bidirectional LSTM	99.8	**99.9**	**99.9**
Self-attention Transformer	**100.0**	**99.9**	**99.9**

A. Emotion Classifier

As demonstrated in Section IV-B, both bidirectional LSTM and Transformer are trained on labelled data to classify the music emotion. As shown in Table V, the number of positive samples is larger than the number of negative samples, so over-sampling method is used to overcome the imbalance problem of the dataset, which means repeatedly using negative samples in every epoch, so that the ratio of positive and negative samples in the training data is close to 1:1.

For bidirectional LSTM, The number of layers is set to 6 and the dimension of hidden state is set to 256. The learning rate is set to 1e-4 with gradually decay. The number of epochs is 30 and the dimension of embedding vector is set to 256. For Transformer, The number of Transformer blocks is set to 6, each Transformer block consists of an 8-head self-attention Transformer encoder layer connected with a LayerNorm [43]. The dimension of input is set to 128. The learning rate and number of epochs are the same with bidirectional LSTM.

We evaluate the classifiers using a 8-fold cross validation approach, in which the testing fold and the training folds have no overlapping data. Table VIII shows the emotion classification accuracy of all datasets created in Section III-C, from it we can see that both the LSTM and Transformer based models can successfully classify the datasets. Therefore we can use the classifier trained on labelled data of the datasets in EBS algorithm.

B. Music Generation

The lyric-melody generator is an encoder-decoder model trained end-to-end on the unlabelled datasets. 9/10 of them are used in the training process and 1/10 are used to evaluate the trained sequence to sequence model. Both GRU and Transformer based neural networks are trained for lyric-melody generation.

For GRU, the encoder and decoder have the same neural structure. The number of layers is set to 4 and the dimension of hidden state is set to 256. The initial hidden state of encoder is initialized with zero vector, and the initial hidden state of decoder is initialized with the last hidden state of encoder. All parameters are initialized from zero mean, 0.08 variance Gaussian distribution. For Transformer, both the encoder and decoder have 12 Transformer blocks, the number of head is set to 16 and the input dimension is set to 256. The loss function is optimized by Adam optimizer with initial learning rate of 1e-4 and decayed after every epoch. The α in equation 2 and equation 4 is set to 1. The λ in equation 5 is set to 1. The batch size is set to 64, 32, 16 for datasets with length 20, 50, 100 respectively.

Figure 5 shows the training process of the GRU based model. When model trained for 0, 1, 5, 10 and 30 epochs, one music segment is generated by using beam search algorithm with beam size 3. We can see that the generated music notes become more and more varied, and the generated lyrics become more and more fluent.

During the training process, over-sampling method is also used for the negative samples in every epoch, ensuring that the ratio of positive and negative samples used to train the generator is close to 1:1.

After training, the GRU and Transformer based networks are evaluated by using the test data. Input test data into the sequence to sequence model, melodies can be generated. We use sequence-level unlikelihood objective [41] during the generation process to eliminate the duplication phenomenon and increase diversity, which means that the predicted probabilities of tokens occurred in the generated sequence are decreased. So the model cannot use high frequency tokens too often. We also implement AutoNLMC on our dataset for comparison, which is a sequence to sequence model consists of one encoder and multiple decoders proposed in [30]. Different from our method, AutoNLMC regards each attribute of the melody as independent and trains decoders separately for each attribute. Then, we compare the melody distributions of ground-truth melodies and melodies generated by AutoNLMC, GRU based model as well as Transformer based model. Detailed quantitative comparison of melody distributions are shown in Table IX and the frequency distribution histograms are shown in Figure 6. In addition, in order to further compare these three generators, the training and testing loss, training and testing perplexity, as well as Jensen-Shannon divergence between the ground-truth distribution and generated distributions are given in Table X. Compared with AutoNLMC, the quality of pitches generated by our models is better, since the pitch distributions generated by our models have higher standard deviation and lower Jensen-Shannon divergence, which means that the

