Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Special Track on AI, the Arts and Creativity

Re-creation of Creations: A New Paradigm for Lyric-to-Melody Generation

Ang Lv[1] , **Xu Tan**[2*] , **Tao Qin**[2] , **Tie-Yan Liu**[2] and **Rui Yan**[1,3*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Microsoft
[3]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education
{anglv, ruiyan}@ruc.edu.cn, {xuta, taoqin, tyliu}@microsoft.com

Abstract

Current lyric-to-melody generation methods struggle with the lack of paired lyric-melody data to train, and the lack of adherence to composition guidelines, resulting in melodies that do not sound human-composed. To address these issues, we propose a novel paradigm called Re-creation of Creations (ROC) that combines the strengths of both rule-based and neural-based methods. ROC consists of a two-stage generation-retrieval pipeline: the creation and re-creation stages. In the creation stage, we train a melody language model using melody data to generate high-quality music fragments, which are stored in a database indexed by key features. In the re-creation stage, users provide lyrics and a preferred chord progression, and ROC infers melody features for each lyric sentence. By querying the database, we obtain relevant melody fragments that satisfy composition guidelines, and these candidates are filtered, re-ranked, and concatenated based on the guidelines and the melody language model scores. ROC offers two main advantages: it does not require paired lyric-melody data, and it incorporates commonly used composition guidelines, resulting in music that sounds more human-composed with better controllability. Both objective and subjective evaluation results on English and Chinese lyrics show the efficacy of ROC.

1 Introduction

In recent years, deep learning has led to significant advances in automatic songwriting, including lyric generation [Malmi *et al.*, 2016; Xue *et al.*, 2021], melody generation [Wu *et al.*, 2020; Zhu *et al.*, 2018], lyric-to-melody generation [Bao *et al.*, 2018; Yu *et al.*, 2021; Sheng *et al.*, 2020; Ju *et al.*, 2021], and melody-to-lyric generation [Ma *et al.*, 2021; Xue *et al.*, 2021; Li *et al.*, 2020]. This paper focuses on one of the fundamental tasks, lyric-to-melody generation, specifically in the pop music genre. While existing methods generate decent melodies, they are limited by two major issues.

First, the mapping from lyric to melody is challenging to learn due to weak correlation between the lyric content and the melody (e.g., a melody can be accompanied by different lyrics as long as the lyric syllables align with notes). A large amount of aligned training data is required, which is rare. Second, the melodies generated by learned mapping patterns do not sound human-composed because most neural-based methods overlook commonly used composition guidelines.

To overcome these issues, we propose a new paradigm for lyric-to-melody generation called Re-creation of Creations (ROC), which combines the merits of both rule-based and neural-based methods. ROC has a generation-retrieval pipeline consisting of two stages: creation and re-creation. In the creation stage, we train a melody language model using melody data to generate high-quality music fragments. We store these fragments in a database indexed by extracted key features, such as tonality, chord, and structure (i.e., whether the melody fragment in a chorus or a verse), which are used for retrieval in the next stage. In the re-creation stage, users provide lyrcis and a preferred chord progression to ROC. Then, ROC infer the melody features associated with each lyric sentence. By querying the database using these melody features, we obtain relevant melody fragments which satisfy the composition guidelines. These candidates are then filtered, re-ranked, and concatenated by guidelines, user-provided chord progression, and the melody language model scores. Concatenating the melody for each lyric ultimately forms a complete song.

ROC has two major advantages. Firstly, ROC does not necessitate paired lyric-melody data, as it relies solely on training a melody language model. Secondly, ROC integrates commonly used composition guidelines, which not only make the generated songs sound more like they were composed by humans but also offer enhanced controllability. In summary, our main contributions are as follows:

(1) We introduce ROC, a novel paradigm for lyric-to-melody generation comprising a creation stage and a re-creation stage. ROC does not require paired lyric-melody data for training and can generate music that sounds more human-composed, guided by composition guidelines.

(2) We develop a series of designs to ensure ROC's effectiveness, including a lyric structure recognition algorithm that segments lyrics into choruses and verses for feature extraction, a short melody fragment generation procedure for im-

*Corresponding authors: Xu Tan (xuta@microsoft.com) and Rui Yan (ruiyan@ruc.edu.cn).

proved generation quality, and a novel composition pipeline based on retrieval and re-ranking that incorporates composition guidelines, etc.

(3) ROC surpasses strong baselines in terms of both objective and subjective metrics.

2 Related Works: Lyric-to-Melody Generation

In the early days, statistical and rule-based methods were employed for lyric-to-melody generation. [Long *et al.*, 2013] focused on lyric-note correlation and proposed a probability model, but overlooked musical knowledge. Another study [Fukayama *et al.*, 2010] studied Japanese prosody and its role in composition, proposing a probability model for generating melodies. Although incorporating more musical patterns, the model still ignored structural features, leading to a lack of repeating segments in the generated songs, making them sound less human-composed. These traditional methods required labor and expertise in music or linguistics, which led to a shift in focus towards neural-based methods.

