

TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method

Zeqian Ju¹, Peiling Lu², Xu Tan^{*2}, Rui Wang², Chen Zhang³,
Songruoyao Wu³, Kejun Zhang³, Xiangyang Li¹, Tao Qin², Tie-Yan Liu²

¹ University of Science and Technology of China

² Microsoft Research Asia, ³ Zhejiang University, China

<https://github.com/microsoft/muzic>

Abstract

Lyric-to-melody generation is an important task in automatic songwriting. Previous lyric-to-melody generation systems usually adopt end-to-end models that directly generate melodies from lyrics, which suffer from several issues: 1) lack of paired lyric-melody training data; 2) lack of control on generated melodies. In this paper, we develop TeleMelody, a two-stage lyric-to-melody generation system with music template (e.g., tonality, chord progression, rhythm pattern, and cadence) to bridge the gap between lyrics and melodies (i.e., the system consists of a lyric-to-template module and a template-to-melody module). TeleMelody has two advantages. First, it is data efficient. The template-to-melody module is trained in a self-supervised way (i.e., the source template is extracted from the target melody) that does not need any lyric-melody paired data. The lyric-to-template module is made up of some rules and a lyric-to-rhythm model, which is trained with paired lyric-rhythm data that is easier to obtain than paired lyric-melody data. Second, it is controllable. The design of the template ensures that the generated melodies can be controlled by adjusting the musical elements in the template. Both subjective and objective experimental evaluations demonstrate that TeleMelody generates melodies with higher quality, better controllability, and less requirement on paired lyric-melody data than previous generation systems.

1 Introduction

Music is a universal natural language for communication. With the rapid development of artificial intelligence, automatic songwriting has drawn much attention from both academia and industry. Automatic songwriting covers many tasks, such as lyric generation (Malmi et al., 2016; Xue et al., 2021), melody generation (Wu et al., 2020; Choi et al.,

2016; Zhu et al., 2018), lyric-to-melody generation (Yu et al., 2021; Sheng et al., 2020; Bao et al., 2019; Lee et al., 2019), and melody-to-lyric generation (Sheng et al., 2020; Watanabe et al., 2018; Li et al., 2020). In this paper, we focus on lyric-to-melody generation, since it is one of the most important and common tasks in songwriting and is still under-explored. Recent years, deep learning techniques have been widely used to develop end-to-end lyric-to-melody systems (Yu et al., 2021; Sheng et al., 2020; Bao et al., 2019; Lee et al., 2019). However, these systems suffer from the following issues: 1) They require large amount of paired lyric-melody data to learn the correlation between syllables in lyrics and notes in melodies (Sheng et al., 2020). However, collecting lots of paired data is quite difficult and costly. Sheng et al. have attempted to alleviate the low-resource challenge by unsupervised pre-training on lyric-to-lyric and melody-to-melody models. However, the utilization of unpaired data helps on the understanding and generation of lyrics and melodies while has little effect on the correlation learning between lyrics and melodies. 2) They generate melodies directly from lyrics, which hinders end users to control musical elements (e.g. tonality and chord progression) over the generation. Without controllability, requirements from users may be ignored and the application scenarios are limited.

In this paper, we propose TeleMelody¹, a two-stage lyric-to-melody generation system with a carefully designed template as a bridge to connect lyrics and melodies. The template contains tonality, chord progression, rhythm pattern, and cadence. This designed template is effective because: 1) It is convenient to be extracted from melodies and predicted from lyrics and can successfully catch their characteristics; 2) It is easy to be manipulated by users on demand. Accordingly, we break down the lyric-to-melody task into a lyric-to-template mod-

^{*}This work was conducted at Microsoft. Corresponding author: Xu Tan, xuta@microsoft.com

¹Tele is from TEmplAtE.

ule and a template-to-melody module. This can reduce the task difficulty, improve data efficiency and achieve better controllability. The details are described as follows:

This two-stage framework can help reduce the difficulty of learning the correlation between lyrics and melodies. In the template-to-melody module, we train a template-to-melody model with templates extracted from melodies by rules. Generating melodies from templates is much easier than from lyrics, since the correlation between templates and melodies is much stronger than that between lyrics and melodies, and the paired template-melody data can be easily got from a self-supervised way. In the lyric-to-template module, rhythm pattern in template is obtained by a lyric-to-rhythm model, which is trained with paired lyric-rhythm data. This paired data is obtained by extracting the rhythm pattern from crawled lyric-audio data through audio processing tools, which is much easier to get than paired lyric-melody data. Cadence is inferred based on punctuation mappings. Chord progression and tonality in the template can be acquired with predefined musical knowledge. In this way, the two modules can rely on self-supervised learning or data mining on external lyric-audio data, which do not require any paired lyric-melody data and are more data-efficient than end-to-end models.

Moreover, benefiting from the template based framework, end users can control the generated melodies by changing the musical elements in templates. Besides, in the sequence-to-sequence based template-to-melody model, we use musical knowledge to guide the learning of attention alignments between the template tokens and the corresponding melody tokens, which can lead to better controllability.

The main contributions of this work are as follows:

- We propose TeleMelody, a two-stage lyric-to-melody system with a carefully designed template as the bridge. It decomposes lyric-to-melody generation into a lyric-to-template module and a template-to-melody module. This framework can help reduce the task difficulty and improve data efficiency.
- Chord progression, tonality, rhythm pattern and cadence designed in templates can help control basic musical elements and high-level

music structures. We introduce alignment regularization based on musical knowledge to ensure better controllability for generated melodies.