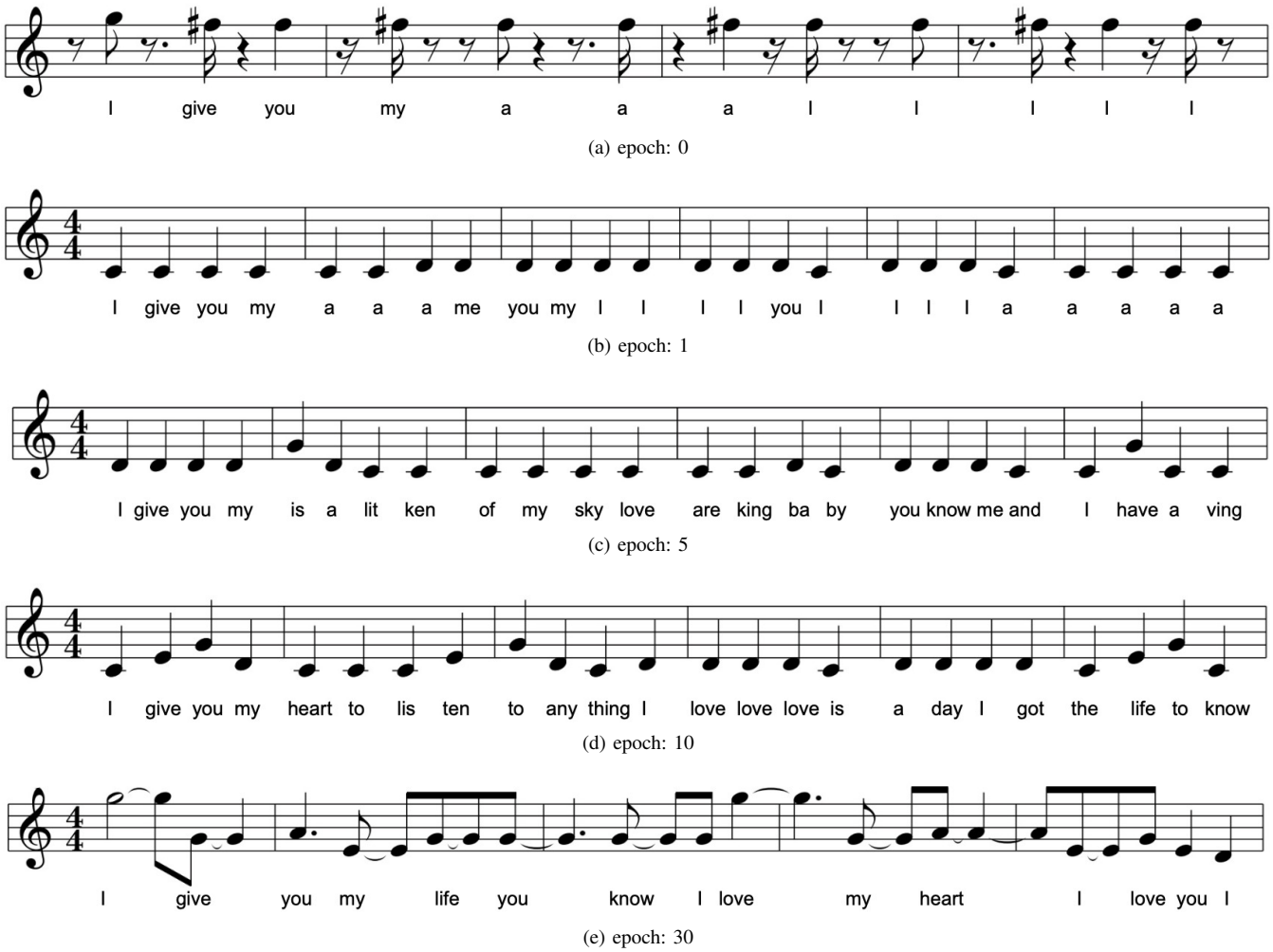


Fig. 5. Generated music segments when model trained for 0, 1, 5, 10 and 30 epochs respectively. The generated music notes become more and more varied, and the generated lyrics become more and more fluent.

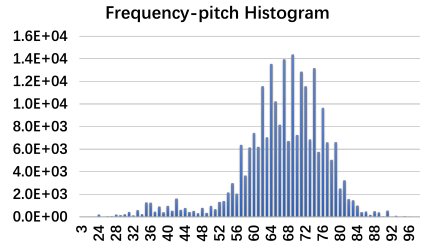
pitches generated by our models are more diverse and closer to the ground-truth data. But the disadvantage of our models is that the generated duration and rest have lower diversity than ground-truth data and AutoNLMC-generated data. Moreover, the training and testing loss, training and testing perplexity of Transformer based generator are much lower than AutoNLMC and GRU based generator. It demonstrates the Transformer has stronger learning ability and can better fit the dataset.

