Neural Melody Composition from Lyrics

Hangbo Bao[1], Shaohan Huang[2], Furu Wei[2], Lei Cui[2], Yu Wu[2], Chuanqi Tan[3], Songhao Piao[1], and Ming Zhou[2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
hangbobao@gmail.com,piaosh@hit.edu.cn
[2] Microsoft Research, Beijing, China
{shaohanh,fuwei,lecu,Wu.Yu,mingzhou}@microsoft.com
[3] Beihang University, Beijing, China
tanchuanqi@nlsde.buaa.edu.cn

Abstract. In this paper, we study a novel task that learns to compose music from natural language. Given the lyrics as input, we propose a melody composition model that generates lyrics-conditional melody as well as the exact alignment between the generated melody and the given lyrics simultaneously. More specifically, we develop the melody composition model based on the sequence-to-sequence framework. It consists of two neural encoders to encode the current lyrics and the context melody respectively, and a hierarchical decoder to jointly produce musical notes and the corresponding alignment. Experimental results on lyrics-melody pairs of 18,451 pop songs demonstrate the effectiveness of our proposed methods. In addition, we apply a singing voice synthesizer software to synthesize the "singing" of the lyrics and melodies for human evaluation. Results indicate that our generated melodies are more melodious and tuneful compared with the baseline method.

Keywords: Neural Melody Composition · Conditional Sequence Generation.

1 Introduction

We study the task of melody composition from lyrics, which consumes a piece of text as input and aims to compose the corresponding melody as well as the exact alignment between generated melody and the given lyrics. Specifically, the output consists of two sequences of musical notes and lyric syllables[4] with two constraints. First, each syllable in the lyrics at least corresponds to one musical note in the melody. Second, a syllable in the lyrics may correspond to a sequence of notes, which increases the difficulty of this task. Figure 1 shows a fragment of a Chinese song. For instance, the last Chinese character '恋' (love) aligns two notes 'C5' and 'A4' in the melody.

There are several existing research works on generating lyrics-conditional melody [1, 16, 8, 6]. These works usually treat the melody composition task as a classification or

[4] A syllable is a word or part of a word which contains a single vowel sound and that is pronounced as a unit. Chinese is a monosyllabic language which means words (Chinese characters) predominantly consist of a single syllable (https://en.wikipedia.org/wiki/Monosyllabic_language).

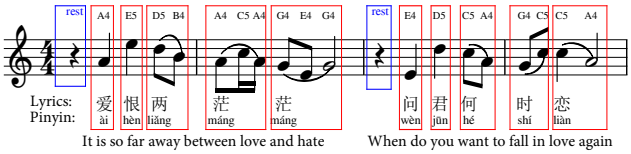


Fig. 1. A fragment of a Chinese song "Drunken Concubine (new version)". The blue rectangles indicate rests, some intervals of silence in a piece of melody. The red rectangles indicate the alignment between the lyrics and the melody, meaning a mapping from syllable of lyrics to musical notes. Pinyin indicates the syllables for each Chinese character.

sequence labeling problem. They first determine the number of musical notes by counting the syllables in the lyrics, and then predict the musical notes one after another by considering previously generated notes and corresponding lyrics. However, these works only consider the "one-to-one" alignment between the melody and lyrics. According to our statistics on 18,451 Chinese songs, 97.9% songs contains at least one syllable that corresponds to multiple musical notes (i.e. "one-to-many" alignment), thus the simplification may introduce bias into the task of melody composition.

In this paper, we propose a novel melody composition model which can generate melody from lyrics and well handle the "one-to-many" alignment between the generated melody and the given lyrics. For the given lyrics as input, we first divide the input lyrics into sentences and then use our model to compose each piece of melody from the sentences one by one. Finally, we merge these pieces to a complete melody for the given lyrics. More specifically, it consists of two encoders and one hierarchical decoder. The first encoder encodes the syllables in current lyrics into an array of hidden vectors with a bi-directional recurrent neural network (RNN) and the second encoder leverages an attention mechanism to convert the context melody into a dynamic context vector with a two-layer bi-directional RNN. In the decoder, we employ a three-layer RNN decoder to produce the musical notes and the alignment jointly, where the first two layers are to generate the pitch and duration of each musical note and the last layer is to predict a label for each generated musical note to indicate the alignment.