- Experimental results demonstrate that TeleMelody significantly outperforms previous end-to-end lyric-to-melody generation models in terms of both objective and subjective evaluations on generation quality, and is capable of better controlling the generated melodies.

2 Background

2.1 Lyric-to-Melody Generation

Considerable development in lyric-to-melody generation has been seen in recent years, from rule-based or statistical methods to deep learning methods. Rule-based or statistical methods usually need lots of manual designs based on domain knowledge in music, and hinder end users to control musical elements. [Monteith et al.](#) generate rhythm pattern based on rules, and construct an n-gram model to predict note pitch². [Sze et al.](#) and [Rabin](#) learn the lyric-note correlation by performing statistical method with limited paired data. In these works, the melody is generated with the control of zero or just one specific musical element. [Fukayama et al.](#) obtain the optimal pitch sequence by maximizing the conditional probability given chords, tonality, accompaniment bass, rhythm and pitch accent information, but the generated melodies may suffer from bad musical structure without repetition patterns. Meanwhile the algorithm cannot be directly applied to lyrics written in “stress accent” languages like English.

Recently, developing lyric-to-melody systems based on machine learning methods attracts lots of attentions. [Scirea et al.](#) allocate the same number of notes as the syllables in lyrics using Markov chain. [Ackerman and Loker](#) leverage random forest model to construct a rhythm model and a melody model separately on paired lyric-audio data. [Bao et al.](#) and [Yu et al.](#) use sequence-to-sequence models to generate melody from lyrics. However, deep learning methods usually require large amount of paired lyric-melody data for learning the correlation between lyrics and melodies. [Sheng et al.](#) attempt to address low-resource challenge by performing pre-training for lyric-to-lyric and melody-to-melody

²[https://en.wikipedia.org/wiki/Pitch_\(music\)](https://en.wikipedia.org/wiki/Pitch_(music))



(a) The melody, lyric, and chord progression.

	C	C	C	C	F	F	G	F	F	F	F	G	G	C	Chord
C major	0	1	2	3	0	1	2	0	1	2	3	0	1	2	Rhythm pattern
(Tonality)	No	No	No	No	No	No	Half	No	No	No	No	No	No	Authentic	Cadence

(b) The corresponding template.

Figure 1: The song “Twinkle Twinkle Little Star” in “C major” tonality.

modules, and incorporating supervised learning into the pre-training to learn a shared latent space between lyrics and melodies. But the challenge is not well addressed since the unpaired data has not been sufficiently utilized on correlation learning between lyrics and melodies. Moreover, these works do not consider controlling specific musical elements of generated melodies. In this paper, we propose TeleMelody, a two-stage template-based system, which consists of a lyric-to-template module and a template-to-melody module. The two-stage framework can help address the issues of limited paired data, and the designed template together with gularization in this framework is able to ensure better controllability over generated melodies.

2.2 Music Background Knowledge

In this subsection, we use the song “Twinkle Twinkle Little Star” in Figure 1a as an example to introduce musical elements in the template.

- Tonality³ is composed of a scale⁴ and a root note. For example, the tonality of the melody in Figure 1a is “C major”, since notes are ordered within pitches in major scale with the root pitch “C”.
- Chord progression⁵ is an ordered sequence of chords. As in Figure 1a, the chord progression is “I-IV-V-IV-V-I” (“C-F-G-F-G-C” in C major scale). Chord progression, interacting with melody, should create a sense of harmony when composing music.
- Rhythm⁶ refers to the pattern of occurrence of notes and rests. In Figure 1a, each note

³<https://en.wikipedia.org/wiki/Tonality>

⁴[https://en.wikipedia.org/wiki/Scale_\(music\)](https://en.wikipedia.org/wiki/Scale_(music))

⁵https://en.wikipedia.org/wiki/Chord_progression

⁶https://en.wikipedia.org/wiki/Rhythm#Composite_rhythm

is aligned with a syllable and notes in green boxes are in the same rhythm patterns.

- Cadence⁷ occurs at the end of phrase and gives a sense of ending in melody, and is often aligned with punctuation marks in lyrics. In Figure 1a, we assign the half cadence⁷ and the authentic cadence⁷ to the comma and the period respectively.

3 Methodology

Figure 2a describes the lyric-to-melody generation system architecture connecting a lyric-to-template module and a template-to-melody module with a template. In this section, we introduce each component (i.e., template, lyric-to-template module, and template-to-melody module) in detail.

3.1 Template

In this subsection, we introduce template in respects of definition, design principles, contained musical elements, and the connection with the generated melodies.

The template is a well-designed sequence of musical elements that can capture the common attributes of lyrics and melodies. With the connection of this template, we decompose lyric-to-melody generation into a lyric-to-template module and a template-to-melody module as shown in Figure 2a.

Concerning task characteristics, we propose following high-level principles for template design: 1) Templates can be extracted directly from melodies so that a template-to-melody model can be trained in a self-supervised way. 2) Templates obtained from lyrics should be in accordance with those extracted from melodies in order to bridge lyrics and melodies in inference. 3) Compared with hidden representations in end-to-end models, templates should be more easily manipulated on demand to achieve better controllability.

Based on above principles, the designed template consists of tonality, chord progression, rhythm pattern, and cadence. The template representation is shown in Figure 2c: we use one token at the start of the sequence to represent the tonality and three consecutive tokens (chord, rhythm pattern, cadence) to represent musical elements of each note. Figure 1b provides the example of template corresponds to the melody in Figure 1a.