Then, we generate lyrics and melodies by using the EBS algorithm introduced in Section IV-C. We use 5 different seed lyrics: "I give you my", "but when you told me", "if I was your man", "I have a dream", "when I got the" and different generators trained on various datasets (length = 20, 50, 100) with various skip-gram models (dimension = 10, 50, 100, 128). For the EBS algorithm, the beam size is set to $(b_1 = 3, b_2 = 3, b_3 = 5)$ and the maximum length is set to 25. We generate 180 segments by using the GRU based generator and LSTM based classifier, in which 60 are positive, 60 are negative and 60 are uncontrolled. Similarly, 180 segments are generated by using the Transformer based generator and Transformer based classifier. Generated samples with required emotion are shown in Figure 7. Then we use the fine-tuned Bert model introduced in Section III-C to objectively evaluate them. The evaluation results are shown in Table XI, which shows that the EBS algorithm successfully controlled the generation process.

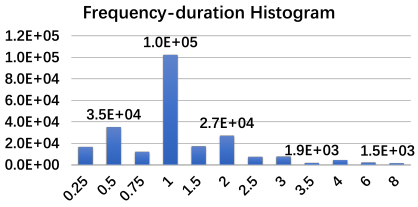
Without control, the generator don't consider the emotions of the generated segments during the generation process. By using EBS algorithm, the generation process is controlled by the music emotion classifier. By using the Bert annotator to measure the generated segments, we can see that the EBS algorithm obviously bias the generation process towards the given emotion. When the required emotion is "positive", more than 75% of the generated music segments are correctly identified as positive by the annotator, and almost no segment is identified as negative. Likewise, when the required emotion is "negative", most of the generated segments is classified as negative and almost no segment is identified as positive. We also observe that EBS algorithm can applied to both traditional GRU or LSTM based model and Transformer based model.

C. Subjective Evaluation

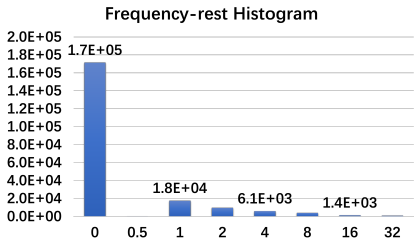
Although objective evaluation indicate that the model is able to generate harmonious lyric and melody to capture the required emotion, it is still difficult to conclude that the



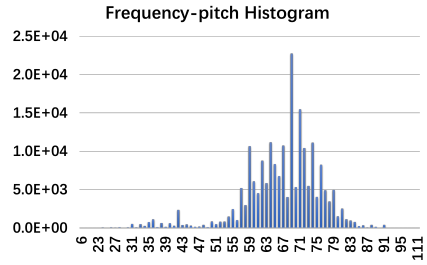
(a) Pitch distribution of ground-truth melody.



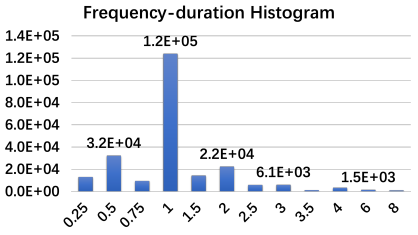
(b) Duration distribution of ground-truth melody.



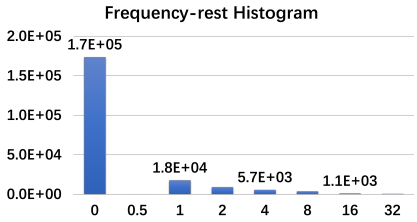
(c) Rest distribution of ground-truth melody.



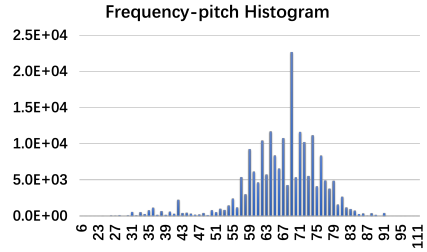
(d) Pitch distribution of melody generated by AutoNLMC.



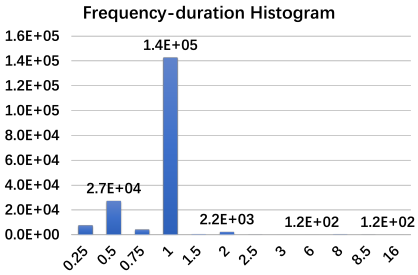
(e) Duration distribution of melody generated by AutoNLMC.



