AI-Lyricist: Generating Music and Vocabulary Constrained Lyrics

Xichu Ma Ye Wang Min-Yen Kan Wee Sun Lee
School of Computing, National University of Singapore, Singapore
ma_xichu@u.nus.edu, {wangye, kanmy, leews}@comp.nus.edu.sg

ABSTRACT

We propose AI-Lyricist: a system to generate novel yet meaningful lyrics given a required vocabulary and a MIDI file as inputs. This task involves multiple challenges, including automatically identifying the melody and extracting a syllable template from multi-channel music, generating creative lyrics that match the input music's style and syllable alignment, and satisfying vocabulary constraints. To address these challenges, we propose an automatic lyrics generation system consisting of four modules: (1) A music structure analyzer to derive the musical structure and syllable template from a given MIDI file, utilizing the concept of expected syllable number to better identify the melody, (2) a SeqGAN-based lyrics generator optimized by multi-adversarial training through policy gradients with twin discriminators for text quality and syllable alignment, (3) a deep coupled music–lyrics embedding model to project music and lyrics into a joint space to allow fair comparison of both melody and lyric constraints, and a module called (4) Polisher, to satisfy vocabulary constraints by applying a mask to the generator and substituting the words to be learned. We trained our model on a dataset of over 7,000 music–lyrics pairs, enhanced with manually annotated labels in terms of theme, sentiment and genre. Both objective and subjective evaluations show AI-Lyricist's superior performance against the state-of-the-art for the proposed tasks.

CCS CONCEPTS

• Computing methodologies → Natural language generation; Applied computing → Sound and music computing.

KEYWORDS

Lyrics Generation, Music, Adversarial Training, Language Learning

ACM Reference format:

Xichu Ma, Ye Wang, Min-Yen Kan, and Wee Sun Lee. 2021. AI-Lyricist: Generating Music and Vocabulary Constrained Lyrics. In 2021 ACM Multimedia Conference (MM '21), October 20-24, 2021, Virtual Event, China. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474085.3475502.

MM '21, October 20–24, 2021, Virtual Event, China.
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8651-7/21/10.
https://doi.org/10.1145/3474085.3475502

1 INTRODUCTION

Recent work in psychology and neuroscience has shown that language learners benefit from singing songs with suitable lyrics [2-9]. However, the number of existing songs is limited. Those with lyrics suitable for language learning are more limited as lyrics are typically written on a restricted set of topics (e.g., love), thus lacking diversity. As a result, most existing lyrics have limited value for language learning. Furthermore, songs with suitable lyrics may not match users' music preferences, while songs that users like might not have lyrics matching their corresponding linguistic ability and learning objectives. This has motivated us to solve a novel and challenging research problem, formulated as generating novel yet meaningful lyrics given the required vocabulary and a MIDI file, which reflects the user's preference as input.

We investigate the problem of generating music and vocabulary constrained lyrics. We aim to generate lyrics based on a piece of music in MIDI format, a small set of keywords (e.g., 5 words) that the lyrics must contain, and language proficiency level (which corresponds to which vocabulary set to choose from for the lyrics). We propose a framework, AI-Lyricist, to generate lyrics that (1) are aligned appropriately to the melody, (2) are relevant to the music in style, (3) are semantically meaningful, (4) include all the given keywords, and (5) only include words in the given vocabulary set (e.g., the most frequently used 3000 words). The architecture of AI-Lyricist is depicted in Figure 1. For simplicity, this paper focuses on English, but the framework can be easily generalized to other languages with corresponding datasets.

We face multiple challenges when generating lyrics for language learning. First, in addition to general requirement of lyrics to be coherent, meaningful, and creative, our generated lyrics are also constrained by vocabulary and grammatical correctness. Second, as a cross-modal problem, the generated lyrics should be relevant to the music in terms of style, and more significantly, precisely aligned to the rhythmic pattern of the melody. Third, to leverage the style information contained in the accompaniments of music, also, to overcome the limitation of input data type in practical applications, we accommodate multi-channel music as input. Therefore, it is necessary yet difficult to automatically analyze the structure of the input music, including melody channel identification, melody phrase partition, repeated phrase detection and syllable template extraction.

To address the above challenges, first, we enhance a paired music–lyrics dataset, adding stylistic labels (theme, sentiment and genre) by human annotations. We adapt the concept of cross-modal feature projection from poetry generation field [25] and learn a deep coupled music–lyrics embedding model from the enhanced dataset with gated-CNN (Convolutional neural network) [22] stylistic features of music and BERT [23] features of paired