The influence of musical elements in templates on generated melodies are described as follows

⁷<https://en.wikipedia.org/wiki/Cadence>

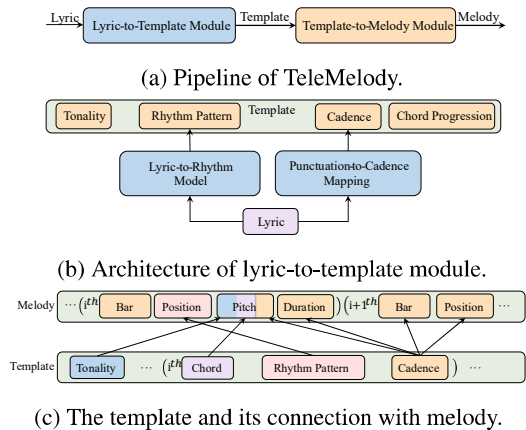


Figure 2: Architecture of TeleMelody.

(shown in Figure 2c): 1) Tonality can control pitch distribution. 2) Chord can influence the harmony of generated melodies. 3) Rhythm pattern can constrain note position and control high-level musical structure with repetitive patterns. 4) Cadence can guarantee the accordance between punctuation in lyrics and note onset intervals in melodies.

3.2 Template-to-Melody Module

The template-to-melody module is to generate melodies from given templates with an encoder-attention-decoder Transformer model in a self-supervised way. In this subsection, we first introduce how to extract musical elements in templates from melodies for model training and then introduce alignment regularization, which can leverage musical knowledge to improve model controllability.

Template Extraction Method We introduce the method for extracting templates (i.e., tonality, chord progression, rhythm pattern, cadence) from melodies: 1) Tonality can be inferred according to the note pitch distribution of a whole melody, following Liang et al.. 2) Chord progression can be inferred based on note pitch distribution through the Viterbi algorithm proposed by Magenta⁸. 3) Rhythm pattern can be inferred based on position information of the note. 4) Cadence can be inferred by rules based on note pitch, onset interval and duration. We classify it in three classes: “no cadence”, “authentic cadence”, and “half cadence”. Detailed rules are described in Appendix D.

Alignment Regularization With the introduction of the template, we can control melody genera-

⁸https://github.com/magenta/note-seq/blob/master/note_seq/chord_inference.py

tion by adjusting musical elements in the template. To further increase model controllability, we introduce musical knowledge to the template-to-melody model through well-designed alignment regularization (Garg et al., 2019) during training. Each designed alignment is imposed on the encoder-decoder attention. This musical knowledge based design can provide the guidance for the model to learn the interpretable alignments between templates and melodies, which can help for better controllability.

We denote m_k as the k^{th} note information token in melody sequence, and t_j as the j^{th} musical element token in template sequence. \hat{w} denotes a 0-1 matrix such that $\hat{w}_{k,j} = 1$ if m_k is aligned with t_j . We simply normalize the rows of matrix \hat{w} to get a matrix w . We expect encoder-decoder attention weight $A_{k,j}$ between m_k and t_j to be closer to $w_{k,j}$ defined as:

$$w_{k,j} = \begin{cases} \frac{1}{T} & \text{if } m_k \text{ is aligned to } t_j, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where T is the number of tokens in the template sequence that m_k is aligned to.

We denote J and K as the number of tokens in the source and target sentence respectively. The alignment regularization term is

$$L_{attn} = \frac{1}{K \times J} \sum_{k=1}^K \sum_{j=1}^J w_{k,j} \log A_{k,j} \quad (2)$$

The overall loss is

$$L = L_{nll} + \lambda_{attn} L_{attn} \quad (3)$$

where L_{nll} is the negative log likelihood loss and λ_{attn} is a hyperparameter.

We give an example in Figure 2c to illustrate the alignment between the consecutive i^{th} and $i+1^{th}$ note in the melody, and the corresponding musical elements in the template. The note information is consisted of bar index, position in a bar, pitch and duration. The alignments mentioned in Equation (1) are designed as follows:

- We add an alignment between tonality and every note pitch in the melody, since tonality controls the pitch distribution of the entire melody.
- We add an alignment between the chord of i^{th} note in the template and pitch of i^{th} note in the melody, since chord influences the note pitch.

- We add an alignment between the rhythm pattern of i^{th} note in the template and the position of i^{th} note in the melody, since rhythm pattern determines the note position.
- We add an alignment between the cadence of i^{th} note in the template with the duration and pitch of i^{th} note in the melody for their close relationship. Besides, we add alignments between cadence of i^{th} note with the bar index and position of $i+1^{th}$ note, since onset intervals between adjacent notes can help distinguish “no cadence” from others.

3.3 Lyric-to-Template Module

In this subsection, we describe the lyric-to-template module, which generates musical elements (i.e., tonality, chord progression, rhythm pattern, and cadence) in templates from lyrics. Tonality and chord progression are weakly correlated with lyrics and can be manipulated on demand. Therefore, we focus on generating rhythm pattern and cadence in the following paragraphs.

Lyric-to-Rhythm Model We introduce a transformer based lyric-to-rhythm model to predict rhythm patterns in the template with given lyrics in an auto-regressive way. To collect adequate lyric-rhythm data for training, we extracted the paired data from crawled lyric-audio data with audio processing tools. The detailed pipeline is similar to Xue et al. and is described in Appendix A.

Punctuation-Cadence Mapping Since punctuation marks in lyrics are closely related to cadences in melodies, we design musical knowledge based mapping as follows: 1) We align “authentic cadence” and “half cadence” with period and comma in lyrics respectively for indicating the end of the sentence. 2) We align the “no cadence” label with each syllable in lyrics since it is a voiced part.