We collect 18,451 Chinese pop songs and generate the lyrics-melody pairs with precise syllable-note alignment to conduct experiments on our methods and baselines. Automatic evaluation results show that our model outperforms baseline methods on all the metrics. In addition, we leverage a singing voice synthesizer software to synthesize the "singing" of the lyrics and melodies and ask human annotators to manually judge the quality of the generated pop songs. The human evaluation results further indicate that the generated lyrics-conditional melodies from our method are more melodious and tuneful compared with the baseline methods.

The contributions of our work in this paper are summarized as follows.

- To the best of our knowledge, this paper is the first work to use end-to-end neural network model to compose melody from lyrics.

· We construct a large-scale lyrics-melody dataset with 18,451 Chinese pop songs and 644,472 lyrics-context-melody triples, so that the neural networks based approaches are possible for this task.
· Compared with traditional sequence-to-sequence models, our proposed method can generate the exact alignment as well as the "one-to-many" alignment between the melody and lyrics.
· The human evaluation verifies that the synthesized pop songs of the generated melody and input lyrics are melodious and meaningful.

2 Preliminary

Table 1. Notations used in this paper

Notations	Description
X	the sequence of syllables in given lyrics
x^j	the j-th syllable in X
M	the sequence of musical notes in context melody
m^i	the i-th musical note in M
m^i_{pit}, m^i_{dur}	the pitch and duration of m^i, respectively
Y	the sequence of musical notes in predicted melody
y^i	the i-th musical note in Y
$y_{<i}$	the previously predicted musical notes $\{y^1, ..., y^{i-1}\}$ in Y
$y^i_{pit}, y^i_{dur}, y^i_{lab}$	the pitch, duration and label of y^i, respectively
$Pitch$	the pitch sequence comprised of each y^i_{pit} in Y
$Duration$	the duration sequence comprised of each y^i_{dur} in Y
$Label$	the label sequence comprised of each y^i_{lab} in Y
h^j_{lrc}	the j-th hidden state in output of lyrics encoder
h^i_{con}	the i-th hidden state in output of context melody encoder
c^i	the dynamic context vector at time step i
c^i_{con}	the i-th melody context vector from context melody encoder
R	indicates the rest, specially

2.1 Concepts from Music Theory

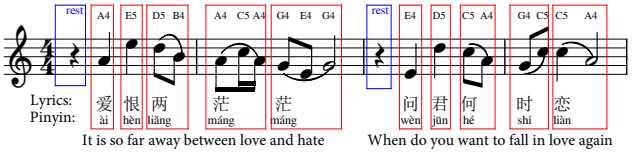
Melody can be regarded as an ordered sequence of many musical notes. The basic unit of melody is the musical note which mainly consists of two attributes: pitch and duration. The pitch is a perceptual property of sounds that allows their ordering on a frequency-related scale, or more commonly, the pitch is the quality that makes it possible to judge sounds as "higher" and "lower" in the sense associated with musical melodies. Therefore, we use a sequence of numbers to represent the pitch. For example, we represent 'C5' and 'Eb6' as 72 and 87 respectively based on the MIDI. [5]. A rest is an interval of silence in a piece of music and we use 'R' to represent it and treat it as a special pitch. Duration is a particular time interval to describe the length of time that the pitch or tone sounds[6], which is to judge how long or short a musical note lasts.

[5] https://newt.phys.unsw.edu.au/jw/notes.html
[6] https://en.wikipedia.org/wiki/Duration_(music)

2.2 Lyrics-Melody Parallel Corpus

Figure 2 shows an example of a lyrics-melody aligned pair with precise syllable-note alignment, where each Chinese character of the lyrics aligns with one or more notes in the melody.