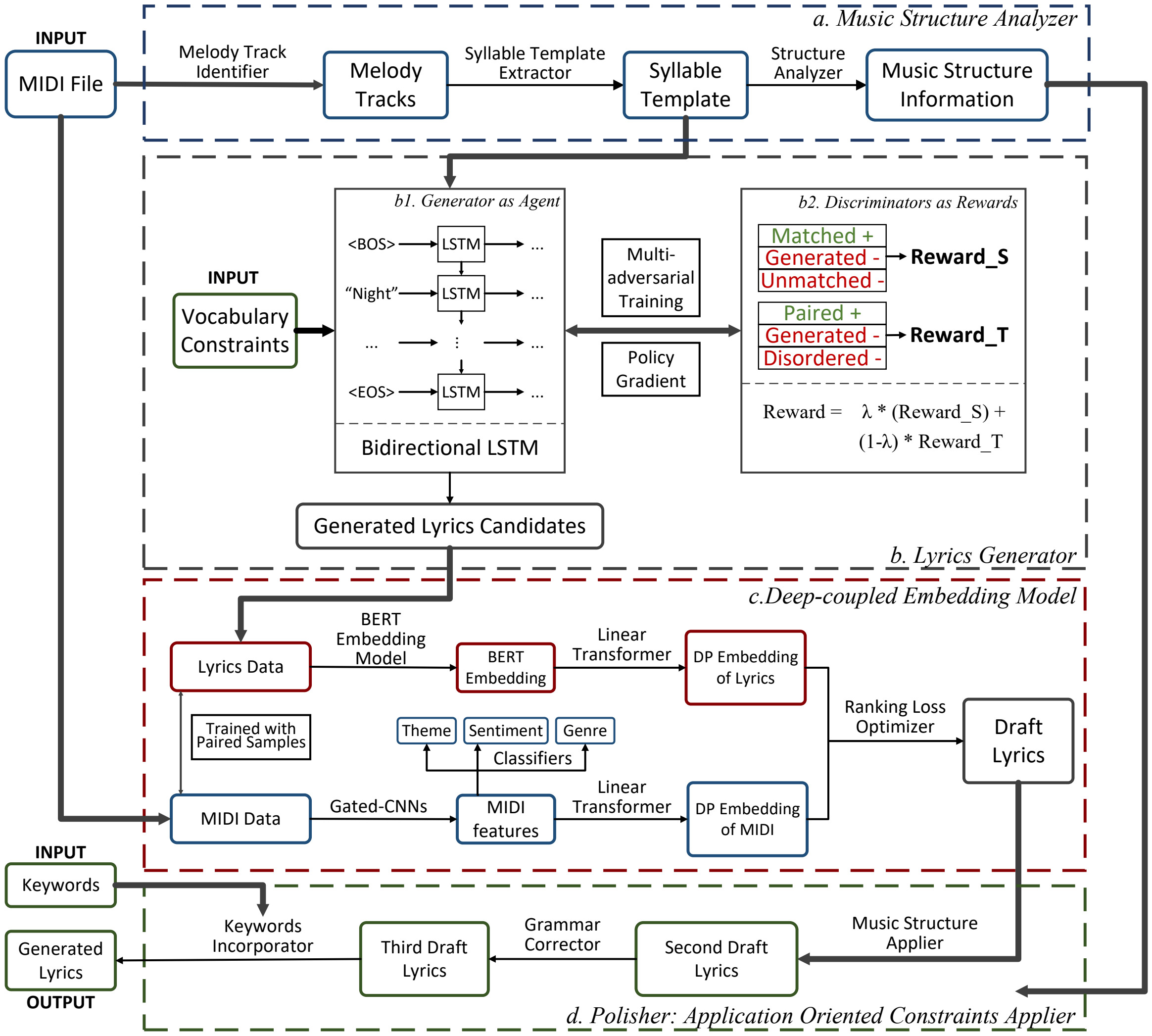

Figure 1: The architecture of proposed automatic lyrics generation system. The generation is executed through the four sub-modules in sequence: a. Music Structure Analyzer, extracting syllable and music structure information from input MIDI; b. A SeqGAN based lyrics generator; c. Deep coupled music-lyrics embedding model, selecting best match for the input MIDI in style from a batch of generated lyrics candidates; and d. Polisher, incorporating personalized constraints regarding language learning.

lyrics. Second, we tackle the four above-mentioned sub-tasks in the automatic analysis of multi-channel music: We utilize the "expected syllable number" of a song to improve the accuracy of melody identification. We partition the melody into phrases and detect repeated sections according to pauses and rhythmic patterns of melody. We also arrange a syllable template for each phrase. Third, we propose a Sequence Generative Adversarial Networks (SeqGAN) based lyrics generation system that ensures the relevance and syllable alignment to the music. We apply a bidirectional Long Short-Term Memory RNN (Bi-LSTM Recurrent Neural Networks) as the generator with multi-adversarial training optimized through policy gradient; two discriminators judge the generated lyrics and give feedback regarding syllable alignment and text quality. Such an architecture brings advantages of avoiding the exposure bias problem and the lack of standard loss in creative long text generation tasks. Specially, in the generator, we condition the input tokens with an additional feature domain of syllable planning, so that the generator can learn the syllable numbers of words and plan the subsequent syllable arrangement on its own.

To make the system useful for language learning, we propose to constrain the lyrics generation with a module called Polisher: To match a user's language level, it exerts a mask on the generator to restrict the selectable words; to achieve a user's learning objectives, it uses new words (i.e. the words to learn) to displace the most similar words in the generated lyrics; to strengthen the impression of new words, the polisher repeats the lyrics in repeated sections; and finally, it corrects possible grammatical errors in the lyrics before publication.

We conducted experiments on an independent paired music–lyrics dataset to generate lyrics from the whole music. The results of the objective evaluation show that our proposed system achieves a desired balance between translative quality, information density, novelty and cross-modal relevance. In the subjective evaluation, our user study shows that our proposed system surpasses compared models regarding fluency, coherence, meaningfulness, poetic aesthetics, syllable alignment to melody and relevance to the music.

The main contributions of this paper are three-folded:

- We propose to generate lyrics from multi-channel music. This is the first attempt to generate lyrics that match both the style and syllable pattern of given music, overcoming the limitation of input types.
- We adopt the concept of deep coupled embeddings from cross-modal tasks to lyrics generation and further propose to incorporate it into a SeqGAN based lyrics generating model where two discriminators are applied to give feedback regarding syllable alignment and text quality, which enables computers to write lyrics in a human-like manner.
- We propose the Polisher module to constrain lyrics generation with mandatory keywords and a vocabulary set. We thus create AI-lyricist, the first personalized lyrics generation system for language learning which accommodates to a user's language level, learning objectives and music preference.

The rest of the paper is organized as follows. Section 2 delves into previous work on lyrics generation and poetry generation. Section 3 details the framework design of AI-Lyricist, detailing its four-module design. Section 4 presents the experimental settings and results. Sections 5 and 6 discusses future work and concludes the paper, respectively.

2 RELATED WORK

2.1 Lyrics Generation

Early studies in automatic lyrics generation (ALG) are largely based on template filling and randomized algorithms [10, 11, 24, 26]. In one study, words are randomly picked to fill a line length template [10] while another applies Dijkstra's algorithm on a word graph, searching for a sentence to match a given syllable template [11]. Tri-gram interpolation is utilized to generate lyric sentence in [24], continuously predicting the next word from the previous two. Tra-la-Lyrics 2.0 generates lyrics with a list of heuristic rules based on analysis of the relationship between lyrics, melodies, syllables and beats [26]. Some subsequent studies have investigated the impact of Markov Models on improving the stylistic and topic coherence of lyrics generation [14, 27]. Although the results of early work lack coherence and meaningfulness, the ideas of word-picking and template use have been extended in later studies.

With the rapid development of deep learning, RNNs have become a mainstay in ALG. Several studies employ different types of RNNs to generate lyrics either predicting the next sentence [13] or next words [12, 15, 16, 28]. Taking this one step further, some studies seek to improve the quality of generated lyrics by incorporating keywords [18, 29], content [30], syllable structure [19, 20], visual features of music's frequency spectrum [31], or existing lyrics [32] as conditions. SeqGAN is also utilized to generate rewritten versions of the original lyrics [33]. Recently, the huge GPT2 model is used in an interactive lyrics generation system [35]. Some researchers have realized the significance of whole song generation, recomposing new lyrics as the condition of melody generation [34]. Their new attempts are novel, however different from the procedure of human songwriting.

To the best of our knowledge, few studies have considered the tight coupling between the music and lyrics. Furthermore, none of the existing ALG systems included application-oriented constraints (i.e., keywords and proficiency-adapted vocabulary set) that are crucial for language learning applications.

2.2 Poetry Generation

Although there are differences between lyrics and poetry [43], poetry generation methods provide insights into ALG tasks. Template and grammar-based approaches [44-46], generative summarization under constrained optimizations [47], and statistical machine translation models [48, 49] are more commonly seen in conventional methods. Specifically, studies apply grammatical and semantic templates in poetry generation [45, 46]. Yan et al. propose a method based on summarization techniques for poem generation, retrieving candidate sentences from a large corpus of poems based on user queries, then cluster and summarize the constituent terms into poetry lines [47]. Other studies treat this problem as a statistical machine translation problem where each line is treated as a translation of the previous line [48, 49].

