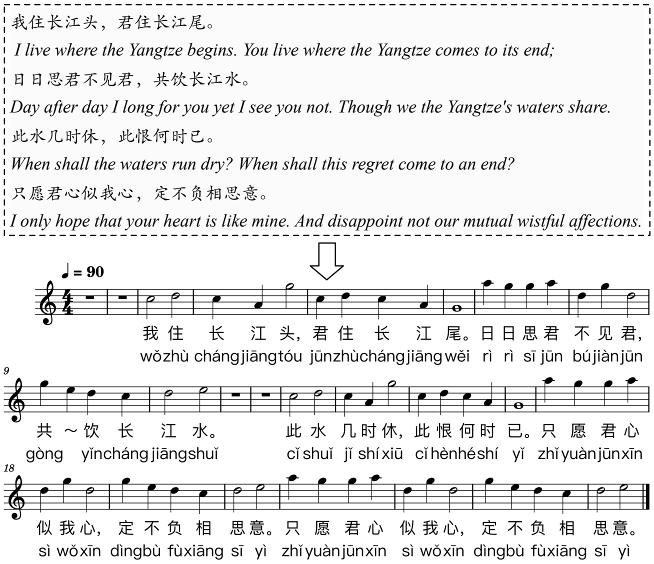
LingGe: An Automatic Ancient Chinese Poem-to-Song Generation System

Yong Shan , Jinchao Zhang , Huiying Ren , Yao Qiu , Jie Zhou
WeChat AI, Tencent
{yeongshan, dayerzhang, feliciaren, yasinqiu, withtomzhou}@tencent.com

Abstract

This paper presents a novel system, named *LingGe* (“伶歌” in Chinese), to generate songs for ancient Chinese poems automatically. LingGe takes the poem as the lyric, composes music conditioned on the lyric, and finally outputs a full song including the singing and the accompaniment. It consists of four modules: rhythm recognition, melody generation, accompaniment generation, and audio synthesis. Firstly, the rhythm recognition module analyzes the song structure and rhythm according to the poem. Secondly, the melody generation module assembles the rhythm into the template and then generates the melody. Thirdly, the accompaniment generation module predicts the accompaniment in harmony with the melody. Finally, the audio synthesis module generates singing and accompaniment audio and then mixes them to obtain songs. The results show that LingGe can generate high-quality and expressive songs for ancient Chinese poems, both in harmony and rhythm.



我住长江头，君住长江尾。
I live where the Yangtze begins. You live where the Yangtze comes to its end;
日日思君不见君，共饮长江水。
Day after day I long for you yet I see you not. Though we the Yangtze's waters share.
此水几时休，此恨何时已。
When shall the waters run dry? When shall this regret come to an end?
只愿君心似我心，定不负相思意。
I only hope that your heart is like mine. And disappoint not our mutual wistful affections.

♩= 90
我 住 长 江 头，君 住 长 江 尾。日 日 思 君 不 见 君，
wǒzhù chángjiāngtóu jūnzhùchángjiāngwěi rì rì sī jūn bújiànjūn
9
共 ～饮 长 江 水。 此 水 几 时 休，此 恨 何 时 已。只 愿 君 心
gòng yǐncháng jiāngshuǐ cǐ shuǐ jǐ shíxiū cǐ hènhéshí yǐ zhǐyuànjūnxīn
18
似 我 心， 定 不 负 相 思 意。只 愿 君 心 似 我 心， 定 不 负 相 思 意。
sì wǒxīn dìngbù fùxiāng sī yì zhǐyuànjūnxīn sì wǒxīn dìngbù fùxiāng sī yì

Figure 1: A song example generated by our system based on the ancient Chinese poem. We demonstrate the accompaniment and singing audio on the demo webpage[3].

1 Introduction

Ancient Chinese poem[1] is an important intangible cultural heritage of China and an artistic carrier of thought, culture, spirit, and emotion with more than thousands of years in history. Unlike modern Chinese poems, it is typically written in classical Chinese and follows specific rules, forms, tones, and rhyme schemes. In ancient years, these poems could be sung into songs with corresponding melodies. However, most of these melodies have been lost for historical reasons. Therefore, it is significant and attractive to recreate melodies and songs for ancient Chinese poems. Technically, it can be formulated as the lyric-to-melody generation task.