4 Experimental Settings

4.1 Dataset

We conduct experiments on both English (EN) and Chinese (ZH) lyric-to-melody generation tasks to evaluate our model. For lyric-to-template module, we collect paired lyric-rhythm data (9,761 samples in English and 74,328 samples in Chinese) following the procedure in Appendix A. As for template-to-melody module, we obtain and process 45,129 midi data from a widely used dataset in

music generation, LMD-matched MIDI dataset⁹ (Raffel, 2016), since the musical elements in designed template do not have much correlation to languages. The majority of the dataset is Pop/Rock, following with Electronic and Country (Ferraro and Lemström, 2018). We provide detailed melody pre-processing in Appendix B. Statistics of lyric-template and template-melody dataset are shown respectively in Table 4 and Table 5 in Appendix C. The code and models can be found at our project¹⁰.

4.2 System Configurations

Both two Transformer models in lyric-to-template and template-to-melody module use the settings of 4 encoder layers, 4 decoder layers, 256 hidden size and 4 attention heads. The dropout is 0.2 in lyric-to-template model and 0.0005 in template-to-melody model. We use Adam optimizer with the learning rate of 0.0005 in both models. The alignment regularization weight λ_{attn} in template-to-melody model is set as 0.05. To ensure the diversity of generated melodies, we use stochastic sampling inference following Huang et al.. The temperature and top- k parameters are set as 0.5 and 2 in lyric-to-rhythm generation, and as 0.5 and as 10 in template-to-melody generation.

4.3 Evaluation Metrics

Objective Evaluation Following Sheng et al., we consider two metrics to measure the similarity of the generated and the ground-truth melodies: Similarity of pitch and duration distribution (PD and DD) and Melody distance (MD). Besides, we use accuracy of tonality, chord, rhythm pattern, and cadence (TA, CA, RA, AA) to measure the consistency between the generated melody and musical elements in the template. The more consistent the generated melody is with the template, the more controllable the model is. For ground-truth melodies, TA, RA, AA are 100%, while CA is less than 100% since introducing notes outside chord are encouraged to avoid monotony in songwriting. Therefore, CA scores is better if it is closer to that of the ground-truth melodies. Accordingly, we consider that the closer TA, CA, RA, and AA are to the ground-truth melodies, the more controllable the model is. The definitions of these metrics are described in Appendix F.

⁹<https://colinraffel.com/projects/lmd>

¹⁰<https://github.com/microsoft/muzic>

Subjective Evaluation We invite 10 participants (including 7 amateurs and 3 professionals) as human annotators to evaluate 10 songs in each language. We require each participant to rate properties of the melodies in a five-point scale, from 1 (Poor) to 5 (Perfect). The whole evaluation is conducted in a blind-review mode. Referring to previous works (Sheng et al., 2020; Zhu et al., 2018; Watanabe et al., 2018), we use following subjective metrics to evaluate the generated melodies: 1) Harmony: Is the melody itself harmonious? 2) Rhythm : Is the rhythm sounds natural and suitable for lyrics? 3) Structure: Does the melody consist of repetitive and impressive segments? 4) Quality: What is the overall quality of the melody?

5 Experimental Results

In this section, we first compare TeleMelody with baselines to demonstrate its effectiveness. Then, we show the analysis of TeleMelody in the aspects of controllability and data efficiency. Audio samples of the generated melodies are available via this link¹¹ and also in Supplementary Materials as described in Appendix G.

5.1 Main Results

We compare the performance of TeleMelody with two baselines: 1) SongMASS (Sheng et al., 2020), the state-of-the-art system that deals with low-resource scenario by end-to-end unsupervised lyric-to-lyric and melody-to-melody pre-training; 2) Transformer baseline, a Transformer model directly trained with paired lyric-melody data. Our TeleMelody (including lyric-to-rhythm and template-to-melody models) has similar number of model parameters with that of SongMASS and Transformer baseline for fair comparison. For English, the two baselines use 8,000 paired lyric-melody data (Yu et al., 2021), and SongMASS additionally uses 362,237 unpaired lyrics and 176,581 unpaired melodies for pre-training. For Chinese, the two baselines use 18,000 paired lyric-melody data (Bao et al., 2019), and SongMASS additionally uses 228,000 unpaired lyrics and 283,000 unpaired melodies crawled from the Web.

As shown in Table 1, TeleMelody significantly outperforms Transformer baseline on all the objective metrics (Improvement on EN: 19.45% in PD, 5.02% in DD, and 0.29 in MD; Improvement on ZH: 38.36% in PD, 13.45% in DD and 3.95

in MD). Compared with SongMASS, TeleMelody also performs better on all the objective metrics (Improvement on EN: 9.54% in PD, 4.18% in DD, and 0.21 in MD; Improvement on ZH: 23.40% in PD, 1.39% in DD and 0.49 in MD). Meanwhile, for all the subjective metrics, TeleMelody is better than both the Transformer baseline and SongMASS. Specifically for the quality, TeleMelody outperforms Transformer baseline by 1.49 in EN and 1.7 in ZH, and outperforms SongMASS by 1.10 in EN and 1.43 in ZH.

The results show that an end-to-end Transformer model has poor performance, since available paired lyric-melody data is limited. In SongMASS, the lyric-to-lyric and melody-to-melody unsupervised pre-training can effectively improve the performance of end-to-end models, but it is still insufficient since the unlabeled data is not effectively utilized to improve the correlation learning between lyrics and melodies. TeleMelody performs the best, since it successfully reduces the difficulty and effectively utilizes the unpaired melodies in the two-stage framework. Besides, the improvements in Table 1 are consistent with the intuition in designing the template: 1) tonality and chord progression in the template can control note pitch and thus help improve PD and harmony; 2) rhythm pattern and cadence in the template can control both onset and duration of notes, and thus help improve DD and rhythm; 3) the repetitive patterns in the template can help improve structure.