The advent of neural networks has led to significant breakthroughs in lyric-to-melody generation. Many methods [Bao *et al.*, 2018; Yu *et al.*, 2021; Sheng *et al.*, 2020; Ju *et al.*, 2021] approach lyric-to-melody generation as a sequence-to-sequence task, learning a mapping from lyric sentences to melody phrases. These end-to-end models require a large amount of paired lyric and melody data; however, the lack of aligned data hinders research. SongMASS [Sheng *et al.*, 2020] trains lyric-to-lyric and melody-to-melody models separately, then conducts interaction between models to circumvent the need for paired data. However, as an end-to-end method, it suffers from low controllability, and its black-box nature does not guarantee that the learned musical patterns resemble human-composed music. TeleMelody [Ju *et al.*, 2021] divides the end-to-end generation pipeline into two stages: lyric-to-template and template-to-melody. Templates bridge the gap between lyrics and melodies, making the generation more controllable and reducing the need for paired data. However, since both stages in TeleMelody are neural-based, it is challenging to introduce composition guidelines, and error accumulation negatively affects generation quality.

We propose ROC to address these weaknesses. ROC does not require paired data, as only melodies are involved in training. In the re-creation stage, the retrieval process enables ROC to explicitly incorporate composition guidelines, making the generated songs sound more human-composed.

3 Background: Pop Music Composition

Creating pop music usually adheres to empirical composition guidelines that ensure good patterns in melody and lyric-melody feature alignment. We introduce a few guidelines that ROC takes advantage of:

- Structure alignment: Pop music usually adopts a verse-chorus form[1], where verses and choruses alternate and repeat. Choruses often contrast with verses in melody, rhythm, harmony, and dynamics, and are typically more instrumentally rich [Doll, 2011]. Consequently, pop music aligns lyric and melody structures, e.g., chorus lyrics with more melodic and harmonic chorus melodies.
- Melody Sharing: Most chorus sections are lyric-similar and contain the primary lyrical material of the song [Gotham *et al.*, 2023]. Thus, the same lyrics typically share a similar or even the same melody.
- Proper tonality: Tonality is the organization of all the tones and harmonies of a piece of music in relation to a tonic. Tonality impacts the emotional atmosphere of a song. For example, major tonality sounds enthusiastic, gorgeous, bright and cheerful while minor tonality sounds cold, melancholy, and magical. A proper tonality according to lyric sentiments helps expressing emotions.
- Pitch range consideration: A song should start with moderate pitches, allowing room for elevation in the chorus. Pitches usually do not fluctuate excessively within a verse or chorus.
- Chord progression guidance: "Chord" refers to multiple musical tones sounded simultaneously. The chord progression refers to the order in which chords are played in a song/piece of music. It wield a substantial influence over the mood, emotional impact, and overall trajectory of the music. For example, the chord progression "`C - Am - G - F`" typically elicits an energetic and emotionally satisfying response.[2] Typically, the chord progression returns to the tonic chord to create a sense of stable and smooth ending.

4 Methodology

The ROC pipeline is depicted in Figure 1 and comprises two stages: creation and re-creation. During the creation stage, we generate and store non-infringing music fragments in a database indexed by key features, which will be utilized in the re-creation stage. In the re-creation stage, users provide the lyrics and a desired chord progression. We then infer the melody features associated with each lyric sentence. By querying the database using these melody features, we obtain relevant melody fragments which satisfy the *structure alignment* guideline in Section 3. These candidates are then filtered, re-ranked, and concatenated by other guidelines and the melody language model scores. Concatenated melody fragments form a complete song. Further details about each stage are explained in this section.

4.1 Creation Stage

Melody Language Model. To ensure originality and avoid infringement, we train an auto-regressive melody language model based on the transformer architecture using melody data to generate new fragments. We represent music in the

[1] In music, form refers to the structure of a musical composition or performance.

[2] Letters such as `C` and `Am` are commonly used as as chord names in music. "Take Me Home, Country Roads" is a famous song using this chord progression:https://www.youtube.com/watch?v=1vrEljMfXYo