In recent years, with the rapid development of artificial intelligence, automatic lyric-to-melody generation has achieved remarkable progress. Previous studies usually adopt end-to-end models to directly generate melodies from lyrics, which requires a large amount of paired lyric-melody data to sufficiently learn the strict lyric-melody feature alignments [Bao *et al.*, 2018; Yu and Canales, 2019; Lee *et al.*, 2019; Sheng *et al.*, 2021; Qian *et al.*, 2022]. However, the paired data is hard and expensive to collect, which limits the performance of these methods. To alleviate this problem, other studies decompose the lyric-to-melody task into two generation stages [Ju *et al.*, 2021], or retrieve pre-generated music pieces from the database according to the key lyric features [Lv *et al.*, 2022]. However, these methods suffer from the data noise produced by auto-labeling tools or neglect the inner rhythm of lyrics, which reduces the melody quality. Moreover, previous systems fail to generate rhythms[2] in harmony with the ancient Chinese poem written in classical Chinese. Briefly, the ancient Chinese poem-to-song generation still needs to be explored.

In this paper, we propose an automatic ancient Chinese poem-to-song generation system called *LingGe*[3]. LingGe takes the poem as the lyric and generates songs conditioned on the lyric. It comprises four modules: rhythm recogni-

[1] https://en.wikipedia.org/wiki/Classical_Chinese_poetry

[2] https://en.wikipedia.org/wiki/Rhythm#Composite_rhythm

[3] Demo webpage: https://boya-music.github.io/lingge

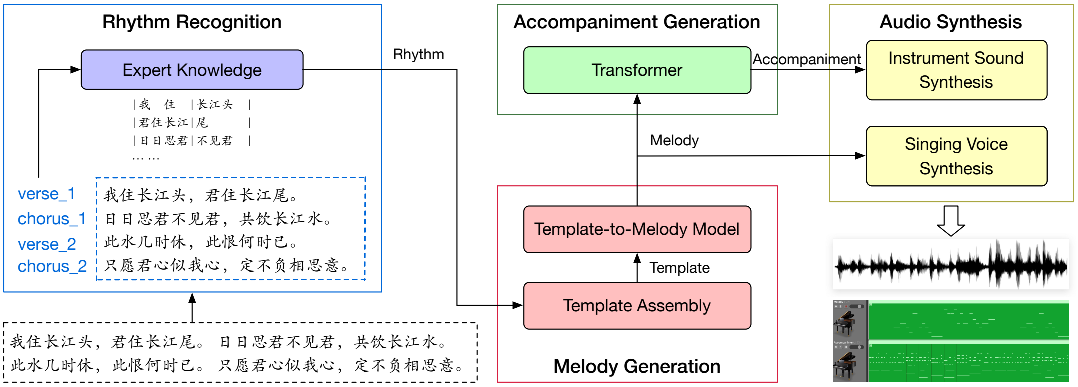


Figure 2: The architecture of LingGe consists of four modules: rhythm recognition, melody generation, accompaniment generation, and audio synthesis. Given an ancient Chinese poem, LingGe composes music and generates a song including the singing and the accompaniment.

tion, melody generation, accompaniment generation, and audio synthesis. In the rhythm recognition module, we introduce prior expert knowledge to analyze the song structure and obtain a consistent rhythm with the lyric. In the melody generation module, we assemble the rhythm into the template and employ a template-to-melody model [Ju *et al.*, 2021] to generate melodies. Unlike previous systems, we trained the template-to-melody model on our collected dataset including 28931 melodies with high-quality chord[4] annotations to enhance the quality of generated melodies. In the accompaniment generation module, we integrate a Transformer model [Vaswani *et al.*, 2017] to predict harmonious accompaniment, which further enhances the performance. Finally, we adopt the audio synthesis module to synthesize singing and accompaniment audio and then mix them to obtain songs. Specifically, we train a singing voice synthesis model based on DiffSinger [Liu *et al.*, 2021] to generate the singing audio. The results show that LingGe can generate high-quality songs for ancient Chinese poems both in harmony and rhythm.

2 Dataset Construction

For training the melody generation module, we crawled melodies with high-quality chord annotations from online websites. Then, we applied the following normalization steps and finally obtained the dataset with 28931 melodies: (1) Keep only MIDI files with a constant tempo and 4/4[5] time signature. (2) Remove empty bars. (3) Normalize the tonality[6] to "C major" or "A minor" since other tonalities can be transposed to these two tonalities based on their scales[7]. (4) Normalize the chord annotations into "C major" or "A minor" tonalities. Next, we extracted the template from the melody and represented the paired template-melody with symbolic tokens. For the melody, we represent them as note sequences where each note is symbolized by four consecutive tokens (bar, position, pitch, and duration). For the template, we will describe the details of template extraction in Section 3.2.

For training the accompaniment generation module, we adopted POP909 dataset [Wang *et al.*, 2020], which contains 909 songs with detailed annotations including the melody track, the chord progression, and the accompaniment track with broken chords style. We combine the melody notes and the chord progression as the input, where each note is symbolized by five consecutive tokens (bar, position, pitch, duration, and chord). Similarly, we represent the accompaniment notes as the output, where each note is symbolized by four consecutive tokens (bar, position, pitch, and duration).