An example of a sheet music:



Lyrics-melody aligned data:

Pinyin		ài		hèn		liǎng			máng				máng			wèn		jūn		hé		shí		liàn	
Lyrics		爱		恨		两			茫				茫			问		君		何		时		恋	
Pitch	R	A4	E5	D5	B4	A4	C5	A4	G4	E4	G4	R	E4	D5	C5	A4	G4	C5	C5	A4	
Duration	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	
Label	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0	1	0	1	0	1	

Fig. 2. An illustration for lyrics-melody aligned data.

The generated melody consists of three sequences $Pitch$, $Duration$ and $Label$ where the $Label$ sequence represents the alignment between melody and lyrics. We are able to rebuild the sheet music with them. $Pitch$ sequence represents the pitch of each musical note in melody and 'R' represents the rest in $Pitch$ sequence specifically. Similarly, $Duration$ sequence represents the duration of each musical note in melody. $Pitch$ and $Duration$ consist of a complete melody but do not include information on the alignment between the given lyrics and corresponding melody.

$Label$ contains the information of alignment. Each item of the $Label$ is labeled as one of $\{0, 1\}$ to indicate the alignment between the musical note and the corresponding syllable in the lyrics. To be specific, a musical note is assigned with label 1 that denotes it is a boundary of the musical note sub-sequence, which aligned to the corresponding syllable, otherwise it is assigned with label 0. We can split the musical notes into the n parts by label 1, where n is the number of syllables of the lyrics, and each part is a musical note sub-sequence. Then we can align the musical notes to their corresponding syllables sequentially. Additionally, we always align the rests to their latter syllables. For instance, we can observe that the second rest aligns to the Chinese character '问' (ask).

3 Approach

In this section, we present the end-to-end neural networks model, termed as **Songwriter**, to compose a melody which aligns exactly to the given input lyrics. Figure 3

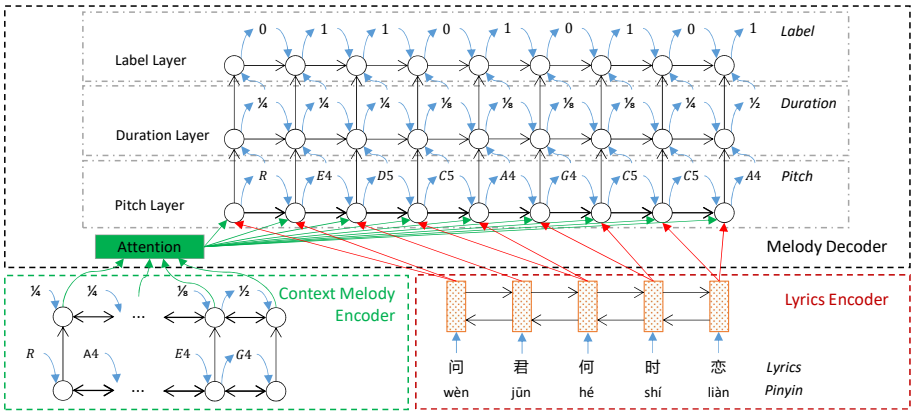


Fig. 3. An illustration of Songwriter. The lyrics encoder and context melody encoder encode the syllables of given lyrics and the context melody into two arrays of hidden vectors, respectively. For decoding the i-th musical note y^i, Songwriter uses attention mechanism to obtain a context vector c^i_{con} from the context melody encoder (green arrows) and counts how many label 1 has been produced in previously musical notes to obtain h^j_{con} to represent the current syllable corresponding to y^i from the lyrics encoder (red arrows) to melody decoder. In melody decoder, the pitch layer and duration layer first predict the pitch y^i_{pit} and duration y^i_{dur} of y^i, then the label layer predicts a label y^i_{lab} for y^i to indicate the alignment.

provides an illustration of Songwriter. Given lyrics as the input, we first divide the lyrics into sentences and then use Songwriter to compose each piece of the melody sentence by sentence.