5.2 Method Analyses

In this subsection, we verify the effect of the designed template and the alignment regularization on controllability and verify data efficiency by testing TeleMelody with varying paired lyric-rhythm training data.

Controllability Study As shown in Table 2, TeleMelody with template is quite high in RA and AA, and close to the ground truth in CA, which indicates that the melody generated by TeleMelody is highly consistent with the rhythm, chord, and cadence in the template. Meanwhile, TeleMelody can also control the tonality with a good TA accuracy with the template. Moreover, the proposed alignment regularization (AR) can further improve the controllability (with closer TA, CA, RA, and AA to the ground truth).

To show the effect of the alignment regularization intuitively, we further visualize the average

¹¹<https://ai-music.github.io/telemelody>

	Objective			Subjective			
	PD(%) \uparrow	DD(%) \uparrow	MD \downarrow	Harmony \uparrow	Rhythm \uparrow	Structure \uparrow	Quality \uparrow
(EN) Transformer Baseline	24.41	47.04	2.69	2.0 (± 0.18)	2.1 (± 0.19)	1.8 (± 0.16)	1.8 (± 0.16)
(EN) SongMASS	34.32	47.88	2.61	2.4 (± 0.20)	2.3 (± 0.20)	2.3 (± 0.20)	2.2 (± 0.19)
(EN) TeleMelody	43.86	52.06	2.40	3.2 (± 0.24)	3.4 (± 0.19)	3.3 (± 0.21)	3.3 (± 0.20)
(ZH) Transformer Baseline	11.40	38.03	6.75	1.5 (± 0.15)	2.0 (± 0.22)	1.5 (± 0.14)	1.5 (± 0.13)
(ZH) SongMASS	26.36	50.09	3.29	1.8 (± 0.19)	2.2 (± 0.20)	1.8 (± 0.16)	1.7 (± 0.17)
(ZH) TeleMelody	49.76	51.48	2.80	3.0 (± 0.19)	3.5 (± 0.19)	3.3 (± 0.19)	3.2 (± 0.19)

Table 1: Objective and subjective evaluation (with 95% confidence interval) of TeleMelody and the baseline systems.

Metric	TA(%)	CA(%)	RA(%)	AA(%)
Ground Truth	100.00	62.10	100.00	100.00
Template	75.64	64.11	99.88	99.94
Template+AR	77.41	62.62	99.99	99.98

Table 2: Results on model controllability. The closer that TA, CA, RA, AA are to Ground Truth, the better the controllability is.

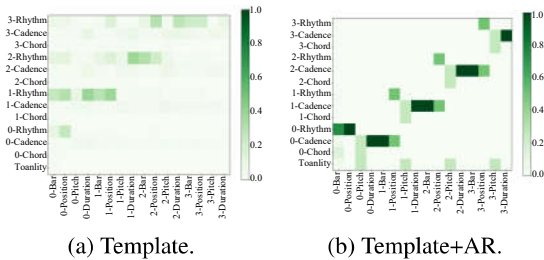


Figure 3: Alignment visualization. We denote x-coordinate “ i -feature” as the feature of the i^{th} note in the melody sequence, and y-coordinate “ i -element” as the musical element of i^{th} note in the template sequence.

encoder-decoder attention weights of all heads in the last layer. As shown in Figure 3, after adding alignment regularization (template + AR), the related elements in the template and the generated melody are clearly aligned, according to the alignment rules introduced in Section 3.2.

In addition, we conduct a case study to illustrate controllability and how the elements in the template affect the generated melody. The basic melody is shown in Figure 4a. We evaluate the control performance from the following aspects:

- Tonality determines pitch distributions in the melody. To verify the control of tonality, we adjust the tonality with fixed chord progression(VI-IV-V-I) in the template. As shown in Figure 4b, when we change the tonality in the template from “C major” to “A minor”, pitch distribution changes and the pitch

of the ending note changes from tonic pitch of “C major” to tonic pitch of “A minor”.

- For each note, a chord is provided in the template, which affects the pitch of the note. As shown in Figure 4c, when we change the chords of the notes in the first and the second bar, the pitches are changed correspondingly.
- Rhythm affects the onset position of the note. For example, in Figure 4, when we use the same rhythm for the first and third bars as labeled by green or blue boxes in each melody, the onset positions of the notes in the two bars are the same.
- Cadence affects the note onset intervals and note pitch at the end of phrases in the generated melody. As shown from pink boxes in Figure 4, note onset intervals at the end of each sentence are larger than other places from melody when cadence of note before comma and period are labeled as “half cadence” or an “authentic cadence”. And the note pitch from the orange box is the tonic pitch when “authentic cadence” is assigned to period.

Data Efficiency Study TeleMelody is data-efficient: 1) In template-to-melody module, we train a model with extracted templates from melodies in a self-supervised way. 2) In lyric-to-template module, only the lyric-to-rhythm model requires paired training data, which is easier to obtain than human-labeled paired lyric-melody data.