For training the audio synthesis module, we collected a 6-hour singing voice synthesis dataset including 100 songs performed by a qualified female vocalist. All the audio files are recorded in a recording studio and sampled at 24kHz with 16-bit quantization. All the songs are phonetically labeled with phoneme boundaries, syllable boundaries, and pitch manually. To improve the training efficiency, we decompose the recorded songs into short utterances according to the rest and the lyric semantic information.

3 Poem-to-Song Generation

As shown in Figure 2, LingGe contains four modules: (1) a rhythm recognition module that analyzes the song structure and the rhythm according to the poem, (2) a melody generation module that assembles the rhythm into the template and then generates melodies, (3) an accompaniment generation module that predicts the accompaniment in harmony with the melody, (4) an audio synthesis module that generates singing audio and accompaniment audio, and mixes them into songs.

3.1 Rhythm Recognition

Ancient Chinese poem is typically written in classical Chinese following specific metrics and forms, which has a good structure and is neat in antithesis. Therefore, we carefully design the rule-based rhythm recognition module to guarantee the harmony of rhythm. First, given the lyric (i.e., an ancient Chinese poem) with specific metrics, we parse the song structure into four sections: `verse_1`, `chorus_1`, `verse_2`, and `chorus_2`. Due to the restriction of metrics, similar sections (e.g., all verses, all choruses) are symmetrical in the number of phrases and syllables. Thus, we also share the

[4] https://en.wikipedia.org/wiki/Chord_(music)
[5] 4/4 denotes that each beat is a 1/4 note and each bar has 4 beats.
[6] https://en.wikipedia.org/wiki/Tonality
[7] https://en.wikipedia.org/wiki/Scale_(music)

rhythm and the melody between similar sections to keep consistent. Then, we analyze the rhythm of each section phrase by phrase based on expert knowledge. Specifically, we develop several possible rhythm patterns for phrases of each length in advance. For example, a 5-syllable phrase can be segmented into two bars with a rhythm of "|2|3|" or "|4|1|". For a new song, we sample rhythm patterns according to the phrase length and then arrange all sections sequentially to obtain the complete rhythm.

3.2 Melody Generation

To alleviate the data scarcity problem in poem-to-song generation, we first extract templates from our melody dataset, then train a template-to-melody model, and finally assemble the lyric-based rhythm into the template to generate melodies.

Template extraction. We extracted musical elements including tonality, chord progression[8], rhythm pattern, pitch[9] pattern, and cadence[10] from melodies, and then combine them into a template. Unlike TeleMelody [Ju *et al.*, 2021], we introduce the novel pitch pattern to constrain the melody trend. Specifically, we detect several melodic movement features ("conjunct", "disjunct", "auxiliary notes", "repeated notes", "chord notes", and "others") and integrate these features into the template. Tonality and chord progression are annotated in the original dataset. Rhythm patterns can be inferred based on the position information of notes in melodies. Cadence can be inferred based on note duration.

Training. We adopt the Transformer [Vaswani *et al.*, 2017] architecture as the template-to-melody model including a 6-layer encoder and a 6-layer decoder, and then train it on our collected paired template-melody dataset. It predicts the melody in an auto-regressive manner.

Inference. We assemble the lyric-based rhythm into the template. Specifically, the tonality is set to "C major". The cadence is inferred by the punctuation of the lyric. The chord progression is sampled from some popular chord progressions. To improve the melody aesthetic, we collect a database where pitch patterns are collected according to the syllable length from our training set in advance. For each phrase, we sample a pitch pattern from the database and then share the same pitch pattern with other phrases with equal syllable lengths. Thus, phrases with equal syllable length will have a similar pitch trend, which brings repetitions and variations into the generated melody. Finally, we polish the generated melody with several heuristic arrangement rules.

3.3 Accompaniment Generation

To further improve the performance, we train a Transformer model to generate harmonious accompaniment. After the melody is generated, it takes the melody notes and the chord progression as the input and then decodes the accompaniment notes with broken chords style. To make the song more natural, we add `intro`, `bridge` and `outro` sections and generate corresponding accompaniment notes.

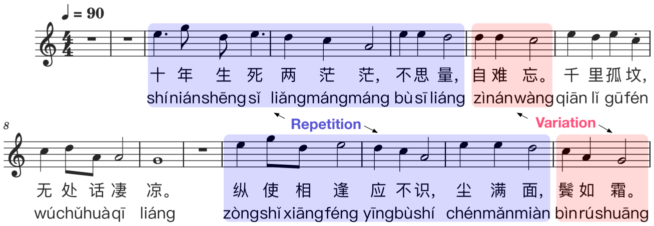

♩ = 90
十 年 生 死 两 茫 茫，不思 量，自难 忘。千 里孤 坟，
shíniánshēngsǐ liǎngmángmáng bùsīliáng zìnánwàng qiān lǐ gūfén
8
无 处 话 凄 凉。纵 使 相 逢 应 不识，尘 满 面，鬓 如 霜。
wúchǔhuàqī liáng zòngshǐxiāngféng yīngbùshí chénmǎnmiàn bìnrúshuāng
Repetition
Variation

Figure 3: A song example generated by LingGe shows musical composition skills such as repetitions and variations.