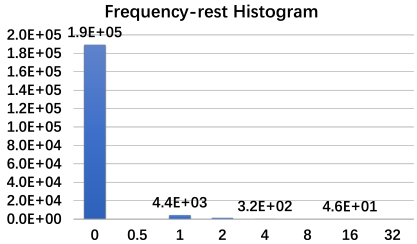
(f) Rest distribution of melody generated by AutoNLMC.



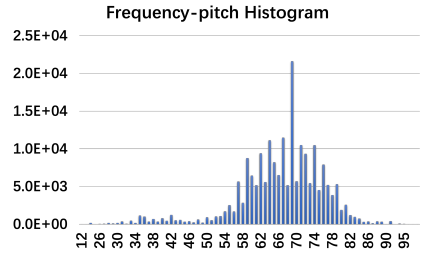
(g) Pitch distribution of melody generated by GRU.



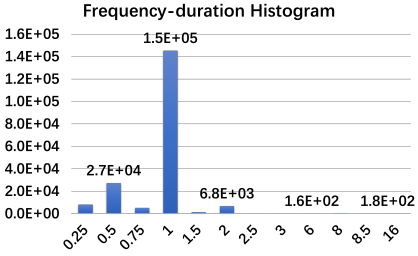
(h) Duration distribution of melody generated by GRU.



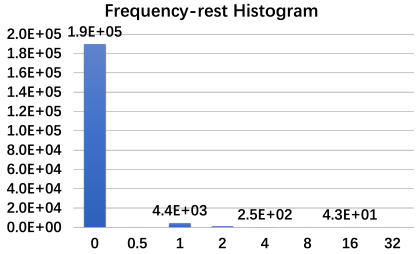
(i) Rest distribution of melody generated by GRU.



(j) Pitch distribution of melody generated by Transformer.



(k) Duration distribution of melody generated by Transformer.



(l) Rest distribution of melody generated by Transformer.

Fig. 6. Distributions of ground-truth melody and generated melody on the testing dataset. (a), (b), (c) show the distribution of ground-truth melody; (d), (e), (f) show the distribution of melody generated by AutoNLMC; (g), (h), (i) show the distribution of melody generated by GRU; (j), (k), (l) show the distribution of melody generated by Transformer.

generated music pieces please human ears and evoke emotions in listeners' hearts. Music composition is a human creative process, so we adapt the subjective evaluation method to evaluate generated lyrics and melodies by our ELMG system. We invited volunteers to evaluate the music data selected from the ground-truth dataset, music segments generated by GRU based model and Transformer based model.

Firstly, the participants should offer their basic information, include their name, age, gender and musicianship experience. Musicianship experience was assessed using a 5-point scale where 1 to 5 means "I've never studied music theory or practice", "I've studied music theory or practice within two years", "I've studied music theory or practice for two to five years", "I've studied music theory or practice for more than five years" and "I have an academic degree in music" respectively. Then, each participant needs to evaluate 18 music pieces. For each piece of music, first play the melody to the participants and ask the participants to classify the emotion conveyed by the melody (positive or negative). Next, the lyric of the this music piece is given to participants. Participants

TABLE IX
DETAILED COMPARISON OF GROUND-TRUTH MELODY DISTRIBUTION AND GENERATED MELODY DISTRIBUTIONS.

Items	Ground-Truth	AutoNLMC	GRU	Transformer
Mean value of pitch	66.55	66.71	66.62	66.51
Standard deviation of pitch	**10.05**	**9.31**	**9.40**	**9.64**
Number of unique pitch value	82	81	81	76
Maximum pitch value	111	111	111	101
Minimum pitch value	3	6	6	12
Mode of duration	1.0	1.0	1.0	1.0
Number of unique duration value	19	18	18	18
Maximum duration value	32.5	32.0	32.0	32.0
Minimum duration value	0.25	0.25	0.25	0.25
Percentage of 1.0 (%)	**43.63**	**52.58**	**74.22**	**76.83**
Mode of rest	0.0	0.0	0.0	0.0
Number of unique rest value	8	8	8	8
Maximum rest value	32.0	32.0	32.0	32.0
Percentage of 0.0 (%)	**80.65**	**81.96**	**96.67**	**96.86**

TABLE X
DETAILED COMPARISON OF AUTONLMC, GRU BASED GENERATOR AND TRANSFORMER BASED GENERATOR. HERE "TRANS." STANDS FOR TRANSFORMER BASED GENERATOR, "JSD" AND "GD" ARE THE ABBREVIATION OF JENSEN-SHANNON DIVERGENCE AND GROUND-TRUTH RESPECTIVELY.