Similarly, deep learning approaches in poetry generation have attained promising results. RNNs are thus widely applied to generate poems that are indistinguishable from those written by human poets [50-62]. For example, unsupervised learning has been used to estimate the stress patterns of words in a poetry corpus, with a finite-state network also used to generate short English love poems [50]. Several RNNs/LSTM based methods write Chinese poems [51-59], where these methods take fluency, rhythm [51], iambic meter [53], style [54], keywords [55], creativity [56] and human interaction [55, 61] into account. A recent study solves the cross-modal problem of generating poetry from a given image [25].

3 APPROACH

This paper delves into automatic lyrics generation from multi-channel music with keywords and vocabulary constraints. We dissect the goal into four sub-tasks and propose a system comprised of four modules to correspondingly fulfill each task: (1) A music structure analyzer which identifies melody from the input MIDI file, extracts the syllable template and detects repeated sections as music structure. (2) A lyrics generator with two discriminators to generate a group of draft lyrics candidates that match the extracted syllable template. For this purpose, the lyrics generation is conducted in a multi-adversarial procedure and optimized through policy gradient. The generator-as-agent iteratively determines the next word selection action, following the policy defined by its parameters. After generating sample lyrics, it observes rewards provided by two discriminative networks, which judge the syllable alignment and text quality of the samples. (3) A deep coupled music–lyrics embedding model which retrieves the best match of the generated draft lyrics candidates, maximizing the cosine similarity of their embedded features. (4) The Polisher which finally imposes keywords and vocabulary constraints to the generated lyrics and also applies the extracted music structure.

3.1 Music Structure Analyzer

Taking multi-channel music as input is a must because the theme, sentiment and genre of music are usually reflected by the accompaniment, such as chord and bass. Another justification for accommodating multi-channel music is that it is not practical for language learning applications to restrict the user's input.

Thus, the main function of the music structure analyzer is to extract the syllable template as well as the music structure (e.g., ABABC) from the input music. To accommodate multi-channel whole music, melody identification is the first step. Although identifying the melody channel from MIDI file remains an open question, we take a shortcut for pop songs by introducing the concept of "expected syllable number" of a song. When the total syllables are too many, it is difficult for the singer to catch up with the melody, especially when the tempo is fast. Similarly, if there are too few syllables, at least one syllable will span multiple notes and will be difficult to sing out. According to the statistics of pop songs, the feasible range of expected syllable number (ES) can be represented by a function of tempo (tm) and song duration (sd):

$$ES \in \left[\frac{1.7 \times sd(s) \times tm(bpm)}{60}, \frac{2.3 \times sd(s) \times tm(bpm)}{60}\right] \quad (1)$$

Specifically, we select a channel as the melody based on a weighted rating function, considering factors including the highest mean pitch (f_1), highest entropy (f_2), highest note-on rate (f_3), lowest overlapped note-on rate (f_4) and smallest distance to the expected syllable number range (f_5), where w_j denotes the weight of the corresponding rating factor.

$$\underset{i}{argmax} \sum_j w_j f_j(track_i) \quad (2)$$

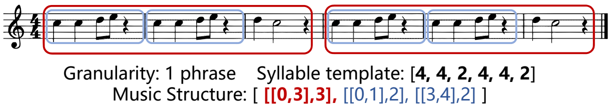
Granularity: 1 phrase
Syllable template: [4, 4, 2, 4, 4, 2]
Music Structure: [[[0,3],3], [[0,1],2], [[3,4],2]]

Figure 2: Extraction of the music structure and syllable template.

The extracted melody is partitioned into phrases by rests longer than a quaver (i.e., eighth note); then a sliding window is applied to detect the longest repeated phrases as repeated sections and recursively dives into each section until reaching a given granularity. For example, in Figure 2, the red boxes are two repeated sections while the blue boxes within are two smaller repeated sections. The numbers of notes in each phrase together form a syllable template of the song. For simplicity, we assume that every syllable in a song's lyrics corresponds to one note of that song's melody. The syllable template and music structure are output as a list of syllable numbers and a list of bi-gram tuples that indicate the start points and lengths of repeated sections.

3.2 Lyrics Generator and Discriminators

3.2.1 Lyrics generator as an agent. A syllable-conditioned bidirectional LSTM-RNN lyrics generator functions as an agent. Instead of an encoder–decoder structure that decodes the music encoding to lyrics, the generator generates lyrics from scratch for two reasons: First, the encoder–decoder structure might limit the creativity of the lyrics generator. As a creative generation task, there is no one-to-one match for music and lyrics. A piece of music can be filled with different lyrics, which might all create excellent works. Second, for lyrics, text quality is more important than the stylistic relevance, so eliminating music decoding allows the generator to be trained on a larger lyrics dataset and thus improves the text quality of generated lyrics. Later we compare the performance of the proposed model with that of an encoder–decoder structure and show the superiority of our approach.

As discussed in [25], we utilize a non-hierarchical RNN rather than a hierarchical language model that is applied in many lyrics generation models. This is because there are relatively few words in lyrics and the hierarchy between lyric lines is less consistent. Instead, we regard <EOS> (end of sentence) as a word in the vocabulary.

A bidirectional LSTM-RNN is used in pre-training while only the forward portion is used in adversarial training. A bidirectional structure is more capable of predicting the syllable distribution of words since at any time step, it holds the accumulated syllable information of both previous and subsequent contents. During the pre-training phase, the generator is trained to predict the next token given a word sequence. If θ denotes the parameters of the generator, the optimization target is to learn θ by maximizing the likelihood of the observed sample $Y = y_{1:T} \in \mathbb{Y}^*$ where T is the maximal length of lyrics and $\mathbb{Y}^*$ is the space of all possible lyrics. The target in pre-training is as following, where c denotes the hidden states:

$$\underset{\theta}{argmax}\left(\prod_{t=1}^{T} p_\theta\left(y_t | y_{1:t-1}, c\right) + \prod_{t=T}^{1} p_{\overleftarrow{\theta}}\left(y_t | y_{T:t+1}, c^{\leftarrow}\right)\right) \quad (3)$$

When it comes to the adversarial training phase, θ is served as a policy. The target is to maximize $E(R_T|\theta)$, the expected total rewards of the whole generated lyrics given by the discriminators at time T:

$$J(\theta) = E(R_T|\theta) = E \sum_{t=1}^{T} r(t|\theta) \quad (4)$$

With this strategy, however, the rewards can only be observed when a set of complete lyrics is generated. To provide immediate reward for intermediate results at any time step t, a Monte-Carlo sampling strategy is applied with a rollout policy to sample remaining words from time step $t + 1$ to T. Therefore, an approximated expected gradient with a single sample can be finally expressed as:

$$\nabla_\theta J(\theta) \approx \sum_{t=1}^{T} \nabla_\theta log\, p_\theta(y_t|y_{1:t-1}) \sum_{t=1}^{T} r(t|\theta) \quad (5)$$

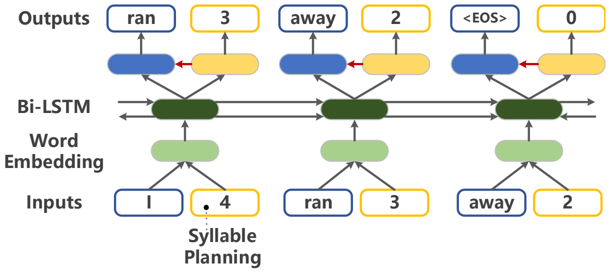

Figure 3: Condition of syllable planning in Bi-LSTM generator.