Since lyric-rhythm data is the only paired data we use, we test the lyric-to-rhythm model with different kinds of lyric-rhythm data to demonstrate the data-efficiency performance on TeleMelody:

1. **human-labeled data:** lyric-rhythm data obtained from the human-labeled paired lyric-

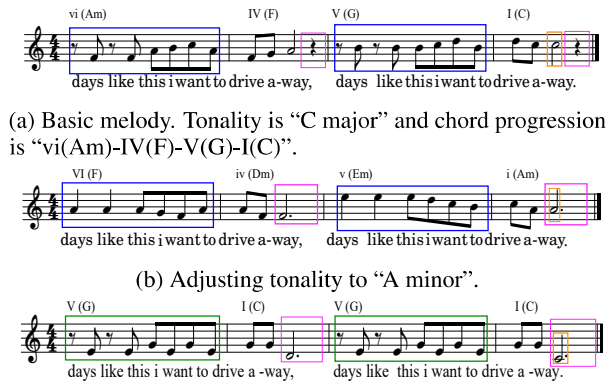


Figure 4: Case study on template adjustment. Similar bars are labeled by blue boxes while repetitive bars are labeled by green boxes.

melody data utilized by baselines described in Section 5.1.

2. **100% crawled data:** lyric-rhythm data obtained from crawled lyric-audio data described in Section 4.1, which is of comparable size with human-labeled data in 1.
3. **50% crawled data:** the same as 2 except only 50% of the crawled lyric-audio data is used.
4. **w.o. data:** no paired lyric-rhythm data is used, since we replace the lyric-to-rhythm model with hand-craft rules. The details of these hand-craft rules are described in Appendix E.

It is shown in Table 3 that:

- Comparing TeleMelody that uses crawled data with TeleMelody that uses human-labeled data, there is no distinct drop in performance, which shows the data efficiency of TeleMelody in that it can achieve good performance without any human-labeled data.
- Comparing TeleMelody that uses 100% of the crawled data with TeleMelody that uses only 50% of the crawled data, the performance only declines slightly, while still outperforms SongMASS on all the metrics.
- To further demonstrate the efficiency on the lyric-to-rhythm model, we replace the lyric-to-rhythm model with hand-craft rules for extracting rhythm pattern from lyrics, so that no paired data is used. The performance of TeleMelody without the proposed lyric-rhythm model significantly degrades, which

	PD(%) \uparrow	DD(%) \uparrow	MD \downarrow
(EN) TeleMelody			
• human-labeled data	46.73	52.95	2.36
• 100% crawled data	43.86	52.06	2.40
• 50% crawled data	41.42	48.59	2.40
• w.o. data	35.60	47.93	2.55
(EN) SongMASS	34.32	47.88	2.61
(ZH) TeleMelody			
• human-labeled data	47.29	54.76	2.24
• 100% crawled data	49.76	51.48	2.80
• 50% crawled data	46.25	50.14	3.12
• w.o. data	30.22	33.84	4.95
(ZH) SongMASS	26.36	50.09	3.29

Table 3: Objective evaluations of systems with varying lyric-rhythm training data. The “human-labeled data” is obtained from the human-labeled paired lyric-melody data utilized by baselines described in Section 5.1, while the “crawled data” is obtained from crawled singing-lyric data described in Section 4.1. We denote “w.o. data” as replacing the model with hand-craft rules described in Appendix E.

demonstrates the advantage of the lyric-to-rhythm model in TeleMelody.

- Comparing with SongMASS that uses human-labeled lyric-melody data, all the above settings of TeleMelody achieves better performance on all the objective metrics in English. These promising results further illustrate the data efficiency of our proposed TeleMelody.

6 Conclusion

In this paper, we proposed TeleMelody, a two-stage lyric-to-melody generation system with music template (e.g., tonality, chord progression, rhythm pattern, and cadence) to bridge the gap between lyrics and melodies. TeleMelody is data efficient and can be controlled by end users by adjusting the musical elements. Both subjective and objective experimental evaluations demonstrate that TeleMelody can generate melodies with higher quality than previous lyric-to-melody generation systems. Experimental studies also verify the data efficiency and controllability of TeleMelody. In future work, we will extend our proposed two-stage framework to other music generation tasks (e.g., melody-to-lyric generation and melody-to-accompaniment generation).

7 Limitations

With the use of the template to connect lyrics with melodies, TeleMelody can outperform the state of

the art, together with less requirement on paired training data. Nevertheless, we still find that the generated melodies are relatively less diverse than human creations. Future work can further improve this in the following aspects: 1) The large-scale model is demonstrated to be effective for learning the underlying data distribution, which can help improve the diversity of generated melodies, but it requires large amount of paired data. Thus, more works on large-scale data annotation and data collection techniques (e.g. accurate audio-to-MIDI conversion) are expected to be developed. 2) Experimental results demonstrate the introduction of the template containing well-designed musical elements in TeleMelody helps address the issue of low-resource data. More musical elements in the template are expected to be extended to other scenarios (e.g. controlling of music style and emotion).

