SongMASS: Automatic Song Writing with Pre-training and Alignment Constraint

Zhonghao Sheng[1,*], Kaitao Song[2,*], Xu Tan[†3], Yi Ren[4], Wei Ye[1], Shikun Zhang[1], Tao Qin[3]
Peking University[1], Nanjing University of Science and Technology[2], Microsoft Research Asia[3], Zhejiang University[4]
{zhonghao.sheng,wye,zhangsk}@pku.edu.cn, kt.song@njust.edu.cn, {xuta, taoqin}@microsoft.com, rayeren@zju.edu.cn

Abstract

Automatic song writing aims to compose a song (lyric and/or melody) by machine, which is an interesting topic in both academia and industry. In automatic song writing, lyric-to-melody generation and melody-to-lyric generation are two important tasks, both of which usually suffer from the following challenges: 1) the paired lyric and melody data are limited, which affects the generation quality of the two tasks, considering a lot of paired training data are needed due to the weak correlation between lyric and melody; 2) Strict alignments are required between lyric and melody, which relies on specific alignment modeling. In this paper, we propose SongMASS to address the above challenges, which leverages masked sequence to sequence (MASS) pre-training and attention based alignment modeling for lyric-to-melody and melody-to-lyric generation. Specifically, 1) we extend the original sentence-level MASS pre-training to song level to better capture long contextual information in music, and use a separate encoder and decoder for each modality (lyric or melody); 2) we leverage sentence-level attention mask and token-level attention constraint during training to enhance the alignment between lyric and melody. During inference, we use a dynamic programming strategy to obtain the alignment between each word/syllable in lyric and note in melody. We pre-train SongMASS on unpaired lyric and melody datasets, and both objective and subjective evaluations demonstrate that SongMASS generates lyric and melody with significantly better quality than the baseline method without pre-training or alignment constraint.

1 Introduction

Automatic song writing is an interesting and challenging task in both research and industry. Two most important tasks in automatic song writing are lyric-to-melody generation (L2M) (Bao et al. 2019; Yu and Canales 2019; Lee, Fang, and Ma 2019) and melody-to-lyric generation (M2L) (Watanabe et al. 2018; Lu et al. 2019; Lee, Fang, and Ma 2019). L2M and M2L can be regarded as sequence to sequence learning tasks and can be modeled by the techniques in natural language processing since both melody and lyric can be represented as discrete token sequence. However, L2M and M2L have distinctive characteristics that differ them from other sequence to sequence learning tasks: 1) lyric and melody are weakly correlated in L2M and M2L while in other tasks (Bahdanau, Cho, and Bengio 2014; Rush, Chopra, and Weston 2015), source and target sequences are strongly correlated in semantics; 2) one word or syllable in lyric always strictly aligns with one or more notes in melody, while other tasks do not require strict alignments. An example of aligned lyric and melody piece is shown in Figure 1.

Paired Aligned Data :

Lyric	Another				day	has	gone	I'm		still	alone	
Pitch	R	G3	E4	D4	C4	B3	C4	R	E4	C4	B3	C4
Duration	$\frac{7}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{5}{16}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{3}{16}$	$\frac{1}{16}$	$\frac{5}{16}$

Figure 1: A song fragment "Another day has gone, I'm still all alone" with its melody. The table shows the alignment of the lyric and melody (pitch and duration).

The above distinctive characteristics throw several challenges in modeling L2M and M2L: 1) They require large amount of paired melody and lyric data to learn the mapping relationship between lyric and melody due to weak correlation. However, it is difficult to collect such large amount of paired data, and thus both tasks suffer from limited paired data; 2) They need additionally generate strict alignments between word/syllable in lyric and note in melody, and thus how to model the alignments well is critical to ensure the generation quality of lyric and melody. Previous works (Bao et al. 2019; Li et al. 2020; Watanabe et al. 2018; Lee, Fang, and Ma 2019) on L2M and M2L have not considered the scenario of limited paired data, and only leverage some greedy decisions for lyric and melody alignment, which cannot well address these challenges. In this paper, we propose SongMASS, an automatic song writing system for L2M and M2L, which addresses the first challenge with masked sequence to sequence pre-training and the second challenge with attention based alignment constraint.

*The first two authors contribute equally to this work.
†Corresponding Author: Xu Tan, xuta@microsoft.com.

Specifically, to handle the challenges of limited paired data, we leverage self-supervised pre-training on large amount of unpaired lyric and melody data. Since L2M and M2L are both sequence to sequence learning tasks, we adopt masked sequence to sequence pre-training (MASS) (Song et al. 2019), which is a popular pre-training method by masking a segment in the input sequence and predicting this segment in the output using an encoder-decoder framework. However, simply using original MASS in L2M and M2L cannot well handle the long lyric and melody sequence in song level and the diversity between lyric and melody modality. Therefore, we introduce two extensions on MASS: 1) Instead of masking in a single sentence in original MASS, we design a song-level masked pre-training strategy to capture longer contextual information, since music usually has repeat structure and relies on long context. 2) Unlike original MASS, we use separate encoder-decoder for lyric-to-lyric and melody-to-melody masked pre-training since they are in different modalities. However, separate training of lyric-to-lyric and melody-to-melody cannot ensure to learn a shared latent space between lyric and melody and thus could harm the transformation between them. Therefore, we add supervised training with paired lyric and melody data to guide the pre-training towards learning a shared latent representation between lyric and melody modality.