Syllable Planning. The generator is conditioned with the syllable template extracted from the given music, so it is more syllable-aware and can better align the generated lyrics to the melody's rhythmic pattern. A study has proposed an LSTM-RNN based generator that predicts the next word and its syllable number, thus counting the total number of syllables of the generated lyrics [19]. We take one step further to input and predict the syllable planning (i.e., how many syllables the remaining lyrics have) thus teaching the generator the syllable number of different words, and how to arrange the subsequent syllables. The bidirectional structure enhances the effects of syllable planning because at any time step t, the generator holds the states of both forward and backward passes.

As shown in Figure 3, the input word is conditioned with additional information of the expected syllable number of the remaining words, and the generator is asked to predict the expected syllable number after picking the word at time step t. The optimization target is as follows:

$$\underset{\theta}{argmax}(\prod_{t=1}^{T} p_{\theta}(s_t|s_{t-1}, c) + \prod_{t=T}^{1} p_{\theta}^{\leftarrow}(s_t|s_{t+1}, c^{\leftarrow})) \quad (6)$$

where s_t denotes the expected remaining syllable number of the current line at time step t. Also, the intermediate result of syllable planning prediction is shared with word prediction as the syllable condition (the red arrows in Figure 3).

3.2.2 Discriminators as Rewards. The proposed discriminators inspect two of the qualities in lyrics generation from multi-channel music: (1) the syllable alignment between lyrics and melody, and (2) the text quality. This means the lyrics should approach the level of human-written lyrics to the greatest extent possible. Therefore, in the adversarial training phase, a syllable alignment discriminator and a text quality discriminator guide the generator by providing their judgment as rewards.

***Syllable Alignment Discriminator*.** This discriminator's goal is to judge whether the input lyrics match the given syllable template. A discriminative network classifies an input pair of {lyrics, syllable template} into three classes: "matched" as positive samples, "unmatched" and "generated" as negative samples. In training, the matched samples come from the ground-truth; the unmatched samples come from randomly paired lyrics and syllable templates, and the generated samples come from the outputs of the generator. To avoid the bias of imbalanced data, the sample numbers of the three different classes are held constant.

A lyrics sample Y and a syllable template S are fed into a bi-directional LSTM-RNN connected with a fully connected layer. Then a softmax layer calculates the probability of being classified as the three classes.

$$C_s = softmax(W_s * BiLSTM_{\gamma}(Y, S) + b_s) \quad (7)$$

where W_s, γ and b_s are parameters to be learned and C_s denotes the probability of the sample falling into the three classes.

***Text Quality Discriminator*.** Similarly, another Bi-LSTM based discriminator is trained to judge the text quality of a lyrics sample, classifying an input lyrics sample into three classes: qualified as positive example while generated and disordered as negative examples. Qualified samples come from ground truth; generated samples are from the generator; and disordered samples are generated either by randomly exchanging some words of a ground-truth lyrics, or stitching word segments of several ground-truth lyrics together. The procedure is expressed by:

$$C_q = softmax(W_q * BiLSTM_{\delta}(Y) + b_q) \quad (8)$$

where W_q, δ and b_q are parameters to be learned, and C_q is the probability of the sample falling into the three classes.

***Overall Rewards*.** The overall rewards $R(Y, S)$ that the generator observes is a linear combination of the probability that a given pair of lyrics and syllable template; i.e., $\{Y, S\}$ is classified as a positive sample by the two discriminators, whose significance is controlled by a parameter λ (summing to 1):

$$R(Y, S) =$$

$$\lambda * C_s(c = matched|Y, S) + (1 - \lambda) * C_q(c = qualified|Y) \quad (9)$$

3.2.3 Multi-Adversarial Training. The target of multi-adversarial training of a single generator and a single discriminator can be expressed as a minimax optimization process:

$$\underset{G}{min}\,\underset{D}{max}\,V\,(G, D) =$$

$$E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z)))] \quad (10)$$

Further, in multi-adversarial training with several discriminators, the optimization objective of the generator can be rewritten as:

$$\underset{G}{Min}\,max\,F(V(D_1, G), \dots, V(D_n, G)) \quad (11)$$

where F is the linear combination function defined in (9) and n equals 2 for this system. Hence, the generator is optimized to gain the largest overall reward by generated samples that both discriminators judge as positive samples; and the two discriminators are optimized in parallel to distinguish positive and negative samples.

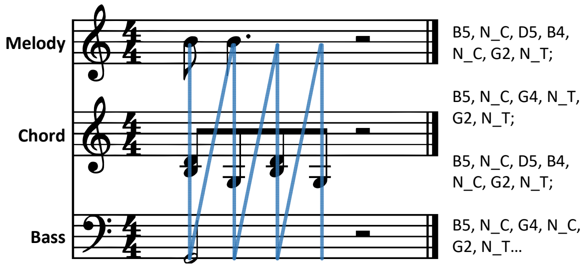
Melody
Chord
Bass
B5, N_C, D5, B4, N_C, G2, N_T;
B5, N_C, G4, N_T, G2, N_T;
B5, N_C, D5, B4, N_C, G2, N_T;
B5, N_C, G4, N_C, G2, N_T...

Figure 4: Zig-zag encoding strategy of multi-channel music. N_C and N_T explicitly indicate the next channel and next time step.

3.3 Deep Coupled Music–lyrics Embedding

Inspired by the effectiveness of the deep coupled embedding model in poetry generation [25], we adapt this idea to lyrics generation from music. A successful song usually includes music and lyrics that complement each other. For example, words with many syllables are more common in soothing songs while compact syllabled words match the fast-tempo music better [63, 64]. Also, in country songs, the lyrics are often more narrative and story-driven than in other genres while dance music can contain meaningless repeating staccatos. With the assumption that paired music and lyrics share perceptive connotation so that they can be projected into a learnable cross-modal embedding space, a deep coupled music–lyrics embedding model allows us to calculate the relevance of a pair of music and lyrics by the cosine similarity of their projected feature vector in the embedding space.

To train such a deep coupled music–lyrics embedding model, we collect a paired music–lyrics dataset with lyrics from the dataset published in [36] and their corresponding MIDI files in the LMD dataset [65]. We further enhance the collected music–lyrics dataset by annotating the songs with stylistic labels, namely theme, sentiment and genres. As shown in Figure 4, we encode multi-channel music by scanning notes of all channels in a zig-zag fashion [40], explicitly indicating the channel shift and time step.