3.1 Lyrics Encoder

We use a bi-directional RNN [15] built by two GRUs [4] to encode the syllables of lyrics which concatenates the syllable feature embedding and word embedding as input $X = \{x^1, ..., x^{|X|}\}$ to the GRU encoders:

$$\boldsymbol{h}^i_{lrc} = f_{\text{GRU}}(\boldsymbol{h}^{i-1}_{lrc}, x^i) \tag{1}$$

$$\boldsymbol{h}^i_{lrc} = f_{\text{GRU}}(\boldsymbol{h}^{i+1}_{lrc}, x^i) \tag{2}$$

$$h^i_{lrc} = \begin{bmatrix} \boldsymbol{h}^i_{lrc} \\ \boldsymbol{h}^i_{lrc} \end{bmatrix} \tag{3}$$

Then, the lyrics encoder outputs $\{h^1_{lrc}, ..., h^{|X|}_{lrc}\}$ to represent the information of each syllable in the lyrics.

3.2 Context Melody Encoder

We use the context melody encoder to encode the context melody $M = \{m^1, ..., m^{|M|}\}$. The encoder is a two-layer RNN that encodes pitch and duration of a musical note respectively at each time step. Each layer is a bi-directional RNN which is built by two

GRUs. For the first layer, we describe the forward directional GRU and the backward directional GRU at time step i as follows:

$$\boldsymbol{h}_{pit}^{i} = f_{\mathrm{GRU}}(\boldsymbol{h}_{pit}^{i-1}, m_{pit}^{i}) \tag{4}$$

$$\boldsymbol{h}_{pit}^{i} = f_{\mathrm{GRU}}(\boldsymbol{h}_{pit}^{i+1}, m_{pit}^{i}) \tag{5}$$

$$h_{pit}^{i} = \begin{bmatrix} \boldsymbol{h}_{pit}^{i} \\ \boldsymbol{h}_{pit}^{i} \end{bmatrix} \tag{6}$$

where m_{pit}^{i} is the pitch attribute of i-th note m^{i}. The bottom layer encodes the output of the first layer and the duration attribute of melody:

$$\boldsymbol{h}_{dur}^{i} = f_{\mathrm{GRU}}(\boldsymbol{h}_{dur}^{i-1}, m_{dur}^{i}, h_{pit}^{i}) \tag{7}$$

$$\boldsymbol{h}_{dur}^{i} = f_{\mathrm{GRU}}(\boldsymbol{h}_{dur}^{i+1}, m_{dur}^{i}, h_{pit}^{i}) \tag{8}$$

$$h_{dur}^{i} = \begin{bmatrix} \boldsymbol{h}_{dur}^{i} \\ \boldsymbol{h}_{dur}^{i} \end{bmatrix} \tag{9}$$

We concatenate the two output arrays of vectors to an array of vectors to represent the context melody sequence:

$$h_{con}^{i} = \begin{bmatrix} h_{pit}^{i} \\ h_{dur}^{i} \end{bmatrix} \tag{10}$$

3.3 Melody Decoder

The decoder predicts the next note y^{i} from all previously predicted notes $\{y^{1}, ..., y^{i-1}\}$ ($y_{<i}$, for short), the context musical notes $M = \{m^{1}, ..., m^{|M|}\}$ and the syllables $X = \{x^{1}, ..., x^{|X|}\}$ of given lyrics. We define the conditional probability when decoding i-th note as follows:

$$\arg\max P(y^{i}|y_{<i}, X, M) \tag{11}$$

To model the three attributes of y^{i}, where we use $\{y_{pit}^{i}, y_{dur}^{i}, y_{lab}^{i}\}$ to respectively represent the pitch, duration and label, we decompose Eq. (11) into Eq.(12):

$$\begin{aligned} P(y^{i}|y_{<i}, X, M) = & P(y_{pit}^{i}|y_{<i}, X, M) \cdot \\ & P(y_{dur}^{i}|y_{<i}, X, M, y_{pit}^{i}) \cdot \\ & P(y_{lab}^{i}|y_{<i}, X, M, y_{pit}^{i}, y_{dur}^{i}) \end{aligned} \tag{12}$$