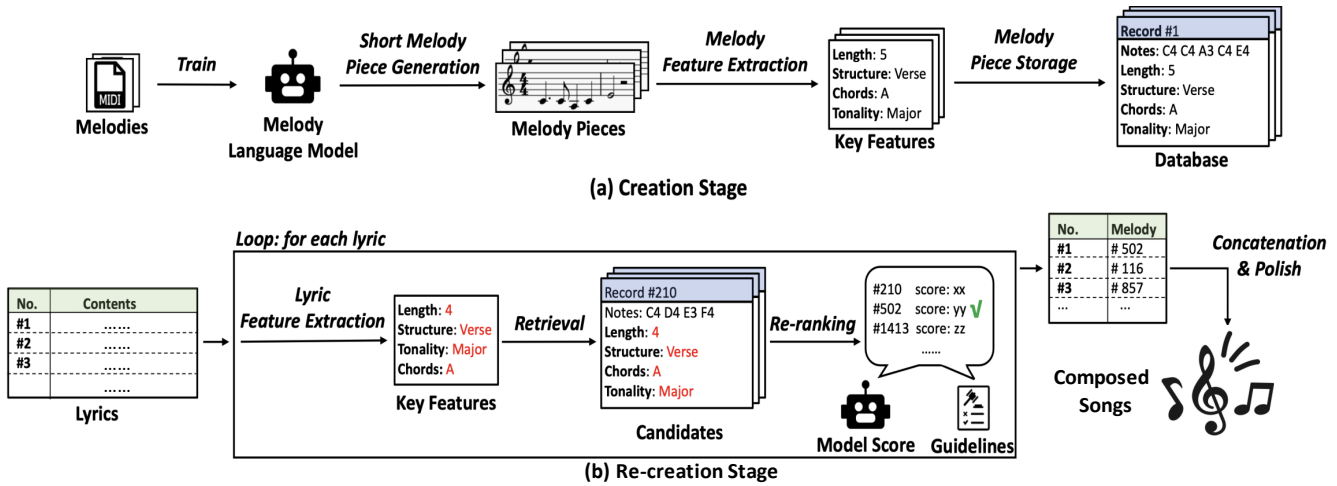

Figure 1: The pipeline of ROC. In creation stage, melody language model generates melody fragments which are stored in the database along with key features. In re-creation stage, we infer the features of each lyric in a song, which are used as the query to retrieve melody fragments. Retrieved fragments are re-ranked by both composition guidelines and the melody language model scores. After concatenating these melody fragments, ROC generates the final song.

style of OctupleMIDI encoding [Zeng *et al.*, 2021]. OctupleMIDI symbolically represents a music sequence as a series of MIDI event tokens $U = \{u_1, u_2, \ldots, u_n\}$. The melody language model is trained by minimizing the negative log-likelihood of predicting the next MIDI event token:

$$L = -\sum_{i}^{n} \log P(u_i | u_{<i}, \Theta), \tag{1}$$

This model is also utilized for re-ranking fragments in the re-creation stage, as discussed in 4.2. However, the neural model struggles to generate high-quality long sequences, so we implement a short melody fragment generation process, where the model predicts the next two bars given the previous two bars. We choose a two-bar prediction interval based on trials; longer intervals yield low-quality fragments and less flexibility in re-creation, while shorter intervals complicate the subsequent concatenation and polishing processes. We discard predicted fragments identical to their ground truth, resulting in a collection of high-quality, original melody fragments.

Melody Feature Extraction and Storage. We identify four key features in a melody fragment for storage: "Length", "Structure", "Chords", and "Tonality". These features also facilitate the incorporation of the guidelines outlined in 3 during the re-creation stage by allowing targeted querying of the database.

- **Length.** The number of notes in a fragment, used for basic alignment between lyrics and melodies. Generally, we determine the length of retrieved fragments based on the number of syllables in a lyric and align one syllable with one note. Sometimes, we allow one syllable to align with multiple notes, as detailed in 4.2.
- **Structure.** It indicates whether a fragment belongs to a chorus or verse. "Structure" of a melody piece is inferred using an algorithm based on the self-similarity matrix [Jayaram, 2018].
- **Chords.** The corresponding chords of a melody fragment, inferred based on note pitch distribution using the Viterbi algorithm [Magenta, 2020].
- **Tonality.** The tonality of a melody fragment, inferred by [Liang *et al.*, 2020].

A generated two-bar fragment is stored as two one-bar fragments and one two-bar fragment. We ignore the bar index and focus on melodic notes and key features. Also, we de-duplicate melody fragments.

Figure 2 demonstrates the feature extraction and storage process for a pop music. The music is divided into two-bar fragments, with the melody language model predicting the next two bars based on previous context. For example, the model predicts bars #3 and #4 using bars #1 and #2. If bars #3 and #4 are part of a verse in the original melody, the predicted bars are treated as verse components. The predicted notes and their key features are stored as a record.

4.2 Re-creation Stage

In this stage, users provide lyrcis a preferred chord progression to ROC. Then, ROC infer the melody features (e.g., length, structure, and tonality, etc.) associated with each lyric sentence. By querying the database using these melody features, we obtain relevant melody fragments which satisfy the *structure alignment* guideline. These candidates are then filtered, re-ranked, and concatenated by guidelines and the melody language model scores. Concatenate melody fragments form a complete song. In this section, we first introduce how to extract features in lyrics as queries for retrieval. Then, we discuss retrieval and re-ranking details.