Three gated-CNN based multi-class classifiers are trained on the three datasets (i.e., theme, sentiment, and genre) independently, optimized by the cross-entropy loss shown in (12), where N denotes the number of classes, y_c is 1 when the sample is of class c, otherwise 0, and p_c denotes the predicted probability of the sample belonging to class c.

$$Loss = -\sum_{c=1}^{N} y_c \, log(p_c) \tag{12}$$

We leverage gated-CNN models instead of traditional LSTM to speed up the training because the encoding can be immensely long when the music consists of many channels.

Then we concatenate the deep features of the penultimate fully-connected layer of the three gated-CNN models to be an $M(M = D \times 3)$-dimension feature vector $o \in \mathbb{R}^M$ as the input of the music for the music–lyrics embedding. The output of the music embedding $\mu \in \mathbb{R}^K$ is a linear transformation of the feature vector o:

$$\mu = W_o \cdot o + b_o \in \mathbb{R}^K \tag{13}$$

where $W_o \in \mathbb{R}^{K \times M}$ and $b_o \in \mathbb{R}^K$ are parameters to be learned.

For the lyrics, we extract the BERT features for each lyrics sample in the enhanced music–lyrics paired dataset to be an L-dimension feature vector $q \in \mathbb{R}^L$ as the input of the lyrics for the music–lyrics embedding. Similarly, the output of the lyrics embedding $\nu \in \mathbb{R}^K$ is calculated by:

$$\nu = W_q \cdot q + b_q \in \mathbb{R}^K \tag{14}$$

where $W_q \in \mathbb{R}^{K \times M}$ and $b_q \in \mathbb{R}^K$ are parameters to be learned.

Finally, the paired music and lyrics are projected to the same embedding space by minimizing a pairwise margin ranking loss with cosine similarity:

$$\begin{aligned} Loss = \textstyle\sum_{\mu} \sum_{n} max(0, m - \mu \cdot \nu + \mu \cdot \nu_n) + \\ \textstyle\sum_{\nu} \sum_{n} max(0, m - \nu \cdot \mu + \nu \cdot \mu_n) \end{aligned} \tag{15}$$

where μ_n and ν_n are embeddings of unpaired contrastive music and lyrics samples, and m is the contrastive margin.

In the entertainment industry, music producers usually recruit several lyricists to write several versions of lyrics for one piece of music, then select the best version as the final draft [66]. In the lyrics generation process (compared to the training process), the deep coupled music–lyrics embedding model imitates this convention by selecting the candidate lyrics most relevant to the input music from a batch of generated candidates, improving the creativity and stylistic relevance.

3.4 Polisher: Application-Oriented Constraints Applier for Language Learning

On top of the draft lyrics, we propose Polisher to apply constraints regarding language learning requirements to the generated lyrics. Given a vocabulary set reflecting the user's mastered words, Polisher will create a mask as an extra condition of the lyrics generator. The probabilities of selectable words will be re-generalized while the probabilities of other words beyond the vocabulary are set to 0. Given keywords as the user's learning objectives, a Word2Vec embedding model is trained on Wikipedia corpus and substitutes the semantically closest words in the draft lyrics with given keywords.

Since repeated practice of listening and reading is crucial for language learning, the Polisher copies the exactly same lyrics to its repeated segments, according to the extracted music structure. This strengthens the impression of the words to learn. For example, for music with a structure of ABACAD, the lyrics are repeated in all A-type segments. Finally, a grammar corrector model corrects possible grammatic errors. A full example of lyrics generation procedures is shown in Figure 1.

Besides assisting language learners, Polisher provides an interactive lyric writing mode for language teachers, allowing them to supervise the lyrics generation. In each round, the lyrics generator will provide a batch of generated next sentence candidates; and the language teacher can pick the best one as the next line; this process will repeat until the full lyrics is completed. Under human supervision, the generated lyrics further improve in global coherence and fluency.

4 EXPERIMENT

4.1 Datasets

For the training of the lyrics generator and deep coupled music–lyrics model, we collect three datasets: LMD-Paired-Music-Lyrics dataset (LPM-Lyrics), a paired music–lyrics dataset, Reddit-Paired-Music-Lyrics dataset (RPM-Lyrics) and a collected large lyrics corpus without the corresponding music, named Large-Lyrics dataset (L-Lyrics).

The IPM-Lyrics Dataset and RPM-Lyrics consist of the lyrics published in [36] and their corresponding MIDI files retrieved from original LMD and Reddit datasets. The L-lyrics consist of a larger number of pop song lyrics, which is also used as the word corpus for the generator. Further, we enhance LPM-Lyrics and RPM-Lyrics by manually annotating each song with stylistic labels of three aspects, namely theme (of 11 classes), sentiment (of 6 classes) and genre (of 18 classes); and we append the syllable planning template to each lyrics sample in all three datasets with automatic syllable analysis tool in NLTK (Natural Language Toolkit) [37].

Table 1. Facts about three datasets

Dataset	# Lyrics	# MIDI	# Lines/lyrics	# Word/line
LPM-Lyrics	7,211	7,211	30.2	4.7
RPM-Lyrics	3,977	3977	29.3	4.5
L-Lyrics	140,435	/	51.5	5.4

Details about the three datasets are listed in Table 1. The L-Lyrics dataset is used to pre-train the lyrics generator and the LPM-Lyrics dataset is used to train the deep coupled music–lyrics embedding model as well as to fine-tune the music conditioned lyrics generator for some compared methods. To ensure the validity of the evaluation, RPM-Lyrics is used as an independent test set in both objective and subjective evaluations.

4.2 Compared Methods

Most of studies on ALG generate lyrics from textual clues or melody. As far as we know, there are few studies on ALG from whole music, let alone for practical applications like language learning. Thus, to evaluate the effectiveness of the proposed system on this task, we select three baseline models: two classical models of Seq2Seq tasks, namely a machine translation model (MT) and an encoder–decoder model (ED), and SeqGAN. These models are the state-of-the-art considering their success on text generation tasks. We compare the baseline models with two proposed systems with different configurations: one generating lyrics by decoding the deep coupled embedding of input music (Pre-Embedding Music to Lyrics with Both discriminators, PRE-M2L-B) and the other, (our final released system), generating lyrics from scratch (Post-Embedding Music to Lyrics with Both discriminators, POST-M2L-B). For ablation study, we also include the performance of POST-M2L (without syllable-awareness), POST-M2L-S (with only syllable discriminator) and POST-M2L-T (with only text-quality discriminator) for comparison.

4.3 Objective Evaluation

Measuring the quality of lyrics is a complicated task because the requirements come from different aspects, especially when the lyrics serves as materials for language learning. We propose to apply five objective metrics to investigate the performance of our proposed system: Novelty, Informative Density, Relevance, BLEU (Bilingual Evaluation Understudy) [42], and Application-oriented satisfaction. An average of generalized values of the five metrics is regarded as the overall quality. The objective evaluation is conducted on the independent RPM-Lyrics dataset.