We use a three-layer RNN as decoder to respectively decode the pitch, duration and label of a musical note at each time step. We define the conditional probabilities of each layer in the decoder:

$$P(y_{pit}^{i}|y_{<i}, X, M) = g_{p}(s_{pit}^{i}, c^{i}, y^{i-1}) \tag{13}$$

$$P(y_{dur}^{i}|y_{<i}, X, M, y_{pit}^{i}) = g_{d}(s_{dur}^{i}, c^{i}, y^{i-1}, y_{pit}^{i}) \tag{14}$$

$$\begin{aligned} P(y_{lab}^{i}|y_{<i}, X, M, y_{pit}^{i}, y_{dur}^{i}) = \\ g_{l}(s_{lab}^{i}, c^{i}, y^{i-1}, y_{pit}^{i}, y_{dur}^{i}) \end{aligned} \tag{15}$$

where $g_p(\cdot)\}$, $g_d(\cdot)$ and $g_l(\cdot)$ are nonlinear functions that output the probabilities of y^i_{pit}, y^i_{dur} and y^i_{lab} respectively. s^i_{pit}, s^i_{dur} and s^i_{lab} are respectively the corresponding hidden states of each layer. c^i is a dynamic context vector representing the M and X:

$$c^i = c^i_{con} + h^j_{lrc} \tag{16}$$

where c^i_{con} is a context vector from context melody encoder and h^j_{lrc} is one of output hidden vectors of lyrics encoder, which represent the x_j that should be aligned to the current predicting y^i. In particular, we set c^i_{con} as a zero vector if there is no context melody as input. From our representation method for lyrics-melody aligned pairs, it is not difficult to understand how to get the x^j that y^i should be aligned to:

$$j = \sum_{t=1}^{i-1} y^t_{lab} \tag{17}$$

c^i_{con} is recomputed at each step by alignment model [2] as follows:

$$c^i_{con} = \sum_{t=1}^{|M|} \alpha^{i,t} h^t_{con} \tag{18}$$

$$\alpha^{i,t} = \frac{exp(e^{i,t})}{\sum_{k=1}^{|M|} exp(e^{i,k})} \tag{19}$$

$$e_{i,k} = \mathbf{v_a}^\intercal tanh(\mathbf{W_a} s^{i-1} + \mathbf{U_a} h^k_{con}) \tag{20}$$

where $\mathbf{v_a}$, $\mathbf{W_a}$ and $\mathbf{U_a}$ are learnable parameters. Finally, we obtain the c^i and then employ the s^i_p, s^i_d, s^i_l and s^i as follows:

$$s^i_{pit} = f_{\mathrm{GRU}}(s^{i-1}_{pit}, c^{i-1}, y^{i-1}_{pit}, h^j_{lrc}) \tag{21}$$

$$s^i_{dur} = f_{\mathrm{GRU}}(s^{i-1}_{dur}, c^{i-1}, y^{i-1}_{dur}, y^i_{pit}, s^i_{pit}) \tag{22}$$

$$s^i_{lab} = f_{\mathrm{GRU}}(s^{i-1}_{lab}, c^{i-1}, y^{i-1}_{lab}, y^i_{pit}, y^i_{dur}, d^i_{dur}) \tag{23}$$

$$s^i = [s^{i\,\intercal}_p; s^{i\,\intercal}_d; s^{i\,\intercal}_l]^\intercal \tag{24}$$

3.4 Objective Function

Given a training dataset with n lyrics-context-melody triples $\mathcal{D} = \{X^{(i)}, M^{(i)}, Y^{(i)}\}_{i=1}^n$, where $X^{(i)} = \{x^{(i)j}\}_{j=1}^{|X^{(i)}|}$, $M^{(i)} = \{m^{(i)j}\}_{j=1}^{|M^{(i)}|}$ and $Y^{(i)} = \{y^{(i)j}\}_{j=1}^{|Y^{(i)}|}$. In addition, $\forall(i,j)$, $y^{(i)j} = (y^{(i)j}_{pit}, y^{(i)j}_{dur}, y^{(i)j}_{lab})$. Our training objective is to minimize the negative log likelihood loss $\mathcal{L}$ with respect to the learnable model parameter θ:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{|Y^{(i)}|} log\ P(y^{(i)j}_{pit}, y^{(i)j}_{dur}, y^{(i)j}_{lab} | \theta, X^{(i)}, M^{(i)}, y_{<j}) \tag{25}$$

where $y_{<j}$ is short for $\{y^1_{(i)}, ..., y^j_{(i)}\}$.