To address the challenges of lyric-melody alignment, we propose to align the word/syllable in lyric and note in melody based on the encoder-decoder attention. Due to long melody and lyric sequence in a song, we split the alignment into sentence level and token level. To ensure sentence-level alignment, we constrain each sentence in target sequence to only attend to the corresponding sentence in source sequence during training and inference. We add an additional [SEP] token as the sentence boundary in each sequence, and during inference, once a [SEP] token is predicted in the target side, we switch the attention to the next source sentence. For token-level alignment, we add constraints on the attention matrix using the ground-truth alignment in the paired training data during training, and use a dynamic programming algorithm on the generated attention matrix during inference to obtain the final alignments.

The contributions of our method are as follows:

- We are the first to leverage pre-training to address the low-resource challenge on L2M and M2L, by introducing two extensions on MASS including song-level masked pre-training and using supervised pre-training to guide the separate encoder-decoder of lyric and melody to the same latent space.
- To handle the alignment between word/syllable in lyric and note in melody, we design the attention-based sentence-level and token-level alignment constraints and a dynamic programming algorithm to obtain precise alignments.
- Experimental results with objective and subjective evaluations demonstrate that SongMASS significantly improves the quality of lyric and melody generation with the help of pre-training and alignment constraint.

2 Background

Automatic Song Writing Automatic song writing usually covers several tasks including lyric generation (Malmi et al. 2015), melody generation (Zhu et al. 2018), lyric-to-melody generation (L2M) (Choi, Fazekas, and Sandler 2016; Yu and Canales 2019) and melody-to-lyric generation (M2L) (Bao et al. 2019; Li et al. 2020). In this work, we focus on L2M and M2L. Choi, Fazekas, and Sandler (2016); Yu and Canales (2019) generated melody conditioned on the lyrics with RNN-based language model. Lee, Fang, and Ma (2019); Bao et al. (2019) used sequence to sequence model for L2M and M2L. However, these works on L2M and M2L usually only used limited paired data, without leveraging large-scale unpaired data. On the other hand, some works only focused on L2M and M2L on the sentence level, assuming there are strict one-to-one mapping in the training data, which cannot compose a complete song. Some other works (Bao et al. 2019; Watanabe et al. 2018) explicitly predicted the alignment flag (e.g., whether switches to next word/syllable when predicting notes or not) in the model, with a greedy decision in the word/syllable or note level, which is not flexible and fail to capture the global alignment in the whole sentence. In this paper, we propose SongMASS, which uses sequence to sequence pre-training method to leverage the unpaired lyric and melody data, and attention-based alignment constraints for global and precise lyric-melody alignment.

Pre-training Methods Pre-trained language models (*e.g.*, BERT (Devlin et al. 2019), GPT (Radford et al. 2018), XLNet (Yang et al. 2019), MASS (Song et al. 2019) and *etc*) have achieved significant progress in natural language processing. They usually employ specific self-supervised tasks and pre-train on large-scale unlabeled data corpus to improve the understanding and generation capability. MASS (Song et al. 2019) is the first and one of most successful pre-training methods for sequence to sequence learning tasks, and several pre-training methods such as BART (Lewis et al. 2019) and T5 (Raffel et al. 2019) are also proposed to handle this kind of task. In this paper, we build our pre-training method upon MASS considering its popularity for sequence to sequence learning tasks and suitability for different modalities. Given a sequence from the unpaired sentence corpus, MASS randomly replaces a segment of tokens with mask tokens and takes the masked sequence as the encoder input and predicts the masked segment in the decoder. We leverage the basic idea of MASS and extend it with several improvements to address the distinctive challenges in the pre-training of L2M and M2L.

Alignment Modeling Alignment modeling builds the correlation between the tokens in source and target sequences, which plays an important role in sequence to sequence tasks. In L2M and M2L tasks, previous works usually used greedy alignment mechanisms to handle the correlation between lyric and melody. For example, Watanabe et al. (2018) used the Needleman-Wunsch algorithm (Needleman and Wunsch 1970) to count the alignment of lyric and melody. Bao et al.

(2019) predicted how many syllables in the predicting word given current note input. However, these greedy alignment strategies cannot provide flexible and global alignments in the sentence level. In other sequence to sequence learning tasks like neural machine translation, Bahdanau, Cho, and Bengio (2014); Luong, Pham, and Manning (2015) introduced attention mechanism to learn the relationship between source and target languages. In this paper, we leverage the attention mechanism to build the global and soft alignment between lyric and melody, and finally design a dynamic programming method to obtain the strict alignment between word/syllable and note.