Novelty. Novelty is a crucial metric for creative text generation tasks to avoid the generator cheating the discriminators by simply repeating common words/phrases in the corpus. We adopt the bi-gram and tri-gram novelty [38] to evaluate how likely the generator will choose infrequent phrases in the corpus. We count the occurrence of n-grams in the training dataset (L-Lyrics) and take the top 2,000 ones as frequent phrases.

$$Novelty - n = m_{n\_IF}/m_n \tag{16}$$

where n denotes either bi-gram or tri-gram, while $\mathrm{m}_{\mathrm{n\_IF}}$ and m_n denote the number of infrequent n-gram phrases and the number of total n-gram phrases in all the generated lyrics respectively.

Informative Density. Informativeness is an important indicator of creativity which measures the unique n-gram phrases the model uses. We utilize Dist-2 and Dist-3 as proposed in [39] to evaluate the informative diversity of the generated lyrics. It is calculated as:

$$Dist - n = m_{n\_U}/m_n \tag{17}$$

where n denotes either bi-gram or tri-gram, while m_{n_U} and m_n denote the number of unique n-gram phrases and the number of total n-gram phrases in all the generated lyrics respectively.

Relevance. Since one piece of music can correspond to several versions of lyrics, referring to the strategy in [25], we utilize the cosine similarity between the deep coupled embedding of the music and that of the generated lyrics as their relevance score, rather than the text similarity between the generated lyrics with ground-truth.

BLEU. We believe BLEU can partly indicate the local text-quality, although it might not be particularly persuasive for evaluating creative text generation tasks since it evaluates how likely the generated result is the translation of the ground-truth.

Application-oriented Satisfaction. This is a binary YES/NO metric. Although not counted in the quantitative final overall quality. It measures the models' usability of whether the model can generate lyrics under given constraints of keywords and a vocabulary set.

4.4 Subjective Evaluation

To a certain extent, subjective evaluation is more significant and better examines the performance of the lyrics generation model because as a subjective creation task, there is so far not a perfectly suitable metric to evaluate how "good" the lyrics writing is in a quantizable and describable manner. Therefore, we conduct a subjective evaluation in Amazon Mechanical Turk with 60 native English speakers. The procedure is as follows:

(1) A participant reads the lyrics generated from the same music by the five compared models and rates the lyrics on a 0-5 scale regarding their fluency, coherence, meaningfulness and poetic aesthetics respectively, with 5 being the best.

(2) A participant listens to a singing sample synthesized with the music and lyrics generated by the 5 models, and rates the lyrics on a 0-5 scale regarding the syllable alignment to the melody and their relevance respectively.

To filter out the potentially invalid reviews, we record the time points of the operations including playing and stopping each song, and choice making. Also, we deliberately insert badly aligned lyrics to the melody as negative samples.

4.5 Results

The results of both the subjective and objective evaluation are shown in Table 2, and an example generated by our final released system is shown in Figure 5. The results of the subjective evaluation show that our proposed system surpasses the baseline models in all the crucial factors of a good lyrics sample for learning language through singing.

The objective evaluation indicates lyrics generated by our proposed system achieve a desirable balance in the three quantitative text quality metrics and are the most relevant to the input music: The informative density of MT and PRE-M2L-B is low, meaning they often repeat the same words, and lack creativity. Furthermore, the Dist-2 score of the encoder–decoder model is extremely high (0.98), meaning that it is picking words randomly. Furthermore, the novelty scores of SeqGAN are low, likely due to overfitting to confuse the discriminators. Finally, only our proposed two models satisfy the constraints of language learning.

The ablation study shows that the syllable awareness mechanism in generator and the syllable discriminator help syllable alignment while the text-quality discriminator improves the relevance and coherence. Also, compared with the PRE-M2L-B model, the POST-M2L-B model performs better in all measures.

5 DISCUSSION

We compare the effectiveness of two reasonable structures of lyrics generator: one decodes the deep coupled embedding of input music to lyrics, while the other generates lyrics from scratch. In both structures, the best match is selected out of all candidates. We found that a decoder does not improve the relevance between generated lyrics and the given music because in both subjective and objective evaluations, generating lyrics from scratch gives a higher relevance score than decoding embeddings. This finding might be attributed to the relatively

Table 2. Results of both (a) objective and (b) subjective evaluations.

a. Objective Evaluation	Novelty-2	Novelty-3	Dist-1	Dist-2	BLEU-1	BLEU-2	BLEU-3	Relevance	Overall	App Satisfaction
MT	0.390	0.190	9.16e-2	0.254	0.22	1.09e-2	3.71e-7	66.52	0.538	**N**
ED	6.02e-3	1.07e-3	4.81e-2	0.980	0.03	2.55e-5	1.53e-6	60.67	0.187	**N**
SeqGAN	5.21e-2	8.5e-2	0.132	0.417	0.17	9.61e-3	5.25e-4	68.22	0.554	**N**
PRE-M2L-B	0.410	0.213	6.82e-3	0.135	0.26	1.40e-2	3.30e-4	66.13	0.590	**Y**
POST-M2L	1.97e-4	0.0	0.164	0.363	1.92e-3	3.53e-6	~0.0	59.03	0.159	**N**
POST-M2L-S	8.09e-2	0.330	0.115	0.251	3.91e-2	3.47e-3	4.31e-4	68.84	0.482	**Y**
POST-M2L-T	0.224	0.144	0.150	0.473	3.67e-2	2.28e-3	6.61e-5	74.82	0.464	**Y**
POST-M2L-B	0.169	0.158	0.108	0.406	0.21	1.08e-2	3.36e-4	70.59	0.601	**Y**

b. Subjective Evaluation	Fluency	Coherence	Meaningfulness	Aesthetics	Syllable Alignment	Relevance	Overall
MT	**2.43**	**2.40**	**2.60**	**2.47**	**2.87**	**2.87**	**2.61**
ED	**2.87**	**2.83**	**2.6**	**2.87**	**3.43**	**2.77**	**2.90**
SeqGAN	**2.83**	**2.80**	**2.63**	**2.67**	**3.23**	**3.00**	**2.86**
PRE-M2L-B	**3.67**	**3.38**	**3.46**	**3.40**	**4.00**	**3.67**	**3.59**
POST-M2L-B	**4.37**	**4.47**	**4.37**	**4.17**	**4.53**	**4.50**	**4.40**