Items	AutoNLMC	GRU	Trans.
Training loss	7.60	7.75	**4.79**
Testing loss	8.69	8.59	**5.21**
Training perplexity	2004.85	2316.11	**120.77**
Testing perplexity	5967.03	5369.26	**183.74**
Pitch JSD vs GD	.0186	.0140	**.0111**
Duration JSD vs GD	**.0069**	.1174	.1499
Rest JSD vs GD	**.0025**	.0694	.0720

TABLE XI
OBJECTIVE EVALUATION OF THE GENERATED MUSIC PIECES. THE CLASSIFIER USED IN EBS AND FINE-TUNED BERT ANNOTATOR ARE UTILIZED TO EVALUATE THE GENERATED LYRICS. "P", "N" AND "U" REPRESENT POSITIVE, NEGATIVE AND UNCONTROLLED RESPECTIVELY.

	Positive	**Negative**	**Unlabelled**	**Total**
GRU P	**45**	0	15	60
GRU N	2	**33**	25	60
GRU U	23	17	20	60
Transformer P	**47**	0	13	60
Transformer N	1	**37**	22	60
Transformer U	25	14	21	60

should to classify the emotion conveyed by this music segment again according to the lyric, in this step, participants are not allowed to change the classification answer of previous question but can make a different decision about the emotion conveyed by this music segment. Finally, we ask the following questions to participants

- Is this melody agreeable to the ears?
- Is this lyric meaningful?

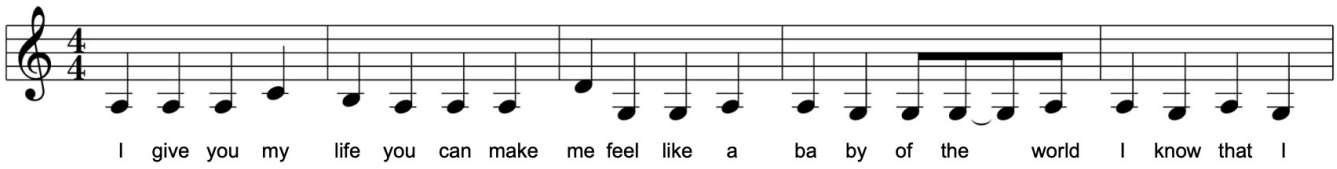
TABLE XII
CLASSIFICATION ACCURACY (%) FOR GROUND-TRUTH (GT) AND GENERATED MUSIC SEGMENTS. HIGHER IS BETTER.

	GT	GRU	Transformer
Positive lyrics	**100.0**	**95.0**	**96.7**
Negative lyrics	**96.7**	**91.7**	**93.3**
Positive melodies	48.3	51.7	50.0
Negative melodies	53.3	46.7	55.0

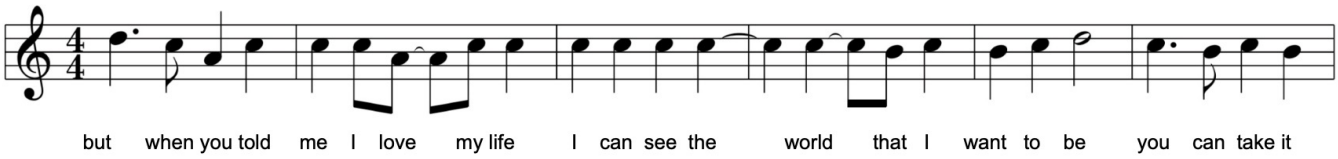
- Are the lyrics and melody compatible?

Participants answer the above questions on a five point discrete scale where 1 to 5 corresponds to "Very bad", "Bad", "Ok", "Good" and "Very good" respectively.