3 Method

3.1 System Overview

The overall architecture of SongMASS for L2M and M2L is shown in Figure 2, which adopts the Transformer (Vaswani et al. 2017) based encoder-decoder framework. We employ separated encoders and decoders for lyric and melody respectively due to the large diversity between lyric and melody. To leverage the knowledge from large-scale unlabeled lyrics or melodies, we perform MASS pre-training for lyric-to-lyric and melody-to-melody in our framework. We pre-train our model in song level to better capture long contextual information for lyric or melody sequence, and incorporate supervised learning (lyric-to-melody and melody-to-lyric) into our pre-training to learn a shared latent space between different modalities. To learn the alignment between word/syllable in lyric and note in melody, we further leverage sentence-level constraint and token-level constraint into our model to guide the alignment between lyric and melody. We use a dynamic programming strategy to obtain the final strict alignment between the lyric and melody. In below, we describe the details of our pre-training methods and alignment strategies in SongMASS.

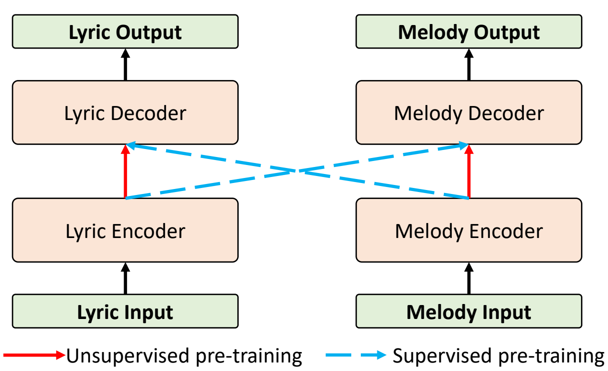


Figure 2: The overall architecture of our SongMASS framework. The red line means unsupervised pre-training on lyric-to-lyric or melody-to-melody. The blue dotted line is supervised pre-training on lyric-to-melody or melody-to-lyric.

3.2 Pre-training Methods

In this subsection, we introduce our pre-training methods, including song-level MASS pre-training to capture long contextual information from the whole song, and supervised pre-training to learn a shared latent space between lyric and melody modality.

Song-Level MASS Pre-training As mentioned in Section 2, the original MASS pre-training is designed to help the model understand and generate sequence in sentence level. However, instead of using sentence-level information, we expect model to capture long contextual information in song level (*i.e.*, the whole song). Therefore, we introduce the song-level MASS pre-training to address this issue.

Denote $\mathcal{X}'$ and $\mathcal{Y}'$ as the corpus of unpaired lyrics and melodies in the song level respectively. For any $x \in \mathcal{X}'$ and $y \in \mathcal{Y}'$, we split the song-level sequence into multiple sentences and insert a special token $[\mathrm{SEP}]$ in the boundary of adjacent sentences. For every sentence from the song-level sequence, we perform the same mask strategy as in the original MASS (as mentioned in Section 2). The details of the masking strategy are shown in Figure 3. The encoder takes the masked song-level sequence as input and the decoder predicts masked fragments corresponding to all the sentences in this song. The formulation of song-level MASS is as follows:

$$
\begin{aligned}
L(\mathcal{X};\theta^{enc},\theta^{dec}) &= \sum_{x\in\mathcal{X}}\sum_{i=1}^{S}\log P(x^{u_i:v_i}|x^{\setminus\{u_i:v_i\}_{i=1}^{S}};\theta^{enc},\theta^{dec}) \\
&= \sum_{x\in\mathcal{X}}\sum_{i=1}^{S}\log\prod_{t=u_i}^{v_i}P(x_t^{u_i:v_i}|x_{<t}^{u_i:v_i},x^{\setminus\{u_i:v_i\}_{i=1}^{S}};\theta^{enc},\theta^{dec}),
\end{aligned} \tag{1}
$$

where S represents the number of sentences in sequence x, $x^{\setminus\{u_i:v_i\}_{i=1}^{S}}$ represents the masked song-level sequence, and $x^{u_i:v_i}$ represents the masked segment in the i-th sentence. We define θ_x^{enc}, θ_x^{dec}, θ_y^{enc}, θ_y^{dec} as the parameters of lyric encoder, lyric decoder, melody encoder and melody decoder. The loss for lyric-to-lyric generation is $L_x = L(\mathcal{X};\theta_x^{enc},\theta_x^{dec})$ and the loss for melody-to-melody is $L_y = L(\mathcal{Y};\theta_y^{enc},\theta_y^{dec})$.