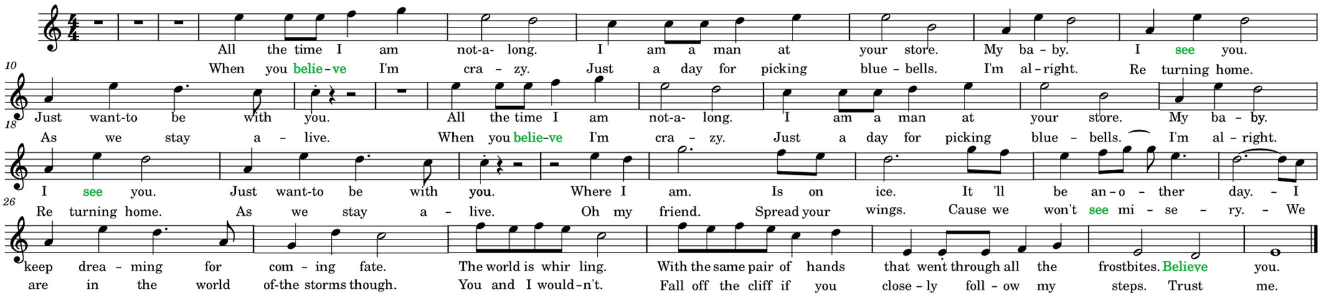
All the time I am not-a- long. I am a man at your store. My ba - by. I see you.
When you belie-ve I'm cra - zy. Just a day for picking blue - bells. I'm al-right. Re turning home.
10
Just want-to be with you. All the time I am not-a- long. I am a man at your store. My ba - by.
As we stay a - live. When you belie-ve I'm cra - zy. Just a day for picking blue - bells. I'm al - right.
18
I see you. Just want-to be with you. Where I am. Is on ice. It 'll be an-o - ther day. - I
Re turning home. As we stay a - live. Oh my friend. Spread your wings. Cause we won't see mi - se - ry. - We
26
keep drea - ming for com - ing fate. The world is whir ling. With the same pair of hands that went through all the frostbites. Believe you.
are in the world of-the storms though. You and I would-n't. Fall off the cliff if you close - ly foll - ow my steps. Trust me.

Figure 5: Demo lyrics generated in Automatic Mode (first line) and Interactive Mode (second line). The input music is a children song called "Fireflies fly" while the keywords is set to be ["see", "believe"] and the words are constrained within a children song corpus. Due to page limit, accompaniment is omitted in the score. A full demo with synthesized vocal singing can be found in the following link: (for review purpose please refer to the uploaded attachments.)

small size of the paired music–lyrics dataset because within the limited data, it is difficult to encode highly abstract information of stylistic relevance and then to decode that information to another modality.

It might be more elegant to utilize an end-to-end architecture, allowing the system to learn everything including the music structure, the keywords and the vocabulary constraints purely from data, thus bypassing the pipeline design and hard-coded rules. However, more subdivided types of datasets and more powerful architecture are needed to tackle such a setting. For example, a paired music–lyrics dataset of children songs and disentanglement of lyrics features to style- and vocabulary-related parts could realize this goal. Considering that the keywords and vocabulary constraints are mandatory requirements of the generated lyrics, we believe the current solution is still a reasonable starting point.

According to the feedback from linguistics, language teachers and song producers, although the generated lyrics is well-aligned to the melody and surprisingly creative, they could be improved in language quality. We notice that while the generated lyrics have strong in-line coherence within each sentence, their global topic coherence can be improved. Although the interactive mode can improve the topic coherence to some extent, exploitation of controlled language generation techniques is of high priority in our future work. Instead of BERT, a non-narrative and non-descriptive text embedding model might fit ALG tasks better in modeling stylistic features.

Automatic analysis of deeper music structure may help the topic transition in lyrics generation as well. A music-structure-aware model may be able to better guide the topic transition of lyrics generation; for example, by learning and extracting hierarchical features of dynamic intensity, finding sub-topics of music sections and the global topic, and so on.

6 CONCLUSION

As the first attempt to generate lyrics from multi-channel music for language learning, we propose a SeqGAN based lyrics generator, with a music structure analyzer, which overcomes the limitation of input types; a deep coupled music–lyrics embedding model, which conditions the lyrics generator with stylistic features of the input music; and Polisher, an application-oriented constraints applier to incorporate personalized language learning requirements. Both objective and subjective evaluations indicate the effectiveness of our proposed system to generate lyrics that are meaningful, novel, singable and usable.

ACKNOWLEDGMENTS

This project was funded in part by research grant R-252-000-B75-112 from the Ministry of Education, Singapore.

REFERENCES

[1] Dania Murad, Riwu Wang, Douglas Turnbull, and Ye Wang. 2018. SLIONS: A Karaoke Application to Enhance Foreign Language Learning. In 2018 ACM Multimedia Conference on Multimedia Conference. ACM, 1679–1687.

[2] Dwayne Engh. 2013. Why Use Music in English Language Learning? A Survey of the Literature. English Language Teaching 6, 2 (2013), 113–127.

[3] Douglas Fisher. 2001. Early language learning with and without music. Reading Horizons 42, 1 (2001), 39.

[4] Arla J Good, Frank A Russo, and Jennifer Sullivan. 2015. The efficacy of singing in foreign-language learning. Psychology of Music 43, 5 (2015), 627–640.

[5] Suzanne L Medina. 1990. The Effects of Music upon Second Language Vocabulary Acquisition. (1990).

[6] Susan Bergman Miyake. 2004. Pronunciation and music. Sophia Junior College Faculty Bulletin 20, 3 (2004), 80.

[7] Andrés Roberto Rengifo. 2009. Improving pronunciation through the use of karaoke in an adult English class. Profile Issues in Teachers Professional Development 11 (2009), 91–106.

[8] Wanda T Wallace. 1994. Memory for music: Effect of melody on recall of text. Journal of Experimental Psychology: Learning, Memory, and Cognition 20, 6 (1994), 1471.

[9] Judith Weaver Failoni. 1993. Music as Means To Enhance Cultural Awareness and Literacy in the Foreign Language Classroom. Mid-Atlantic Journal of Foreign Language Pedagogy 1 (1993), 97–108.

[10] Sameerchand Pudaruth, Sandiana Amourdon and Joey Anseline. 2014. Automated generation of song lyrics using CFGs. 2014 Seventh International Conference on Contemporary Computing (IC3). IEEE, 2014. 613-616.

[11] Hieu Nguyen and Brian Sa. 2019. Rap lyric generator. New York, USA. 1-3. 2009.

[12] Potash, Peter, Alexey Romanov, and Anna Rumshisky. 2015. Ghostwriter: Using an LSTM for automatic rap lyric generation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 1919-1924.

[13] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. Dopelearning: A computational approach to rap lyrics generation. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 195-204.

[14] Gabriele Barbieri, Franc¸ois Pachet, Pierre Roy, and Mirko Degli Esposti. 2012. Markov Constraints for Generating Lyrics with Style. Ecai 242 (2012). 115-120.

[15] Xing Wu, Zhikang Du, Yike Guo, and Hamido Fujita. 2019. Hierarchical attention based long short-term memory for Chinese lyric generation. Applied Intelligence 49.1 (2019): 44-52.

[16] Sung-Hwan Son, Hyun-Young Lee, Gyu-Hyeon Nam, and Seung-Shik Kang. 2019. Korean Song-lyrics Generation by Deep Learning. Proceedings of the 2019 4th International Conference on Intelligent Information Technology. 96-100.

[17] Asir Saeed, Suzana Ili´c, and Eva Zangerle. 2019. Creative GANs for generating poems, lyrics, and metaphors. arXiv preprint arXiv:1909.09534 (2019).