Lyric Feature Extraction. As mentioned in *Structure Alignment*, Section 3, lyric-melody feature alignment is essential in pop music composition. Therefore, ROC infers the melody features associated with each lyric sentence for retrieving relevant melody fragments. Among the features

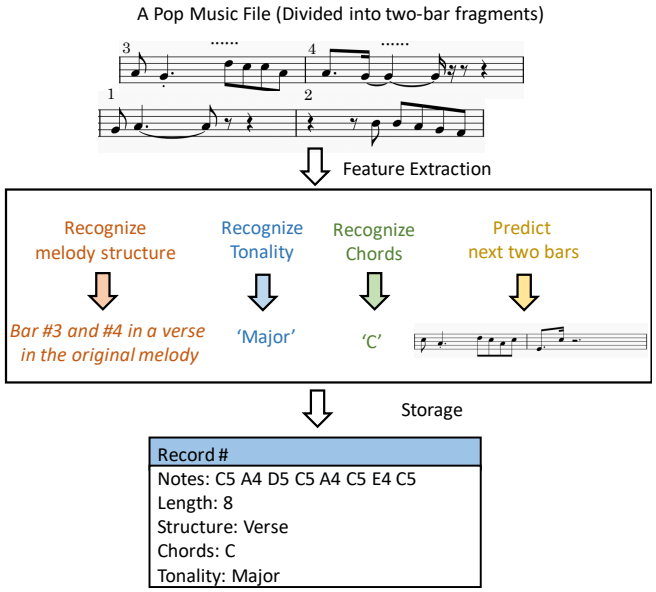

Figure 2: A case showing the feature extraction and storage of a melody fragment in the creation stage.

Algorithm 1 Lyric Structure Recognition with (K, L) *Repeat* Algorithm.

Input:
The string S abstracted from lyrics;
The segmentation granularity g.
Initialize:
Set all elements in $struct$ array as 0.
while $True$ **do**
  Find $R[L, K]$ with the largest L from S.
  if $L > g$ and $K > 1$ **then**
    Assign the $struct$ value of each element in $R[L, K]_i$ as the index in S of each element in $R[L, K]_1$, where i ∈ [2,K].
    Remove elements with non-zero $struct$ value from S.
  else
    break
  end if
end while

mentioned in 4.1, "Chords" are inferred based on the user-provided chord progression. "Length" is determined by the number of syllables in a lyric. "Tonality" is automatically set to major or minor, depending on the positive or negative sentiment of the lyrics. We use third-party libraries for Chinese [Deng, 2020] and English [Loria, 2020] sentiment analysis. To infer "Structure", we design a heuristic algorithm that recognizes structures (choruses and verses) in lyrics and considers the *melody sharing* guideline from Section 3 to determine which lyrics should share melodies with other lyrics. Details are provided below.

First, we define some preliminaries. Assume a song contains n sentences, each represented by the number of syllables

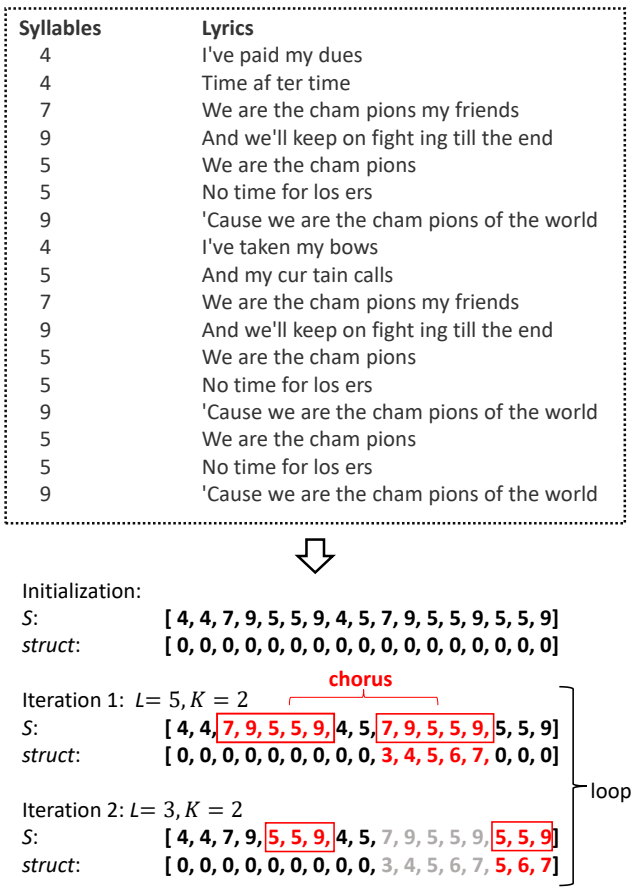

Figure 3: A case illustrating Algorithm 1. The original lyrics are simplified. We use red to highlight the operation in each step and grey to indicate these elements are currently skipped in S.

in it. The lyrics of a song can be abstracted into a number string S. A substring in S is called a (K, L) *Repeat* if it is of length L and repeats K times non-overlappingly in S. We denote the collection of these repetitive substrings as $R[L, K]$. $R[L, K]_i$ denotes the i-th repeat in $R[L, K]$, where i ∈ [1,K].