4 Experiments

4.1 Dataset

We crawled 18,451 Chinese pop songs, which include melodies with the duration over 800 hours in total, from an online Karaoke app. Then preprocess the dataset with rules as described in Zhu at al. [19] to guarantee the reliability of the melodies. For each song, we convert the melody to C major or A minor that can keep all melodies in the same tune and we set BPM (Beats Per Minute) to 60 to calculate the duration of each musical note in the melody. We further divide the lyrics into sentences with their corresponding musical notes as lyrics-melody pairs. Besides, we set a window size as 40 to the context melody and use the previously musical notes as the context melody for each lyrics-melody pair to make up lyrics-context-melody triples. Finally, we obtain 644,472 triples to conduct our experiments. We randomly choose 5% songs for validating, 5% songs for testing and the rest of them for training.

4.2 Baselines

As melody composition task can generally be regarded as a sequence labeling problem or a machine translation problem, we select two state-of-the-art models as baselines.

- **CRF** A modified sequence labeling model based on CRF [7] which contains two layers for predicting $Pitch$ and $Duration$, respectively. For "one-to-many" relationships, this model uses some special tags to represent a series of original tags. For instance, if a syllable aligns two notes 'C5' and 'A4', we use a tag 'C5A4' to represent them.
- **Seq2seq** A modified attention based sequence to sequence model which contains two encoders and one decoder. Compared with Songwriter, Seq2seq uses attention mechanism [2] to capture information on the given lyrics. Seq2seq may not guarantee the alignment between the generated melody and syllables in given lyrics. To avoid this problem, Seq2seq model stops predicting when the number of the label 1 in predicted musical notes is equal to the number of syllables in the given lyrics.

4.3 Implementation

For all the models used in this paper, the number of recurrent hidden units is set to 256. In the context melody encoder and melody decoder, we treat the $pitch$, $duration$, and $label$ as tokens and use word embedding to represent them with 128, 128, and 64 dimensions, respectively. In the lyrics encoder, we use GloVe [12] to pre-train a char-level word embedding with 256 dimensions on a large Chinese lyrics corpus and use Pinyin [7]as the syllable features with 128 dimensions.

We use Adam [5] with an initial learning rate of 0.001 and an exponential decay rate of 0.9999 as the optimizer to train our models with batch size as 64, and we use the cross entropy as the loss function.

[7] https://en.wikipedia.org/wiki/Pinyin

Table 2. Automatic evaluation results

	Teacher-forcing				Sampling
	PPL	P	R	F_1	BLEU
CRF	3.39	45.5	47.6	45.9	2.02
Seq2seq	2.21	70.9	72.0	70.8	3.96
Songwriter	**2.01**	**75.3**	**76.0**	**75.3**	**6.63**

4.4 Automatic evaluation

We use two modes to evaluate our model and baselines.

- **Teacher-forcing**: As in [13], models use the ground truth as input for predicting the next-step at each time step.
- **Sampling** Models predict the melody from given lyrics without any ground truth.

Metrics We use the F_1 score to the automatic evaluation from Roberts et al. [13]. Additionally, we select three automatic metrics for our evaluation as follows.

- **Perplexity (PPL)** This metric is a standard evaluation measure for language models and can measure how well a probability model predicts samples. Lower PPL score is better.
- **(weighted) Precision**, **Recall** and $\mathbf{F_1}$ [8] These metrics measure the performance of predicting the musical notes.
- **BLEU** This metric [11] is widely used in machine translation. Higher BLEU score is better.