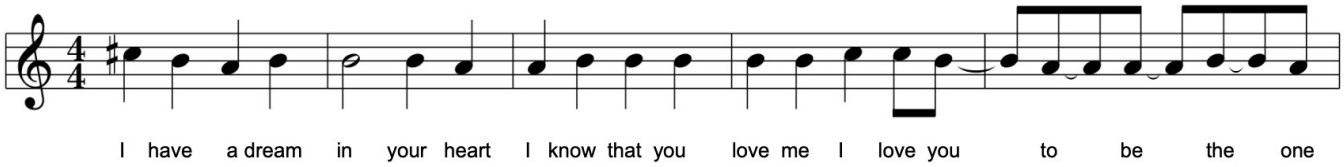
We invited 20 participants for our subjective evaluation, where 10 are male and 10 are female. They have an average age of approximately 24.5 years and the average musicianship experience is 2.45. Detailed subjective classification results are shown in Table XII. We can see that only by listening to the melodies, participants cannot distinguish the emotion of the music segments. Even on the ground-truth data, the classification accuracy is about 50%. After reading the lyrics, the classification accuracy has increased to more than 90%. This demonstrates that emotions in our dataset are mainly conveyed by lyrics and the ELMG system proposed by us successfully learned to generate lyrics to represent the required emotion. We also investigate the quality of music segments by asking questions, such as "Is this melody agreeable to the ears?", "Is this lyric meaningful?" and "Are the lyrics and melody compatible?". The results are shown in Figure XIII. We can see that both GRU based generator and Transformer based generator can successfully generate music segments of almost the same high quality as the training dataset. Even though Transformer has stronger learning ability and can better fit the training data, the quality of music segments generated by Transformer dose not obviously beyond GRU. We think that the quality of the dataset is the bottleneck of our ELMG



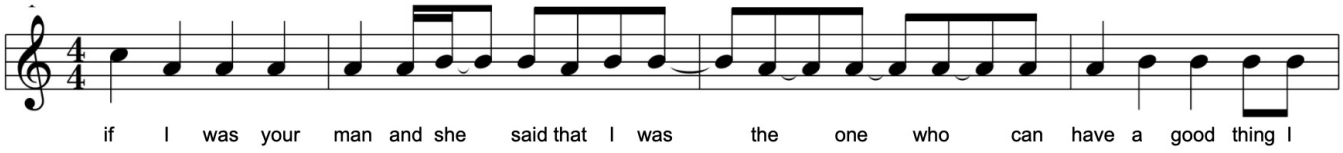
(a) Generated by GRU, the required emotion is positive and the seed lyric is 'I give you my'.



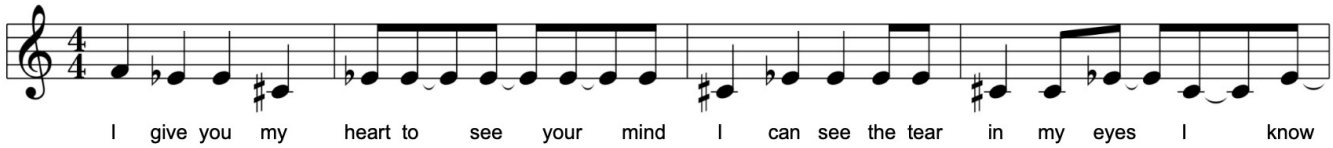
(b) Generated by GRU, the required emotion is positive and the seed lyric is 'but when told me'.



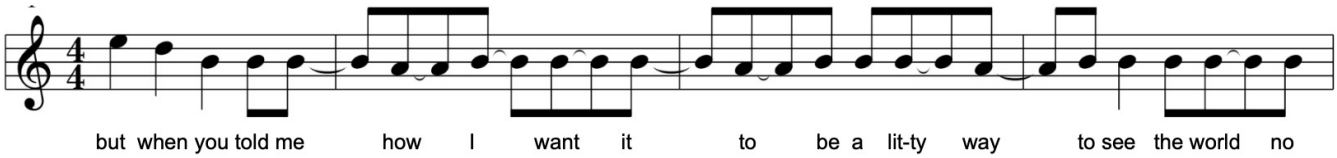
(c) Generated by Transformer, the required emotion is positive and the seed lyric is 'I have a dream'.



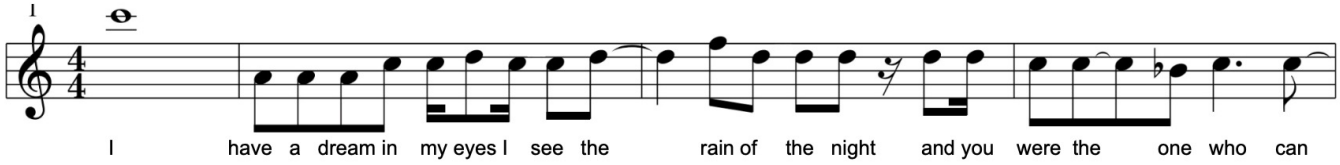
(d) Generated by Transformer, the required emotion is positive and the seed lyric is 'if I was your man'.