3.4 Audio Synthesis

To further enrich the song, we synthesize the singing and accompaniment audio. For the singing, we train a singing voice synthesis model to generate expressive singing voices based on lyrics and melodies. It consists of an acoustic model that produces the mel-spectrogram conditioned on the music score and a vocoder that reconstructs the singing voice from the mel-spectrogram. We use the diffusion-based DiffSinger [Liu *et al.*, 2021] as the acoustic model, which can be efficiently trained by optimizing ELBO without adversarial feedback, and generates realistic mel-spectrograms strongly matching the ground truth distribution. Meanwhile, we use the HiFi-GAN model [Kong *et al.*, 2020] as the vocoder, which can efficiently synthesize high-quality speech audio. Specifically, we pretrained the HiFi-GAN vocoder on the OpenSinger dataset [Huang *et al.*, 2021], which is a large-scale, multi-singer Chinese singing voice dataset. To ensure the generalization, we further fine-tuned the pre-trained vocoder on our singing dataset. For the accompaniment, we employ the fluidsynth [Newmarch and Newmarch, 2017] tool to generate audio with piano timbre. Finally, we mix the singing and accompaniment audio into the complete song for the poem.

4 Results

Figure 3 shows a composition result generated by LingGe. More audible results are available on the demo webpage[3]. It demonstrates the following points: (1) The structure and rhythm of the song are consistent with the poem. (2) The melody is harmonious, with musical composition skills (e.g., repetitions, variations), suitable for the lyrics. (3) The accompaniment is in harmony with the melody. (4) The singing voice is natural and expressive.

5 Conclusion

In this paper, we develop a novel system, LingGe, to challenge the task of generating songs for ancient Chinese poems. LingGe consists of four modules: rhythm recognition, melody generation, accompaniment generation, and audio synthesis. The rhythm recognition module analyzes the song structure and rhythm. The melody generation module assembles the template and generates the melody. The accompaniment generation module predicts the accompaniment from the melody. Finally, the audio synthesis module generates the singing and accompaniment audio and then mixes them into a complete song. The results show that LingGe can generate high-quality and expressive songs from ancient Chinese poems automatically.

[8] https://en.wikipedia.org/wiki/Chord_progression

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

[10] https://en.wikipedia.org/wiki/Cadence

References

[Bao *et al.*, 2018] Hangbo Bao, Shaohan Huang, Furu Wei, Lei Cui, Yuehua Wu, Chuanqi Tan, Songhao Piao, and M. Zhou. Neural melody composition from lyrics. In *Natural Language Processing and Chinese Computing*, 2018.

[Huang *et al.*, 2021] Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[Ju *et al.*, 2021] Zeqian Ju, Peiling Lu, Xu Tan, Rui Wang, Chen Zhang, Songruoyao Wu, Ke jun Zhang, Xiang-Yang Li, Tao Qin, and Tie-Yan Liu. Telemelody: Lyric-to-melody generation with a template-based two-stage method. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

[Kong *et al.*, 2020] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

[Lee *et al.*, 2019] Hsin-Pei Lee, Jhih-Sheng Fang, and Wei-Yun Ma. icomposer: An automatic songwriting system for chinese popular music. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[Liu *et al.*, 2021] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *AAAI Conference on Artificial Intelligence*, 2021.

[Lv *et al.*, 2022] Ang Lv, Xu Tan, Tao Qin, Tie-Yan Liu, and Rui Yan. Re-creation of creations: A new paradigm for lyric-to-melody generation. *ArXiv*, abs/2208.05697, 2022.

[Newmarch and Newmarch, 2017] Jan Newmarch and Jan Newmarch. Fluidsynth. *Linux Sound Programming*, pages 351–353, 2017.

[Qian *et al.*, 2022] Tao Qian, Jiatong Shi, Shuai Guo, Peter Wu, and Qin Jin. Training strategies for automatic song writing: A unified framework perspective. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4738–4742, 2022.

[Sheng *et al.*, 2021] Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. Songmass: Automatic song writing with pre-training and alignment constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13798–13805, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2020] Ziyu Wang, K. Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia. Pop909: A pop-song dataset for music arrangement generation. In *International Society for Music Information Retrieval Conference*, 2020.

[Yu and Canales, 2019] Yi Yu and Simon Canales. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17:1 – 20, 2019.