References

- Margareta Ackerman and David Loker. 2017. [Algorithmic songwriting with alysia](#). In *Computational Intelligence in Music, Sound, Art and Design*, pages 1–16, Cham. Springer International Publishing.
- Hangbo Bao, Shaohan Huang, Furu Wei, Lei Cui, Yu Wu, Chuanqi Tan, Songhao Piao, and Ming Zhou. 2019. [Neural melody composition from lyrics](#). In *NLPCC 2019*.
- Keunwoo Choi, György Fazekas, and Mark B. Sandler. 2016. [Text-based LSTM networks for automatic music composition](#). *CoRR*, abs/1604.05358.
- Andres Ferraro and Kjell Lemström. 2018. [On large-scale genre classification in symbolically encoded music by automatic identification of repeating patterns](#). In *Proceedings of the 5th International Conference on Digital Libraries for Musicology, DLfM '18*, page 34–37, New York, NY, USA. Association for Computing Machinery.
- Satoru Fukayama, Kei Nakatsuma, Shinji Sako, Takuya Nishimoto, and Shigeki Sagayama. 2010. [Automatic song composition from the lyrics exploiting prosody of the japanese language](#). *Sound and Music Computing Conference (SMC)*, pages 299–302.
- Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. [Jointly learning to align and translate with transformer models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.
- Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinulescu, and Douglas Eck. 2018. [Music transformer](#).
- Hsin-Pei Lee, Jih-Sheng Fang, and Wei-Yun Ma. 2019. [iComposer: An automatic songwriting system for Chinese popular music](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 84–88, Minneapolis, Minnesota. Association for Computational Linguistics.
- Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. [Rigid formats controlled text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online. Association for Computational Linguistics.
- Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. 2020. [PiRhDy: Learning Pitch-, Rhythm-, and Dynamics-Aware Embeddings for Symbolic Music](#), page 574–582. Association for Computing Machinery, New York, NY, USA.
- Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. [Dopelearning: A computational approach to rap lyrics generation](#). In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 195–204, New York, NY, USA. Association for Computing Machinery.
- Kristine Monteith, Tony R Martinez, and Dan Ventura. 2012. [Automatic generation of melodic accompaniments for lyrics](#). In *ICCC*, pages 87–94.
- Michael O. Rabin. 1963. [Probabilistic automata](#). *Information and Control*, 6(3):230–245.
- Colin Raffel. 2016. [Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching](#). Columbia University.
- Marco Scirea, Gabriella A.B. Barros, Noor Shaker, and Julian Togelius. 2015. [Smug: Scientific music generator](#). In *Proceedings of the 6th International Conference on Computational Creativity, ICCO 2015*, Proceedings of the 6th International Conference on Computational Creativity, ICCO 2015, pages 204–211. Brigham Young University.
- Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. 2020. [Songmass: Automatic song writing with pre-training and alignment constraint](#). In *AAAI 2021*.
- Raymond Ka Wai Sze, Raymond Chi-Wing Wong, and Cheng Long. 2013. [T-music: A melody composer based on frequent pattern mining](#). In *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE '13, page 1332–1335, USA. IEEE Computer Society.

Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. 2018. [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)*, pages 163–172, New Orleans, Louisiana. Association for Computational Linguistics.

Jian Wu, Xiaoguang Liu, Xiaolin Hu, and Jun Zhu. 2020. [Popmnet: Generating structured pop music melodies using neural networks](#). *Artificial Intelligence*, 286:103303.

Langqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. 2021. [Deeprapper: Neural rap generation with rhyme and rhythm modeling](#). In *ACL-IJCNLP 2021*.

Yi Yu, Abhishek Srivastava, and Simon Canales. 2021. [Conditional lstm-gan for melody generation from lyrics](#). *ACM Trans. Multimedia Comput. Commun. Appl.*, 17(1).

Hongyuan Zhu, Qi Liu, Nicholas Jing Yuan, Chuan Qin, Jiawei Li, Kun Zhang, Guang Zhou, Furu Wei, Yuanchun Xu, and Enhong Chen. 2018. [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, KDD '18*, page 2837–2846, New York, NY, USA. Association for Computing Machinery.

A Pipeline of Collecting Lyric-Rhythm Data

We crawl paired lyric-singing audio data from the Web and then utilize spleeter¹², a public music separation tool, to separate the vocal from the accompaniment part. The timestamps of lyrics and the tempo of melodies are two necessary information during collection of rhythm information. To extract lyric timestamps, we first split the singing audio into sentence-level segments with crawled start and end timestamp. We convert lyrics into sequences of phonemes via Phonemizer¹³ and then obtain the vocal-lyric alignment in phoneme level with an audio alignment tool, Montreal Forced Aligner¹⁴. Based on these phoneme-level vocal-lyric alignments, we can obtain the corresponding timestamp of each lyric. To extract tempo information, we perform a direct estimation from the accompaniment audio with an audio information

¹²<https://github.com/deezer/spleeter>

¹³<https://github.com/bootphon/phonemizer>

¹⁴<https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner>

retrieval tool, librosa¹⁵. Finally, with tempo information and timestamp of each lyric, we can infer beat-level onset, that is, the corresponding rhythm of each lyric.

Considering that syllables in lyrics correspond to notes in melodies, we encode each syllable as a lyric token. For Chinese, each character has only one syllable. And for English, we divide each English word into a number of syllables, and represent each syllable as a lyric token.

B Processing Details of Melody and Template

Melody In this paper, we only consider the melody with a constant tempo and a time signature of 4/4¹⁶. Each note is represented by four consecutive tokens (bar, position, pitch and duration). We use 256 tokens to represent different bars and 16 tokens to represent different positions in a bar with granularity of 1/16 note. We use 128 tokens to represent pitch values following the MIDI format. We use 16 tokens to represent duration values ranging from a 1/16 note to a whole note.

We perform pre-processing to get high-quality melody data as follows: First, we extract the melody from the track¹⁷ with at least 50 notes that has the highest average pitch among all the tracks, and then delete polyphonic notes. Second, we normalize the tonality to “C major” or “A minor”, and normalize the pitches to fit the range of vocals. Finally, we filter the empty bars in each sample. After these steps, we can obtain the processed target melody dataset. To construct paired template-melody dataset, we utilize our proposed knowledge-based rules to extract the corresponding template from the target melody.