Results The results of the automatic evaluation are shown in Table 2. We can see that our proposed method outperforms all models in all metrics. As Songwriter performs better than Seq2seq, it shows that the exact information of the syllables can enhance the quality of predicting the corresponding musical notes relative to attention mechanism in traditional Seq2seq models. In addition, the CRF model demostrates lower performance in all metrics. In CRF model, we use a special tag to represent multiple musical notes if a syllable aligns more than one musical note, which will produce a large number of different kinds of tags and result in the CRF model is difficult to learn from the sparse data.

4.5 Human evaluation

Similar to the text generation and dialog response generation [18, 14], it is challenging to accurately evaluate the quality of music composition results with automatic metrics. To this end, we invite 3 participants as human annotators to evaluate the generated melodies from our models and the ground truth melodies of human creations.

[8] We calculate these metrics by scikit-learn with the parameter average set as 'weighted': `http://scikit-learn.org/stable/modules/classes.html\#module-sklearn.metrics`

Table 3. Human evaluation results in blind-review mode

Model	Overall	Emotion	Rhythm
Seq2seq	3.28	3.52	2.66
Songwriter	**3.83**	**3.98**	**3.52**
Human	4.57	4.50	4.17

We randomly select 20 lyrics-melody pairs, the average duration of each melody approximately 30 seconds, from our testing set. For each selected pair, we prepare three melodies, ground truth of human creations and the generated results from Songwriter and Seq2seq. Then, we synthesized all melodies with the lyrics by NiaoNiao [9] using default settings for the generated songs and ground truth, which is to eliminate the influences of other factors of singing. As a result, we obtain 5 (annotators) $\times$ 3 (melodies) $\times$ 20 (lyrics) samples in total. The human annotations are conducted in a blind-review mode, which means that human annotators do not know the source of the melodies during the experiments.

Metrics We use the metrics from previous work on human evaluation for music composition as shown below. We also include an *emotion* score to measure the relationship between the generated melody and the given lyrics. The human annotators are asked to rate a score from 1 to 5 after listening to the songs. Larger scores indicate better quality in all the three metrics.

- **Emotion** Does the melody represent the emotion of the lyrics?
- **Rhythm** [19, 17] When listening to the melody, are the duration and pause of words natural?
- **Overall** [17] What is the overall score of the melody?

Results Table 3 shows the human evaluation results. According to the results, Songwriter outperforms Seq2seq in all metrics, which indicates its effectiveness over the Seq2seq baseline. On the "Rhythm" metrics, human annotators give significantly lower scores to Seq2seq than Songwriter, which shows that the generated melodies from Songwriter are more natural on the pause and duration of words than the ones generated by Seq2seq. The results further suggest that using the exact information of syllables is more effective than the soft attention mechanism in traditional Seq2seq models in the melody composition task. We can also observe from Table 3 that the gaps between the system generated melodies and the ones created by human are still large on all the three metrics. It remains an open challenge for future research to develop better algorithms and models to generate melodies with higher quality.

[9] A singing voice synthesizer software which can synthesize Chinese song, `http://www.dsoundsoft.com/product/niaoeditor/`

5 Related Work

A variety of music composition works have been done over the last decades. Most of the traditional methods compose music based on music theory and expert domain knowledge. Chan et al. [3] design rules from music theory to use music clips to stitch them together in a reasonable way. With the development of machine learning and the increase of public music data, data-driven methods such as Markov chains model [10] and graphic model [9] have been introduced to compose music.

Generating a lyrics-conditional melody is a subset of music composition but under more restrictions. Early works first determine the number of musical notes by counting the syllables in lyrics and then predict the musical notes one after another by considering previously generated notes and corresponding lyrics. Fukayama et al. [6] use dynamic programming to compute a melody from Japanese lyrics, the calculation needs three human well-designed constraints. Monteith et al. [8] propose a melody composition pipeline for given lyrics. For each given lyrics, it first generates hundreds of different possibilities for rhythms and pitches. Then it ranks these possibilities with a number of different metrics in order to select a final output. Scirea et al. [16] employ Hidden Markov Models (HMM) to generate rhythm based on the phonetics of the lyrics already written. Then a harmonical structure is generated, followed by generation of a melody matching the underlying harmony. Ackerman et al. [1] design a co-creative automatic songwriting system ALYSIA base on machine learning model using random forests, which analyzes the lyrics features to generate one note at a time for each syllable.