In most songs, the chorus is often the longest repeating segment, and segments with invariant lyrics have the same syllables and should share the melody (as described in Section 3). Consequently, the problem becomes searching for all $R[L, K]$ in S where $L > g$, with g being a searching granularity to control the minimum length of repetitive segments. In ROC, we set g as 2 by default. The $R[L, K]$ with the globally longest L indicates chorus parts, while the remaining parts are considered verses. Moreover, segments within the $R[L, K]$ for any L share the same melody with others.

To record the melody sharing relationship, we use an auxiliary array called $struct$ of length n. If the X-th lyric should share a melody with the Y-th lyric, the X-th element in $struct$ is assigned as Y. Initially, all elements in $struct$ are "0", indicating no sharing relationship. The algorithm details are shown in Algorithm 1 and Figure 3 provides an intuitive

illustration of the algorithm, using *We Are the Champions*[3] by Queen as an example. In the first iteration, the algorithm recognizes the chorus from "We are the champions, my friends" to "Cause we are the champions of the world." In the second iteration, the previously recognized chorus is skipped due to a non-zero $struct$ value. The loop stops as there are no more repetitive segments longer than g, which is 2 here. Lyrics corresponding to zero $struct$ values will independently retrieve melodies during the retrieval and re-ranking process, whereas those with non-zero $struct$ values will share melodies from the lyrics indicated by their respective non-zero values.

Retrieval and Re-ranking. With the extracted features, ROC performs the following operations to assign the best matching melody to each lyric sentence:

(1) If the $struct$ value of a lyric is "0", we use both features extracted from lyrics and user-provided chords to retrieve melody candidates from the database. Taking into account the *pitch range consideration* guideline from Section 3, retrieved candidates are filtered as:

- The first note should be in the range of G3 to F4. This avoids an overly pitched verse followed by an even higher pitched chorus, making suitable candidates rare.
- The first pitch in a melody fragment must be less than 8 semitones apart from the last pitch of the determined melody. This ensures ease of singing.

After filtering, we concatenate each candidate with the determined melody and use the melody language model to score them. The score is the normalized joint probability of a melody fragment (in the form of MIDI-event sequence):

$$s = (\prod_{i}^{n} P(u_i|u_{<i}, \Theta))^{\frac{1}{n}}, \tag{2}$$

(2) If the $struct$ value of a lyric is non-zero, we use the $struct$ value as the index to determine whose melody the current lyric should share with. For example, in Figure 1, when composing the second chorus, we directly reuse the melodies of the first chorus.

(3) Special cases. Since melody fragments have a maximum length of two bars, some long lyrics may retrieve no candidates. In this case, we split the lyrics into smaller pieces, retrieve melodies for each piece, and then concatenate them. Additionally, ROC supports one syllable aligning with multiple notes, which occurs with a certain probability. When this happens, ROC retrieves fragments with more notes than the number of syllables in the lyric and randomly decides which notes to connect.

Finally, we concatenate the retrieved fragments to form the final composition result.

5 Experimental Settings

5.1 Dataset

We use LMD-matched MIDI dataset [Raffel, 2016], which contains 45,129 MIDI data. First, we separate tracks [Guo *et al.*, 2020] and extract melodies. Tonalities are normalized to "C major" or "A minor". Ten percent of the data constitutes the validation set for training the melody language model. All data are used for constructing the database through the short melody fragment generation procedure, and there are 139,678 records in the database. As the test set, we select 20 English and Chinese songs.

5.2 Model Details and Baselines

The melody language model in ROC is a 4-layer decoder-only transformer [Vaswani *et al.*, 2017]. Each layer has 4 attention heads, and 256 input/output dimension. We use Adam optimizer [Kingma and Ba, 2015] with β=$(0.9, 0.98)$. The initial learning rate is 0.0001. We apply early stop scheme with 20 epochs patience and select the best checkpoint by perplexity on the valid set. The model applies top-5 decoding scheme.

We choose SongMASS [Sheng *et al.*, 2020] and TeleMelody [Ju *et al.*, 2021] as the representative of end-to-end baselines and non end-to-end baselines, respectively. Because the original SongMASS is trained with English lyrics, we follow [Ju *et al.*, 2021] to obtain a Chinese version. To evaluate the effectiveness of our lyric structure recognition algorithm, we also compare our algorithm with self-similarity matrix based methods [Fell *et al.*, 2018; Watanabe *et al.*, 2016] with pretrained embeddings GloVe [Pennington *et al.*, 2014] for English and CA8 [Li *et al.*, 2018] for Chinese.

5.3 Evaluation Metrics

We carry out objective and subjective evaluations of the generated songs. A notable issue in the melody generation field is the lack of universal metrics. In previous works, such as TeleMelody [Ju *et al.*, 2021], researchers often measure the differences between generated songs and ground truth. However, we argue that focusing on similarity to the ground truth does not effectively evaluate a model's creativity. In this paper, we primarily rely on human evaluation (subjective metrics) and employ objective metrics only to qualitatively reflect the generation quality.