Supervised Pre-training Although MASS pre-training can help the model understand and generate lyric and melody respectively, the model cannot learn to generate melody from lyric and lyric from melody. What is worse is that the encoder-decoder models for lyric and melody cannot align in the same latent space and may deviate from each other, which will harm the transformation between lyric and melody. To prevent them from deviating and help align them together, we leverage the supervised training on lyric-melody paired data in the pre-training process. Given paired corpus $(\mathcal{X},\mathcal{Y})$, the loss of the supervised pre-training is

$$
\begin{aligned}
L(\mathcal{X},\mathcal{Y};\theta^{enc},\theta^{dec}) &= \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})}\log P(y|x;\theta^{enc},\theta^{dec}) \\
&= \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})}\log\prod_{t=1}^{|y|}P(y_t|y_{<t},x;\theta^{enc},\theta^{dec}).
\end{aligned} \tag{2}
$$

The supervised pre-training is applied on both lyric-to-melody and melody-to-lyric generation. The loss for lyric-

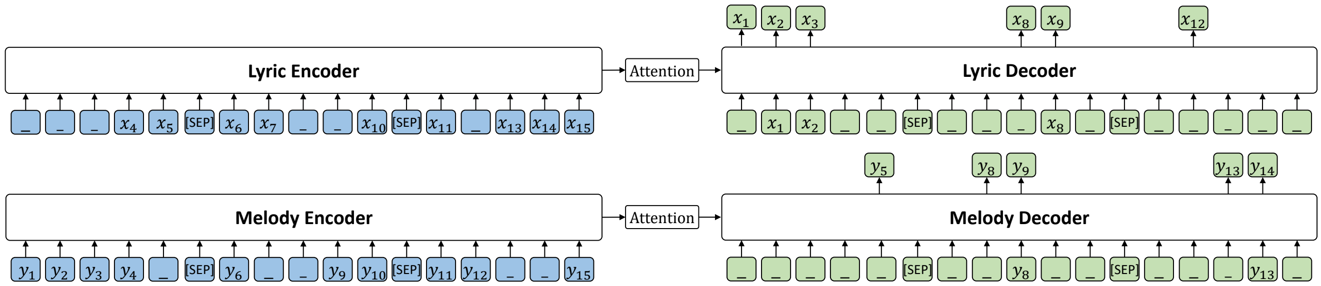


Figure 3: The song-level MASS pre-training.

to-melody generation is $L_{xy} = L(\mathcal{X}, \mathcal{Y}; \theta_x^{enc}, \theta_y^{dec})$ and the loss for melody-to-lyric is $L_{yx} = L(\mathcal{Y}, \mathcal{X}; \theta_y^{enc}, \theta_x^{dec})$.

Finally, the total pre-training loss is

$$L_{pt} = L_x + L_y + L_{xy} + L_{yx}, \tag{3}$$

where L_x and L_y are the unsupervised MASS pre-training loss described in Equation 1, and L_{xy} and L_{yx} are the supervised pre-training loss. During fine-tuning, we only use L_{xy} for lyric-to-melody generation and L_{yx} for melody-to-lyric generation.

3.3 Alignment Strategy

In this subsection, we describe how to learn the alignment between lyric and melody in SongMASS. The basic idea is to leverage the encoder-decoder attention to infer the alignment between each word/syllable in lyric and note in melody. In order to extract strict alignment, we explicitly add constraints on the attention to learn effective attention patterns during training and inference. Due to the long song-level sequence, we divide our alignment strategy into sentence-level constraint and token-level constraint.

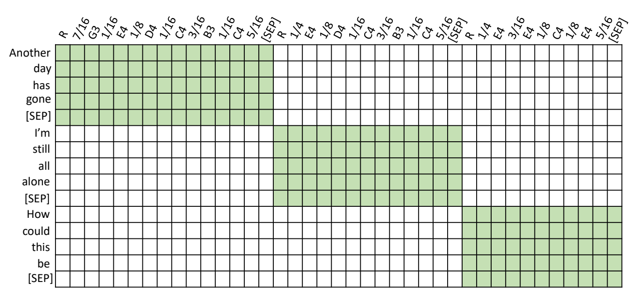


Figure 4: Sentence-level attention mask.

Sentence-Level Constraint A song consists of multiple lyric sentences and melody phrases. In song writing, the given lyrics and melodies are naturally grouped into sentences or phrases. The lyric sentences and the melody phrases are strictly aligned in the training data. So we constrain that each sentence (lyric sentence or melody phrase) in the target sequence can only attend to the corresponding sentence (melody phrase or lyric sentence) in source sequence. Specifically, we apply a sentence-level constraint mask on the encoder-decoder attention. We denote y_i and x_j as the i-th token in the target sequence and j-th token in the source sequence respectively. We assume the representations of x_j and y_i from the previous Transformer layer as h_j^{enc} and h_i^{dec}. So the attention score between target token y_i and source token x_j is computed as:

$$f(i,j) = \frac{h_i^{dec} W^Q (h_j^{enc} W^K)^T}{\sqrt{d_z}} + M(i,j), \tag{4}$$

$$A(i,j) = \frac{\exp f(i,j)}{\sum_j \exp f(i,j)} \tag{5}$$

where $A(i,j)$ calculates the attention score between the y_i and x_j. W^Q,$W^K \in \mathbb{R}^{d_z \times d_z}$ are model parameters, and d_z is the dimension of the hidden representations. $M(i,j)$ represents the mask element between y_i and x_j, whose value is set as follows:

$$M(i,j) = \begin{cases} 0 & \mathrm{ID(y_i)} = \mathrm{ID(x_j)} \\ -\infty & \mathrm{ID(y_i)} \neq \mathrm{ID(x_j)} \end{cases}. \tag{6}$$

where $\mathrm{ID(x)}$ gets the index of the sentence that the token x belongs to. M is used as our sentence-level alignment constraint, as shown in Figure 4. Besides, we insert a special token [SEP] in the sentence boundary of the input and output sequences as shown in Figure 3, to help the model better capture the sentence boundary information and identify which sentence of the input sequence should to be attended to at the current step during the inference stage. Benefiting from such design, we guarantee the number of sentences in the generated sequences is consistent with the input sequence.