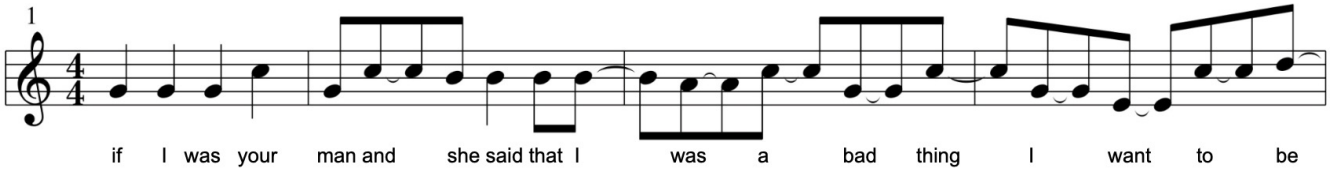
(e) Generated by GRU, the required emotion is negative and the seed lyric is 'I give you my'.



(f) Generated by GRU, the required emotion is negative and the seed lyric is 'but when told me'.



(g) Generated by Transformer, the required emotion is negative and the seed lyric is 'I have a dream'.



(h) Generated by Transformer, the required emotion is negative and the seed lyric is 'if I was your man'.

Fig. 7. Generated Samples of the ELMG system. A piece of seed lyric and the required emotion are given. 4 positive samples and 4 negative samples generated by different generators are selected.

TABLE XIII
ANSWERS OF QUESTIONS GIVEN BY 20 PARTICIPANTS, THESE QUESTIONS ARE MEASURED BY A FIVE POINT DISCRETE SCALE, IN WHICH 1 TO 5 CORRESPONDS TO "VERY BAD", "BAD", "OK", "GOOD" AND "VERY GOOD" RESPECTIVELY. THE "AVG" SHOWS THE AVERAGE SCORE. HIGHER IS BETTER.

Questions	Ground-truth						GRU						Transformer					
	1	2	3	4	5	Avg	1	2	3	4	5	Avg	1	2	3	4	5	Avg
How meaningful are the lyrics?	0	7	17	31	5	**3.6**	2	4	25	29	0	3.4	0	5	29	26	0	3.4
How sounds good are the melodies?	1	8	26	22	3	3.3	4	6	16	29	5	**3.4**	2	6	27	25	0	3.3
Are the lyrics and melody compatible?	0	7	23	25	5	3.5	0	4	20	32	4	3.6	1	4	17	30	8	**3.7**

system. The ELMG system has the potential to generate music with higher quality if a better dataset is given.

VI. CONCLUSION

In this paper, we construct a large-scale paired lyric-melody dataset with emotion labels and propose Emotional Lyric and Melody Generator (ELMG) system for emotion-conditioned music generation. Firstly, we find that dataset annotators trained on in-domain data are more reliable than models trained on out-of-domain data. Then, both GRU and Transformer based encoder-decoder network trained on our dataset successfully learned to compose lyric and melody. Next, emotional beam search (EBS) algorithm is designed to control the generation process by using a music emotion classifier, which let the generated music segments represent the specific given emotion. Finally, subjective and objective evaluations demonstrate that the EBS algorithm can bias the generation process to required emotions.

In addition, music generation with emotions is still unexplored well and a challenging problem in deep learning area. The new dataset created in this work only has single track in the melody and the emotion annotator only focus on the lyric. The quality of the dataset limits the effectiveness of our proposed model. Collect large-scale polyphonic music dataset with emotion labels is a valuable further work for us.

ACKNOWLEDGMENTS

This research is supported by A*STAR under its AME YIRG grant (Project No. A20E6c0101).