Objective Metrics. (1) Diversity (Dist-n) [Li *et al.*, 2016]: this metric is widely used in NLP fields to measure the diversity of generation, i.e., how many unique n-grams in generated songs. This metric can measure the quality of music to a certain extent because a song having few unique n-grams is very monotonous. (2) Entropy (Ent-n) [Zhang *et al.*, 2018]: Dist-n neglects the frequency difference of n-grams. As a complement, we also compute Entropy which reflects how evenly the n-gram distribution is for a given melody:

$$Ent = \frac{1}{\sum_w F(w)} \sum_{w \in V} F(w) \log \frac{F(w)}{\sum_w F(w)}, \tag{3}$$

where V is the set of all n-grams, $F(w)$ represents the frequency of n-gram w.

Subjective Metrics. Objective metrics can only qualitatively reflect the generation quality and subjective metrics evaluate the quality better. Therefore, we recruit 10 evaluators having basic music knowledge to evaluate the performance of lyric-to-melody system from the following five aspects: (I) Structure (*Struc*): how well the the melody structure matches lyric structure? Specifically, whether lyrics with

[3]https://www.youtube.com/watch?v=04854XqcfCY

Models	Objective				Subjective					
	Dist-1	**Dist-2**	**Ent-1**	**Ent-2**	**Struc**	**Rhy**	**LMC**	**CLC**	**Melo**	**HL**
SongMASS (EN)	0.62	4.32	2.18	3.83	2.80	2.80	2.60	3.00	3.30	2.40
TeleMelody (EN)	0.81	4.61	2.30	3.88	3.20	3.30	2.90	3.80	3.80	3.10
ROC (EN)	**0.97**	**6.81**	**2.58**	**4.40**	**4.50**	**4.00**	**4.20**	**4.00**	**4.00**	**3.60**
SongMASS (ZH)	0.45	3.97	2.05	3.75	2.30	2.60	2.50	3.00	2.90	2.50
TeleMelody (ZH)	0.75	4.54	2.32	3.85	3.20	3.50	2.90	3.80	3.70	3.00
ROC (ZH)	**0.91**	**6.60**	**2.57**	**4.41**	**4.50**	**4.10**	**4.20**	**4.00**	**4.10**	**3.40**

Table 1: Objective and subjective evaluation results on Chinese and English lyric-to-melody test set.

similar rhythm patterns have similar melodies? (II) Rhythmic (*Rhy*): is the rhythm of a song flexible? (III) Lyrics and melodies compatibility (*LMC*): is lyric-melody feature alignment significant, e.g., when the lyrics enter the chorus, does the melody have a pitch lift or emotional intensity? Do similar lyrics share similar melodies? (IV) Cadence and lyric ending compatibility (*CLC*): whether cadences in the song sounds harmonic and whether there is an appropriate pause at the end of a lyric? (V) Melodic (*Melo*): is the melody beautiful and attractive? (VI) Human-composed Likelihood (*HL*): To what extent do the generated songs sound like human creations? In each aspect, evaluators can score from "1" for bad to "5" for good.

Evaluators listen to all songs generated in experiments in a random order. To eliminate familiarity bias that may arise from evaluators having previously heard the original songs, we display pseudo-lyrics (e.g., using spaces to represent syllable locations). After scoring, evaluators are informed of the actual lyrics, allowing them to focus on the structure when assessing relevant aspects.

6 Experimental Results

6.1 Main Results

Table 1 shows results of objective and subjective judgement. In objective experiments, ROC outperforms baselines in each language. The comprehensive gains on all metrics demonstrate the effectiveness of our new paradigm for lyric-to-melody generation: (1) Higher diversity scores of ROC imply that there are more melodic motions, which prevents the melody being unattractive. More diverse melodies are more likely to promote the emotion expression. (2) Higher entropy scores indicate that diverse notes are distributed more evenly than those of baselines, that is, the attractiveness and the ability of better emotion expression are more likely to maintain from the start to the end.

The above conclusions are confirmed in the subjective experiments, where ROC also outperforms baselines by a large margin in two languages: (I) ROC achieves significant gains in *Struc* thanks to lyric structure recognition and melody sharing scheme. In baselines, perhaps an implicit structural feature is captured during training, there are some weak structural patterns, but they are not as evident and neat as those in ROC. Also thanks to explicit structure features, we can distinguish chorus and verse which is an explicit activation for pitch range change or emotion expression promotion. (II) ROC has an improvement in *Rhy* because of more flexible notes, e.g., durations vary much more often than those in baselines. This is because in ROC, fragments are short whereas baselines suffer from modeling longer-term dependency. (III) Due to the feature match between melody fragments and lyrics, we also beat baselines on *LMC* by a large margin. (IV) Because the pipeline of ROC includes pause and cadence polish, *CLC* is ensured. (V) ROC generates more melodic songs (highest *Melo*). Better *Struc*, *Rhy*, *LMC* and *CLC* are also factors making songs more beautiful, improving *Melo*. (VI) ROC produces songs that resemble human compositions by incorporating commonly used composition guidelines, imbuing the generated music with patterns typically found in human-composed works. Overall, both the objective and subjective evaluation results demonstrate that the new paradigm ROC outperforms conventional generation paradigm.