Token-Level Constraint Unlike sentence-level alignment, the alignment choices between each word/syllable and note are more flexible. Therefore, we propose a regularization term on the encoder-decoder attention during the training on paired data, and apply a dynamic programming algorithm on the attention matrix to obtain the final strict alignment during inference. We expect the attention weight between y_i and x_j to follow:

$$u(i,j) = \begin{cases} \frac{1}{T} & \text{if } y_i \text{ is aligned to } x_j, \\ 0 & \text{Otherwise,} \end{cases} \tag{7}$$

where T is the number of tokens in the source sentence that y_i is aligned to. As shown in Figure 5, we add a regularization term to constrain the attention weights:

$$L_{att} = \frac{1}{N * M} \sum_{i=1}^{M} \sum_{j=1}^{N} \|A(i,j) - u(i,j)\|_2, \tag{8}$$

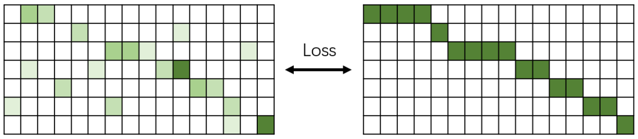


Figure 5: Token-level guided attention mask.

where $\|\cdot\|$ represents L2-Norm. N and M are the number of tokens in the source and target sentence respectively. Finally, the loss function is:

$$L = L_{pt} + \alpha \cdot L_{att}, \tag{9}$$

where α is the hyper-parameter of L_{att}, and L_{pt} is the pre-training loss defined in Equation 3.

When all tokens in a sentence are generated and the attention matrix A is obtained, we infer the global alignment by applying a dynamic programming algorithm as shown in Algorithm 1. We consider the following cases: a target token is aligned to one or many source tokens, and a source token is aligned to one or many target tokens. For the first case, as shown in Line 7 - 12 in Algorithm 1, we search for a k that the alignment between y_i and $x_{[K+1:j]}$ reaches the highest score, which is calculated by summing all the corresponding attention weights. Similarly for the second case, as shown in Line 13 - 18 in Algorithm 1, we search for a k that the alignment between $y_{[k+1:i]}$ and $x_{[j]}$ reaches the highest score. We take the average weights as score in the second case, since the weights of the target sequence dimension are not normalized like that of the source sequence. We choose the higher score of the two cases and save the aligned pair.

Algorithm 1 DP for Melody-Lyric Alignment

1: **Input**: Attention matrix $A \in \mathbb{R}^{N\times M}$, score matrix $F \in \mathbb{R}^{(N+1)\times(M+1)}$, path matrix Path, source sequence x and target sequence y. N and M are the length of x and y.
2: **Output**: The aligned pairs list D.
3: **Initialize**: F is initialized as $-\infty$. Path is initialized as an empty matrix with a shape of $(N+1)\times(M+1)$.
4: $F[0][0] = 0$
5: **for** $i = 1$ to T **do**
6:   **for** $j = 1$ to S **do**
7:     **for** $k = 0$ to $j-1$ **do**
8:       score $= F[i-1][k] + \sum_{h=k+1}^{j} A[i][h]$
9:       **if** score $\geq F[i][j]$ **then**
10:         $F[i][j] =$ score, Path$[i][j] = (i-1, k)$
11:       **end if**
12:     **end for**
13:     **for** $k = 0$ to $i-1$ **do**
14:       score $= F[k][j-1] + \sum_{h=k+1}^{i} \frac{A[h][j]}{i-k}$
15:       **if** score $\geq F[i][j]$ **then**
16:         $F[i][j] =$ score, Path$[i][j] = (k, j-1)$.
17:       **end if**
18:     **end for**
19:   **end for**
20: **end for**
21: $m, n = M, N$
22: **while** $m \neq 0$ **and** $n \neq 0$ **do**
23:   $i, j =$ Path$[m][n]$
24:   add the aligned pair ($x_{[j+1:n]}$,$y_{[i+1:m]}$) to D
25:   $m, n = i, j$
26: **end while**
27: **return** D