Template In this paper, template contains musical elements including tonality, chord progression, rhythm pattern, and cadence. We only consider “C major” and “A minor” tonalities for simplicity, since other tonalities can be transposed to these two tonalities based on their scales. Chord consists of a root note and a chord quality. We consider 12 chord roots ($C, C\sharp, D, D\sharp, E, F, F\sharp, G, G\sharp, A, A\sharp, B$) and 7 chord qualities (major, minor, diminished, aug-

¹⁵<https://github.com/librosa/librosa>

¹⁶4/4 denotes that that each beat is a 1/4 note and each bar has 4 beats.

¹⁷https://en.wikipedia.org/wiki/Multitrack_recording

mented, major⁷, minor⁷, half diminished), resulting in 84 possible chords in total. We use 4 tokens ranging from 0 to 3 to represent rhythm patterns, that is, beat-level onset position in a bar. For cadence, we consider “half cadence”, “authentic cadence” and “no cadence”, which is aligned with comma, period and other syllables in lyrics respectively.

C Dataset Statistics

Dataset statistics are shown in Table 4 and 5.

# of data samples	9,761
Average # of words per song	26.02
Average # of syllables per song	31.15
Average # of punctuation marks per song	3.79
(a) English	
# of data samples	74,328
Average # of words per song	78.77
Average # of punctuation marks per song	9.30
(b) Chinese	

Table 4: Statistics of lyric-rhythm dataset.

# of data samples	157,702
Average # of notes per song	28.76
Average # of bars per song	5.93

Table 5: Statistics of template-melody dataset.

D Cadence Extraction Rule

In template-to-melody module, we extract cadence from melodies in training stage through cadence extraction rule as follows: 1) “no cadence” is assigned to the note when the note duration is short (e.g., less than 1 beat¹⁸) or the onset interval between the current note with the next note is small (e.g., less than 1.5 beats). 2) “authentic cadence” is assigned to the note when the note is the root note¹⁹ of tonic²⁰ chord, or the inferred chord of this note is tonic chord. There is also a probability of p (e.g., 0.3) for labeling this note with “authentic cadence” when the note is other notes in tonic chord rather than root note, and the onset interval is large (e.g., more than 2 beats). 3) “half cadence” is assigned to

¹⁸[https://en.wikipedia.org/wiki/Beat_\(music\)](https://en.wikipedia.org/wiki/Beat_(music))

¹⁹[https://en.wikipedia.org/wiki/Root_\(chord\)](https://en.wikipedia.org/wiki/Root_(chord))

²⁰[https://en.wikipedia.org/wiki/Tonic_\(music\)](https://en.wikipedia.org/wiki/Tonic_(music))

notes outside the above two situations. In lyric-to-template module, we directly obtain cadence from lyrics in inference stage through punctuation-to-cadence mapping.

Therefore, a question may arise: is there a gap in cadence between training and inference? To answer this question, we explore the statistics of cadences in template-melody dataset. The results are shown in Table 6:

- Notes labeled with “no cadence”, which are aligned with syllables in lyrics, have short duration and onset interval. Notes labeled with “half cadence”, which are aligned with commas in lyrics, have $4.91\times$ longer duration and $4.22\times$ longer onset interval than those labeled with “no cadence”. Notes labeled with “authentic cadence”, which are aligned with periods in lyrics, have $5.94\times$ longer duration and $5.11\times$ longer onset interval than those labeled with “no cadence”. This is consistent with musical knowledge, since punctuation marks are usually aligned with pauses.
- The average number of “no cadence” is $5.86\times$ greater than the average number of “half cadence” and “authentic cadence” combined. This ratio is similar to the ratio of syllables to punctuation marks in lyrics, as shown in Table 4.

Cadence	Average # per song	Duration	Onset Interval
No	24.57	1.74	2.29
Half	2.53	8.54	9.66
Authentic	1.66	10.34	11.70

Table 6: Statistics of notes labeled with different cadences in template-to-melody dataset. Both duration and onset interval are quantified to 1/16 note.

E Lyric-to-Rhythm Hand-Craft Rules

We design several lyric-to-rhythm rules to demonstrate that TeleMelody can be applied in the scenario without any paired data. Specifically, for English lyrics, a note is corresponding to a syllable, and we generate the rhythm patterns syllable by syllable, where the onset interval between a note and its previous note is 2 beats if its corresponding syllable is the start of a sentence, is 1 beat if its corresponding syllable is the start of a word but not the start of a sentence, and is 0.5 beat otherwise. For Chinese lyrics, a note is corresponding

to a character, and we generate the rhythm patterns character by character, where the onset interval between a note and its previous note is 2 beats if its corresponding character is the start of a sentence, and is 1 beat otherwise.

F Definitions of TA, RA, CA, and AA

We evaluate model controllability with accuracy of tonality, rhythm pattern, chord, and cadence (TA, RA, CA, AA), that is, the proportion of notes in consistent with given template. Specifically, a melody is in consistent with tonality if the inferred tonality is the same as the template tonality; a note is in consistent with rhythm pattern if its position is in accord with given rhythm pattern information; a note is in consistent with chord if its pitch is within the chord; a note is in consistent with cadence if both its duration and its onset interval between this note and the next note comply with the extraction rules described in Section 3.2.

G Samples of Generated Melodies

Samples of generated melodies are available via this link²¹ and are available in “sample.zip”. We utilize XStudioSinger²² tool to synthesise singing and ChordPulse²³ for accompaniment for better listening experience.

²¹<https://ai-music.github.io/telemelody>

²²<https://singer.xiaoice.com>

²³<http://www.chordpulse.com>