6.2 Method Analyses

To better study the effect of each component in ROC and explore properties of ROC more thoroughly, we analyze the impact of the structure recognition algorithm, model scores and composition guidelines in retrieval and re-ranking, and the size of database. Because of the slight performance difference in different languages, we report the average scores of two languages in below.

Study on Structure Recognition. We disable the lyric structure recognition and report results in Table 2. Because *CLC* is guaranteed by polish operations in ROC, it is stable in this study and thus is omitted. Evaluators reflect that if we do not distinguish chorus and verse, the model will continue the song without an emotion activation or an explicit change of style so that melodies will be flat and less emotional, resulting in a smaller pitch range (lower Dist-1). Because lyric structure recognition is the foundation of the melody sharing scheme in ROC, without melody sharing, each sentence has its own unique melody, and thus Dist-2 increases. Overall, (w/o. recog.) impairs the generation quality by a large margin according to subjective evaluation because the rhythm is hurt and songs do not sound human-composed due to the lack of alignment between lyrics and melodies. Because the melody language model and guidelines ensure the basic quality and stability, the entropy scores maintain. This study reveals that aligning the structure of melodies to that of the lyrics is indispensable to high-quality lyric-to-melody generation.

Models	Objective				Subjective			
	Dist-1	Dist-2	Ent-1	Ent-2	Struc	Rhy	LMC	Melo
ROC	0.94	6.71	2.58	4.41	4.50	4.10	4.20	4.10
ROC w/o. recog.	0.84	7.80	2.58	4.41	2.20	3.90	2.10	3.60

Table 2: Study on lyric structure recognition.

Models	Objective				Subjective	
	Dist-1	Dist-2	Ent-1	Ent-2	Rhy	Melo
ROC	0.94	6.71	2.58	4.41	4.10	4.10
ROC w/o. model	0.69	3.57	2.41	3.62	3.80	3.70
ROC w/o. guidelines	0.91	9.07	2.76	4.79	4.20	3.60

Table 3: Study on re-ranking schemes.

Database Size	Objective				Subjective	
	Dist-1	Dist-2	Ent-1	Ent-2	Rhy	Melo
20%	0.93	6.54	2.49	4.28	3.80	3.90
50%	0.94	6.67	2.57	4.34	3.80	4.00
80%	0.96	6.60	2.56	4.35	4.00	4.10
100%	0.94	6.71	2.58	4.41	4.10	4.10

Table 4: Study on database size.

Study on Model Scores and Composition Guidelines. We study the impact of model scores and composition guidelines in retrieval and re-ranking on the performance of ROC. We remove the melody language model and composition guidelines respectively. Table 3 shows experimental results. Because *Struc*, *CLC* and *LMC* are unrelated to this study, their scores hardly change and thus are omitted. With only composition guidelines, too many candidates remain, and thus there is a lot of randomness in the final determination. The melodies are so diverse that *Rhy* increases a little. But too much diversity also decreases *Melo*. When composition guidelines are removed, there are also too many candidates remaining for the melody language model to score, and thus the running speed is 70 times slower than that of ROC with only composition guidelines. Lower diversity and entropy indicate that the melody is monotonous, which can be confirmed by subjective metrics *Rhy* and *Melo*. Overall, when composition guidelines are removed, songs sound dull and the melody progression is not harmonic as before whereas when the melody language model scores are removed, the quality of different parts of a song varies because of randomness. This study reveals that the melody language model scores and composition guidelines complement each other in retrieval and re-ranking, which are both crucial to the quality and efficiency of ROC.

Study on the Size of Database. Performance of ROC depends on the size of database. For example, if there is no melody that satisfies both the length requirements and chord progressions, ROC has to compromise, e.g., using the tonic chord as an alternative. Therefore, we study the effect of the size of database. We prune the database to 20%, 50%, and 80% of the full size, respectively. Results are listed in Table 4. First, as we expect, the running time and the database size are positively correlated. The running time increases from 3.99 seconds per song to 10.17 seconds per song when the size increases from 20% to 100 %. Second, because we remove data from the database randomly, the average quality and distribution of music fragments do not change, so *Dist* and *Ent* basically maintain. Because *Struc*, *LMC*, *CLC* is not determined by the database size and these metrics do not change, they are omitted. To our surprise, we find that as long as the average quality of melody fragments is satisfying, the generation quality is stable even though only 20% data remain. However, in practice, when we use 20%-size database, sometimes there are no candidates with matching features.