4 Experiments and Results

4.1 Experimental Setup

Dataset *Unpaired Lyric and Melody.* We use "380,000+ lyrics from MetroLyrics"[1] as our unpaired lyrics for pre-training, which contains 362,237 songs. The lyrics in each song are split into sentences by the line break. For unpaired melodies, we choose "The Lakh MIDI Dataset" (Raffel 2016)[2]. The dataset contains 176,581 MIDI files with complete tracks, and we extract the melody tracks by Midi-miner[3]. Finally, we get 65,954 melodies as our unpaired data for pre-training. According to the characteristics of vocal melody, we consider the pitch and duration tokens of each note as the melody sequence. Each melody is transposed to the scale of C major or A minor. All the notes are shifted by octave so that the most pitches of the song fall into one-lined octave (MIDI pitch from 60 to 71). For unpaired melody MIDI file, we calculate the starting beat and duration of the note based on the absolute time and the BPM (Beats Per Minute), all the notes are aligned to 1/16 notes as paired data. We spread the melodies into sequences of pitch-duration patterns, as melody sequences for our model. For example, the melody in Figure 1 will be represented as "R, 7/16, G3, 1/16, E4, 1/8 ...". During pre-training, we simply split the unpaired melodies into phrases according to the average phrase length in paired data, since there is no natural phrase segmentation symbol in the MIDI files. *Paired Lyric and Melody.* We use the LMD dataset (Yu and Canales 2019)[4] which contains aligned melodies and lyrics from 7,998 songs. We apply the same operation, as aforementioned, to process melody and lyric data. The lyrics/melodies are split into sentences/phrases based on the annotations.

Model Configuration and Training We choose Transformer (Vaswani et al. 2017) as our basic model structure, which consists of 6 encoder/decoder layers. The hidden size and filter size of each layer are set as 512 and 2048. The number of attention heads is 8. We use the same masking strategy as in Song et al. (2019). We use Adam optimizer (Kingma and Ba 2015) with a learning rate of 5e-4. The model is trained on a NVIDIA Tesla T4 GPU card, and each mini-batch contains 4096 tokens. During training, we apply dropout with the rate of 0.1. The hyper-parameter α is set as 0.5. The dataset is split as training/valid/test set with a ratio of 8:1:1. Our baseline is a standard Transformer model, using the same model configuration with SongMASS but without any pre-training or alignment constraints.

[1] https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics

[2] https://colinraffel.com/projects/lmd

[3] https://github.com/ruiguo-bio/midi-miner

[4] https://github.com/yy1lab/Lyrics-Conditioned-Neural-Melody-Generation

	Lyric-to-Melody				Melody-to-Lyric
	PD (%) ↑	DD (%) ↑	MD ↓	PPL ↓	PPL ↓
Baseline	38.20	52.00	2.92	3.27	37.50
SongMASS	57.00	65.90	2.28	2.41	14.66
− pre-training	43.50	57.00	2.79	3.72	45.10
− separate encoder-decoder	55.00	64.80	2.32	2.53	15.57
− supervised loss	47.20	53.60	3.29	2.92	27.50
− alignment	56.10	65.20	2.36	2.07	8.54

Table 1: Results of lyric-to-melody and melody-to-lyric generation in objective evaluation.

4.2 Evaluation Metrics

In this subsection, we introduce the objective and subjective metrics used in this paper to evaluate the quality of lyric-to-melody and melody-to-lyric generation.

Objective Evaluation We mainly measure the similarity between the generated melody and ground-truth melody in lyric-to-melody generation, in terms of pitch and duration distribution and melody sequence, which are described below. We use perplexity (PPL) to measure the model fitness for both lyric-to-melody and melody-to-lyric generations. Besides, we also use alignment accuracy to measure alignment quality in two generation tasks, which is also described below.

- *PD* and *DD* (Pitch and Duration Distribution Similarity): We calculate the distribution (frequency histogram) of pitches and durations in melodies, and measure the similarity (average overlapped area (Ren et al. 2020)) of the distribution between generated melodies and ground-truth melodies: $\frac{1}{N_s}\sum_{i=1}^{N_s} OA(\mathrm{Dis}_i, \hat{\mathrm{Dis}}_i)$, where Dis_i and $\hat{\mathrm{Dis}}_i$ represent the pitch or duration distribution of the i-th generated and ground-truth song respectively, N_s is the number of songs in the testset, OA represents the average overlapped area.
- *MD* (Melody Distance): To evaluate the pitch trend of the melody, we spread out the notes into a time series of pitch according to the duration, with a granularity of 1/16 note. We subtract each pitch with the average pitch of the entire sequence for normalization. To measure the similarity between the generated and ground-truth time series with different lengths, we use dynamic time warping (Berndt and Clifford 1994) to measure their distance.
- *Alignment Accuracy*: To evaluate the alignment between melodies and lyrics, for each token in the source sequence, we calculate how many tokens in the target sequence (generated or ground-truth) are aligned to it, and check if the number of the tokens in the generated sequence equals to that in the ground-truth sequence. We calculate the ratio of equals among all source tokens and all songs in the test set to obtain the alignment accuracy.