REFERENCES

[1] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *Journal of King Saud University-Computer and Information Sciences*, 2020.
[2] Y. Zhao, X. Xia, and R. Togneri, "Applications of deep learning to audio generation," *IEEE Circuits and Systems Magazine*, vol. 19, no. 4, pp. 19–38, 2019.
[3] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep learning techniques for music generation–a survey," *arXiv preprint arXiv:1709.01620*, 2017.
[4] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," *arXiv preprint arXiv:2011.06801*, 2020.
[5] https://colinraffel.com/projects/lmd/.
[6] https://www.reddit.com/r/datasets/.
[7] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[8] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4364–4373.
[9] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.
[10] J. Ens and P. Pasquier, "Mmm: Exploring conditional multi-track music generation with the transformer," *arXiv preprint arXiv:2008.06048*, 2020.
[11] M. Czyz and M. Kedziora, "Automated music generation using recurrent neural networks," in *International Conference on Dependability and Complex Systems*. Springer, 2021, pp. 22–31.
[12] N. Carroll, "Art and mood: Preliminary notes and conjectures," *The monist*, vol. 86, no. 4, pp. 521–555, 2003.
[13] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," *arXiv preprint arXiv:2103.06125*, 2021.
[14] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
[15] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, "Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," *arXiv preprint arXiv:2108.01374*, 2021.
[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[17] D. Edmonds and J. Sedoc, "Multi-emotion classification for song lyrics," in *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2021, pp. 221–235.
[18] A. Vijayakumar, M. Cogswell, R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra, "Diverse beam search for improved description of complex scenes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[19] W. Kool, H. Van Hoof, and M. Welling, "Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3499–3508.
[20] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1362–1371.
[21] A. Roberts, J. Engel, and D. Eck, "Hierarchical variational autoencoders for music," in *NIPS Workshop on Machine Learning for Creativity and Design*, vol. 3, 2017.
[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
[23] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.
[24] L. Ferreira, L. Lelis, and J. Whitehead, "Computer-generated music for tabletop role-playing games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, 2020, pp. 59–65.
[25] C.-F. Huang and C.-Y. Huang, "Emotion-based ai music generation system with cvae-gan," in *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*. IEEE, 2020, pp. 220–222.
[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
[27] H. Bao, S. Huang, F. Wei, L. Cui, Y. Wu, C. Tan, S. Piao, and M. Zhou, "Neural melody composition from lyrics," in *CCF Inter-*

national Conference on Natural Language Processing and Chinese Computing. Springer, 2019, pp. 499–511.
[28] Y. Yu, A. Srivastava, and S. Canales, "Conditional lstm-gan for melody generation from lyrics," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1, pp. 1–20, 2021.
[29] Y. Yu, F. Harscoët, S. Canales, G. Reddy, S. Tang, and J. Jiang, "Lyrics-conditioned neural melody generation," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 709–714.
[30] G. R. Madhumani, Y. Yu, F. Harscoët, S. Canales, and S. Tang, "Automatic neural lyrics and melody composition," *arXiv preprint arXiv:2011.06380*, 2020.
[31] Y. Yu, A. Srivastava, and R. R. Shah, "Conditional hybrid gan for sequence generation," *arXiv preprint arXiv:2009.08616*, 2020.
[32] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "Goemotions: A dataset of fine-grained emotions," *arXiv preprint arXiv:2005.00547*, 2020.
[33] E. Çano and M. Morisio, "Moodylyrics: A sentiment annotated lyrics dataset," in *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, 2017, pp. 118–124.
[34] R. Malheiro, R. Panda, P. Gomes, and R. P. Paiva, "Emotionally-relevant features for classification and regression of music lyrics," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 240–254, 2016.
[35] S. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 174–184.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.
[37] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.
[38] C. L. Krumhansl and E. J. Kessler, "Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys." *Psychological review*, vol. 89, no. 4, p. 334, 1982.
[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[40] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
[41] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, "Neural text generation with unlikelihood training," *arXiv preprint arXiv:1908.04319*, 2019.
[42] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," *arXiv preprint arXiv:1702.01806*, 2017.
[43] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, "Understanding and improving layer normalization," *arXiv preprint arXiv:1911.07013*, 2019.

Chunhui Bao received the B.S. degree in Computer Science and Technology from Sichuan University, Chengdu, China, in 2019. He is currently working toward the MSc degree in Information Systems with the School of Computing and Information Systems, Singapore Management University, Singapore. His research interests include deep learning, natural language processing, and music generation.

Qianru Sun is an Assistant Professor of Computer Science in the School of Computing and Information Systems, Singapore Management University, since Aug 2019. From 2018 to 2019, she was a Research Fellow at the National University of Singapore and the MPI for Informatics. From 2016 to 2018, she held the Lise Meitner Award Fellowship and worked at the MPI for Informatics. In 2016, she obtained her Ph.D. degree from Peking University. In 2014, she visited at the University of Tokyo. Her research interests are computer vision and machine learning that aim to develop efficient algorithms and systems for visual understanding.