7 Conclusion

To address the two main issues in current lyric-to-melody generation methods—the lack of paired lyric-melody data for training and insufficient adherence to composition guidelines, resulting in melodies that do not sound human-composed—we propose ROC, a novel paradigm for lyric-to-melody generation. ROC divides the process into two stages: creation and re-creation. ROC does not require paired lyric-melody data for training and better aligns features between lyrics and melodies. Both objective and subjective experimental results demonstrate ROC's effectiveness.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC Grant No. 62122089), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. Ang Lv is supported by the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China.

References

[Bao *et al.*, 2018] Hangbo Bao, Shaohan Huang, Furu Wei, Lei Cui, Yu Wu, Chuanqi Tan, Songhao Piao, and Ming Zhou. Neural melody composition from lyrics. *CoRR*, abs/1809.04318, 2018.

[Deng, 2020] Da Deng. Chinese sentiment analysis library, which supports counting the number of different emotional words in the text. https://github.com/hiDaDeng/cnsenti, 2020.

[Doll, 2011] Christopher Doll. Rockin' out: Expressive modulation in verse-chorus form. Music Theory Online 17/3 (2011), § 2., 2011.

[Fell *et al.*, 2018] Michael Fell, Yaroslav Nechaev, Elena Cabrio, and Fabien Gandon. Lyrics segmentation: Textual macrostructure detection using convolutions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2044–2054, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[Fukayama *et al.*, 2010] Satoru Fukayama, Kei Nakatsuma, Shinji Sako, Takuya Nishimoto, and Shigeki Sagayama. Automatic song composition from the lyrics exploiting prosody of japanese language. *Proceedings of the 7th Sound and Music Computing Conference, SMC 2010*, 01 2010.

[Gotham *et al.*, 2023] Mark Gotham, Kyle Gullings, Chelsey Hamm, Bryn Hughes, Brian Jarvis, Megan Lavengood, and John Peterson. Open music theory. VERSE-CHORUS FORM, 2023.

[Guo *et al.*, 2020] Rui Guo, Ivor Simpson, Thor Magnusson, Chris Kiefer, and Dorien Herremans. A variational autoencoder for music generation controlled by tonal tension. In *Joint Conference on AI Music Creativity (CSMC + MuMe)*, 2020.

[Jayaram, 2018] Vivek Jayaram. Pychorus: Python module for detecting musical choruses. https://github.com/vivjay30/pychorus, 2018.

[Ju *et al.*, 2021] Zeqian Ju, Peiling Lu, Xu Tan, Rui Wang, Chen Zhang, Songruoyao Wu, Kejun Zhang, Xiangyang Li, Tao Qin, and Tie-Yan Liu. Telemelody: Lyric-to-melody generation with a template-based two-stage method. *CoRR*, abs/2109.09617, 2021.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[Li *et al.*, 2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

[Li *et al.*, 2018] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[Li *et al.*, 2020] Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online, July 2020. Association for Computational Linguistics.

[Liang *et al.*, 2020] Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. Pirhdy: Learning pitch-, rhythm-, and dynamics-aware embeddings for symbolic music. *CoRR*, abs/2010.08091, 2020.

[Long *et al.*, 2013] Cheng Long, Raymond Chi wing Wong, and Raymond Ka Wai Sze. T-music: A melody composer based on frequent pattern mining. *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 1332–1335, 2013.

[Loria, 2020] Steven Loria. Textblob v0.16.0 simple, pythonic, text processing–sentiment analysis, part-of-speech tagging, noun phrase extraction, translation, and more. https://github.com/sloria/textblob, 2020.

[Ma *et al.*, 2021] Xichu Ma, Ye Wang, Min-Yen Kan, and Wee Sun Lee. *AI-Lyricist: Generating Music and Vocabulary Constrained Lyrics*, page 1002–1011. Association for Computing Machinery, New York, NY, USA, 2021.

[Magenta, 2020] Magenta. A serializable note sequence representation and utilities. https://github.com/magenta/note-seq, 2020.

[Malmi *et al.*, 2016] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 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, 2016. Association for Computing Machinery.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global

vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[Raffel, 2016] Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.

[Sheng *et al.*, 2020] 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. *CoRR*, abs/2012.05168, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Watanabe *et al.*, 2016] Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan Smith, and Masataka Goto. Modeling discourse segments in lyrics using repeated patterns. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1959–1969, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

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

[Xue *et al.*, 2021] Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. Deeprapper: Neural rap generation with rhyme and rhythm modeling. *CoRR*, abs/2107.01875, 2021.

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

[Zeng *et al.*, 2021] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. MusicBERT: Symbolic music understanding with large-scale pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 791–800, Online, August 2021. Association for Computational Linguistics.

[Zhang *et al.*, 2018] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1815–1825, Red Hook, NY, USA, 2018. Curran Associates Inc.

[Zhu *et al.*, 2018] Hongyuan Zhu, Qi Liu, Nicholas Jing Yuan, Chuan Qin, Jiawei Li, Kun Zhang, Guang Zhou, Furu Wei, Yuanchun Xu, and Enhong Chen. Xiaoice band: A melody and arrangement generation framework for pop music. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 2837–2846, New York, NY, USA, 2018. Association for Computing Machinery.