Subjective Evaluation For subjective evaluation, we invite 5 participants with professional knowledge in music and singing as human annotators to evaluate 10 songs (338 pairs of generated lyric sentences and melody phrases) randomly selected from our test set. We require each annotator to answer some questions using a five-point scale, from 1 (Poor) to 5 (Perfect). The whole evaluation is conducted in a blind-review mode. Inspired by Watanabe et al. (2018), the metrics to evaluate the generated lyrics are as follows: 1) *Listenability*: Is the lyric sounds natural with the melody? 2) *Grammaticality*: It the lyric grammatically correct? 3) *Meaning*: Is the lyric meaningful? 4) *Quality*: What is the overall quality of the lyric? The metrics to evaluate the melody are as follows: 1) *Emotion* (Bao et al. 2019): Does the melody represent the emotion of the lyrics? 2) *Rhythm* (Zhu et al. 2018): Are the note durations and pauses of the melody sound natural? 3) *Quality* (Watanabe et al. 2018): What is the overall quality of the melody?

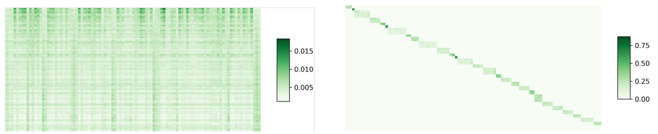
Metric	**Baseline**	**SongMASS**
Lyric		
Listenability	1.67 ± 0.62	2.00 ± 0.65
Grammaticality	3.00 ± 0.76	3.27 ± 0.59
Meaning	2.20 ± 0.68	3.20 ± 0.68
Quality	2.27 ± 0.46	3.00 ± 0.38
Melody		
Emotion	2.40 ± 1.06	3.53 ± 0.64
Rhythm	2.33 ± 1.18	2.87 ± 0.74
Quality	2.33 ± 1.05	2.93 ± 0.70

Table 2: Subjective evaluation results. Average scores and standard deviations are shown for each measure.

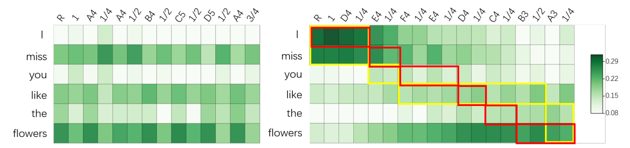
4.3 Results

The main results of the objective evaluation of lyric-to-melody and melody-to-lyric generations are shown in Table 1. The baseline model uses the same model structure with SongMASS, but does not leverage unsupervised melody and lyric data for pre-training and does not leverage attention-based alignment constraints. It can be seen that SongMASS greatly outperforms the baseline model in all objective metrics. The subjective evaluations are shown in Table 2, from which we can see that the lyrics and melodies generated by SongMASS obtain better average scores in all subjective metrics. These results demonstrate the effectiveness of SongMASS in generating high-quality lyric and melody[5]. We further conduct ablation study to verify the effectiveness of pre-training and alignment constraint in Song-

[5] Melody and lyric samples are available at: https://musicgeneration.github.io/SongMASS/



(a) Left: without sentence-level constraints. Right: with sentence-level constraints.



(b) Left: without token-level constraints. Right: with token-level constraints.

Figure 6: Attention visualization. All of the results are displayed on the average attention score of all heads in the last layer of the encoder-decoder attention in Transformer. In Figure 6(b), the red blocks are the alignments searched by our dynamic programming algorithm while the yellow blocks are by the greedy algorithm described in the second paragraph in Section 4.4.

MASS. As shown in Table 1, removing each component results in worse performance than SongMASS[6], demonstrating the contribution of pre-training and alignment constraint.

4.4 Method Analysis

Pre-training Method We further investigate the effectiveness of each design in pre-training method, including using separate encoder-decoder for lyric-to-lyric and melody-to-melody pre-training and using supervised pre-training to learn a shared latent space between lyric and melody. From Table 1, removing separate encoder-decoder (i.e., using shared encoder-decoder) and removing supervised loss both result in worse performance than SongMASS, which demonstrates the effectiveness of the two designs.

	L2M Acc ↑	M2L Acc ↑
SongMASS	62.6	45.4
- TC	62.1	44.8
- SC	56.2	44.0
- TC - SC	55.3	43.8
- TC - SC - PT	48.3	37.1
- DP	15.7	11.3

Table 3: Analyses of the designs in alignment constraints.

Alignment Strategy We study the effectiveness of the sentence-level and token-level alignment constraints (denoted as SC and TC respectively) on the alignment accuracy (denoted in Section 4.2) between melodies and lyrics. The results are shown in Table 3. It can be seen that both token-level and sentence-level (especially sentence-level) constraints can improve alignment accuracy. It is interesting that pre-training (PT) also benefits alignment, which is probably because the patterns of lyrics and melodies are better captured with pre-training. Finally, we investigate the alignment accuracy without dynamic programming (DP) algorithm. In this case, we implement a naive alignment algorithm on attention weight matrix, which greedily decides to add another token to the current one-to-many or many-to-one alignment or to start a new alignment pair at each time step. When the sequence reaches the last token, we align all the remaining tokens of the other sequence to that token to ensure all tokens are aligned. We find that the alignment accuracy is drastically decreased without DP in Table 3, showing the importance of DP for accurate alignments.