[18] Jie Wang, and Xinyan Zhao. 2019. Theme-aware generation model for Chinese lyrics. arXiv preprint arXiv:1906.02134 (2019).

[19] Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. 2018. A melody-conditioned lyrics language model. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. 2018: 163-172.

[20] Xu Lu, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A Syllable-Structured, Contextually-Based Conditionally Generation of Chinese Lyrics. In PRICAI. 257-265.

[21] Chisa Nakamura, and Takehisa Onisawa. 2009. Music/lyrics composition system considering user's image and music genre. 2009 IEEE International Conference on Systems, Man and Cybernetics. 1764-1769.

[22] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In International conference on machine learning 2017. 933-941.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[24] Ananth Ramakrishnan A, Sankar Kuppan, and Sobha Lalitha Devi. 2009. Automatic generation of Tamil lyrics for melodies. In Proceedings of the workshop on computational approaches to linguistic creativity. 40-46.

[25] Bei Liu, Jianlong Fu, Makoto P Kato, and Masatoshi Yoshikawa. 2018. s narrative description: Generating poetry from images by multi-adversarial training. In 2018 ACM Multimedia Conference on Multimedia Conference. 783-791.

[26] Hugo Gonçalo Oliveira. 2015. Tra-la-Lyrics 2.0: Automatic Generation of Song Lyrics on a Semantic Domain. Journal of Artificial General Intelligence 6. 87–110.

[27] Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. 2014. Modeling Structural Topic Transitions for Automatic Lyrics Generation. In PACLIC 28. 422–431.

[28] Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A hierarchical attention based seq2seq model for Chinese lyrics generation. In Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2019). 279–288.

[29] Rongsheng Zhang, Xiaoxi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2020. Youling: an AI-assisted Lyrics Creation System. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 85-91.

[30] Nikola I. Nikolovy, Eric Malmiz, Curtis G. Northcuttx, and Loreto Parisi. 2020. Rapformer: Conditional Rap Lyrics Generation with Denoising Autoencoders. Proceedings of the 13th International Conference on Natural Language Generation. 2020. 360-373.

[31] Vechtomova, Olga, Gaurav Sahu, and Dhruv Kumar. Generation of lyrics lines conditioned on music audio clips. arXiv preprint arXiv:2009.14375 (2020).

[32] Enrique Manjavacas, Mike Kestemont, and Folgert Karsdorp. 2019. Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In Proceedings of the 12th International Conference on Natural Language Generation. 301–310.

[33] Chen, Yihao, and Alexander Lerch. Melody-Conditioned Lyrics Generation with SeqGANs. arXiv preprint arXiv:2010.14709 (2020).

[34] Gurunath Reddy M, Yi Yu, Florian Harscoët, Simon Canales, Suhua Tang. Automatic Neural Lyrics and Melody Composition. arXiv preprint arXiv:2011.06380 (2020).

[35] Rongsheng Zhang, et al. 2020. Youling: an AI-Assisted Lyrics Creation System." Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 85-91.

[36] Yi Yu and Simon Canales. 2019. Conditional LSTM-GAN for melody generation from lyrics. arXiv preprint arXiv:1908.05551 (2019).

[37] Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1. 63–70.

[38] Zhen Xu, Bingquan Liu, Baoxun Wang, SUN Chengjie, Xiaolong Wang, Zhuoran Wang, and Chao Qi. 2017. Neural Response Generation via GAN with an Approximate Embedding Layer. In EMNLP. 628–637.

[39] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 110–119.

[40] Yichao Zhou, Wei Chu, Sam Young, and Xin Chen. Bandnet: A neural network-based, multi-instrument Beatles-style midi music composition machine. arXiv preprint arXiv:1812.07126, 2018.

[41] Lantao Yu, Weinan Zhang, JunWang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In ACL. 311–318.

[43] A. Singhi and D. G. Brown. Are poetry and lyrics all that different? In Proceedings of the 15th International Conference on Music Information Retrieval, ISMIR'14, pages 471–476, 2014.

[44] Hisar Maruli Manurung. 1999. A chart generator for rhythm patterned text. In Proceedings of the First International Workshop on Literature in Cognition and Computer. 15–19.

[45] Hugo Oliveira. 2009. Automatic generation of poetry: an overview. Universidade de Coimbra (2009).

[46] Hugo Gon¸calo Oliveira. 2012. PoeTryMe: a versatile platform for poetry generation. Computational Creativity, Concept Invention, and General Intelligence 1 (2012), 21.

[47] Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. 2013. i, Poet: Automatic Chinese Poetry Composition through a Generative Summarization Framework under Constrained Optimization.. In IJCAI. 2197–2203.

[48] Long Jiang and Ming Zhou. 2008. Generating Chinese couplets using a statistical MT approach. In COLING. 377–384.

[49] Jing He, Ming Zhou, and Long Jiang. 2012. Generating Chinese Classical Poems with Statistical Machine Translation Models.. In AAAI.

[50] Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic Analysis of Rhythmic Poetry with Applications to Generation and Translation. In EMNLP. 524–533.

[51] Xingxing Zhang and Mirella Lapata. 2014. Chinese Poetry Generation with Recurrent Neural Networks.. In EMNLP. 670–680.

[52] Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating Chinese Classical Poems with RNN Encoder-Decoder. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. 211–223.

[53] Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. 2016. Chinese song iambics generation with neural attention-based model. arXiv:1604.06274.

[54] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In EMNLP. 3143–3153.

[55] Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. 2018. Stylistic Chinese poetry generation via unsupervised style disentanglement. In EMNLP. 3960–3969.

[56] Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. Jiuge: A human-machine

collaborative Chinese classical poetry generation system. In ACL: System Demonstrations. 25–30.

[57] Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. 2017. Flexible and creative Chinese poetry generation using neural memory. In ACL 55. 1364–1373.

[58] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Zonghan Yang. 2018. Chinese poetry generation with a working memory mode. In IJCAI-18. 4553–4559.

[59] Zhe Wang, Wei He, Hua Wu nad Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. In COLING 2016. 1051–1060.

[60] Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating Topical Poetry. In EMNLP. 1183–1191.

[61] Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an Interactive Poetry Generation System. ACL (2017), 43–48.

[62] Jack Hopkins and Douwe Kiela. 2017. Automatically Generating Rhythmic Verse with Neural Networks. In ACL, Vol. 1. 168–178.

[63] Randolph B Johnson, David Huron, and Lauren Collister. Music and lyrics interactions and their influence on recognition of sung words: an investigation of word frequency, rhyme, metric stress, vocal timbre, melisma, and repetition priming. Empirical Musicology Review, 9(1):2–20, 2013.

[64] Eric Nichols, Dan Morris, Sumit Basu, and Christopher Raphael. 2009. Relationships between lyrics and melody in popular music. In ISMIR 2009. 471–476.

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

[66] Takamitsu shimazaki. 2015. A study book for lyricists. Rittor Music. Tokyo, Japan.