6 Conclusion and Future Work

In this paper, we propose a lyrics-conditional melody composition model which can generate melody and the exact alignment between the generated melody and the given lyrics. We develop the melody composition model under the encoder-decoder framework, which consists of two RNN encoders, lyrics encoder and context melody encoder, and a hierarchical RNN decoder. The lyrics encoder encodes the syllables of current lyrics into a sequence of hidden vectors. The context melody leverages an attention mechanism to encode the context melody into a dynamic context vector. In the decoder, it uses two layers to produce musical notes and another layer to produce alignment jointly. Experimental results on our dataset, which contains 18,451 Chinese pop songs, demonstrate our model outperforms baseline models. Furthermore, we leverage a singing voice synthesizer software to synthesize "singing" of the lyrics and generated melodies for human evaluation. Results indicate that our generated melodies are more melodious and tuneful. For future work, we plan to incorporate the emotion and the style of lyrics to compose the melody.

References

1. Ackerman, M., Loker, D.: Algorithmic songwriting with alysia. In: International Conference on Evolutionary and Biologically Inspired Music and Art. pp. 1–16. Springer (2017)

2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2014), http://arxiv.org/abs/1409.0473
3. Chan, M., Potter, J., Schubert, E.: Improving algorithmic music composition with machine learning. In: Proceedings of the 9th International Conference on Music Perception and Cognition, ICMPC (2006)
4. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1724–1734 (2014), http://aclweb.org/anthology/D/D14/D14-1179.pdf
5. Diederik P. Kingma, J.B.: Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR) (2015)
6. Fukayama, S., Nakatsuma, K., Sako, S., Nishimoto, T., Sagayama, S.: Automatic song composition from the lyrics exploiting prosody of the japanese language. In: Proc. 7th Sound and Music Computing Conference (SMC). pp. 299–302 (2010)
7. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
8. Monteith, K., Martinez, T.R., Ventura, D.: Automatic generation of melodic accompaniments for lyrics. In: ICCC. pp. 87–94 (2012)
9. Pachet, F., Papadopoulos, A., Roy, P.: Sampling variations of sequences for structured music generation. In: Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR'2017), Suzhou, China. pp. 167–173 (2017)
10. Pachet, F., Roy, P.: Markov constraints: steerable generation of markov sequences. Constraints **16**(2), 148–172 (2011)
11. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. pp. 311–318. Association for Computational Linguistics (2002)
12. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
13. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A hierarchical latent vector model for learning long-term structure in music. arXiv preprint arXiv:1803.05428 (2018)
14. Schatzmann, J., Georgila, K., Young, S.: Quantitative evaluation of user simulation techniques for spoken dialogue systems. In: 6th SIGdial Workshop on DISCOURSE and DIALOGUE (2005)
15. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. Trans. Sig. Proc. **45**(11), 2673–2681 (Nov 1997). https://doi.org/10.1109/78.650093, http://dx.doi.org/10.1109/78.650093
16. Scirea, M., Barros, G.A., Shaker, N., Togelius, J.: Smug: Scientific music generator. In: ICCC. pp. 204–211 (2015)
17. Watanabe, K., Matsubayashi, Y., Fukayama, S., Goto, M., Inui, K., Nakano, T.: A melody-conditioned lyrics language model. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). vol. 1, pp. 163–172 (2018)
18. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks. In: EMNLP. pp. 670–680 (2014)
19. Zhu, H., Liu, Q., Yuan, N.J., Qin, C., Li, J., Zhang, K., Zhou, G., Wei, F., Xu, Y., Chen, E.: Xiaoice band: A melody and arrangement generation framework for pop music. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2837–2846. ACM (2018)