[6]Removing alignment constraint causes slightly better performance in PPL, which indicates that attention constraint may harm the fitting capability of the model, but still result in better generation accuracy in terms of PD, DD and MD. We also demonstrate in Table 3 that alignment constraint indeed improves the alignment accuracy of the generated results.

Alignment Visualization To better highlight the advantages of our alignment strategy, we further visualize some cases from the lyric-to-melody tasks, as shown in Figure 6. Figure 6(a) shows the attention weights of the whole song with and without sentence-level alignment constraints. It can be seen that the attention weights without sentence-level constrains are dispersed in all positions of the whole long sequence, and the target token cannot attend to the correct source sentences. When using sentence-level constraints, there are monotonous alignments between source and target sequence, which demonstrates the effectiveness of sentence-level alignments. Figure 6(b) shows the differences of whether using token-level constraints or not. We find that the attention distributions without the token-level constraints are chaotic. When applying token-level attention constraints, there are obvious diagonal trend in the attention weights, which further enable the dynamic programming algorithm to find a better alignment path as marked in red rectangles. These results demonstrate the effectiveness of token-level alignment constraints.

5 Conclusion

In this paper, we have proposed SongMASS, an automatic song writing system for both lyric-to-melody and melody-to-lyric generation, which leverages masked sequence to sequence pre-training and attention-based alignment constraint. We introduce some specific designs based on MASS for lyric-to-lyric and melody-to-melody pre-training, including song-level unsupervised pre-training and supervised pre-training loss to learn a shared latent space between lyric and melody. Furthermore, we introduce the sentence-level and token-level alignment constraints, and a dynamic programming algorithm to obtain accurate alignments between lyric and melody. Experimental results show that our proposed SongMASS greatly improves the quality of lyric-to-melody and melody-to-lyric generation compared with the baseline. For future work, we will investigate other sequence to sequence pre-training methods and more advanced alignment algorithms for lyric-to-melody and melody-to-lyric generation.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints* abs/1409.0473.

Bao, H.; Huang, S.; Wei, F.; Cui, L.; Wu, Y.; Tan, C.; Piao, S.; and Zhou, M. 2019. Neural Melody Composition from Lyrics. In *NLPCC*, volume 11838, 499–511.

Berndt, D. J.; and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, 359–370. Seattle, WA, USA:.

Choi, K.; Fazekas, G.; and Sandler, M. B. 2016. Text-based LSTM networks for Automatic Music Composition. *CoRR* abs/1604.05358.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *ICLR*.

Lee, H.-P.; Fang, J.-S.; and Ma, W.-Y. 2019. iComposer: An Automatic Songwriting System for Chinese Popular Music. In *NAACL*, 84–88.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* .

Li, P.; Zhang, H.; Liu, X.; and Shi, S. 2020. Rigid Formats Controlled Text Generation. In *ACL*, 742–751.

Lu, X.; Wang, J.; Zhuang, B.; Wang, S.; and Xiao, J. 2019. A Syllable-Structured, Contextually-Based Conditionally Generation of Chinese Lyrics. In *PRICAI*, volume 11672, 257–265.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*, 1412–1421.

Malmi, E.; Takala, P.; Toivonen, H.; Raiko, T.; and Gionis, A. 2015. DopeLearning: A Computational Approach to Rap Lyrics Generation. *CoRR* abs/1505.04771.

Needleman, S. B.; and Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48: 443–453.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI* .

Raffel, C. 2016. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. *PhD Thesis* .

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* .

Ren, Y.; He, J.; Tan, X.; Qin, T.; Zhao, Z.; and Liu, T. 2020. PopMAG: Pop Music Accompaniment Generation. *CoRR* abs/2008.07703. URL https://arxiv.org/abs/2008.07703.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*, 379–389.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*, volume 97, 5926–5936.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Watanabe, K.; Matsubayashi, Y.; Fukayama, S.; Goto, M.; Inui, K.; and Nakano, T. 2018. A Melody-Conditioned Lyrics Language Model. In *NAACL*, 163–172.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NIPS*, 5753–5763.

Yu, Y.; and Canales, S. 2019. Conditional LSTM-GAN for Melody Generation from Lyrics. *CoRR* abs/1908.05551.

Zhu, H.; Liu, Q.; Yuan, N. J.; Qin, C.; Li, J.; Zhang, K.; Zhou, G.; Wei, F.; Xu, Y.; and Chen, E. 2018. XiaoIce Band: A Melody and Arrangement Generation Framework for Pop Music. In *KDD*, 2837–2846.

Appendix

Visualization Examples

To demonstrate the advantages of our methods in alignment, we further randomly choose some cases for attention visualization. The results are shown Figure 7. From Figure 7, we observe obvious monotonous alignments in each case, and the dynamic programming algorithm achieves more precise alignments than greedy alignments.

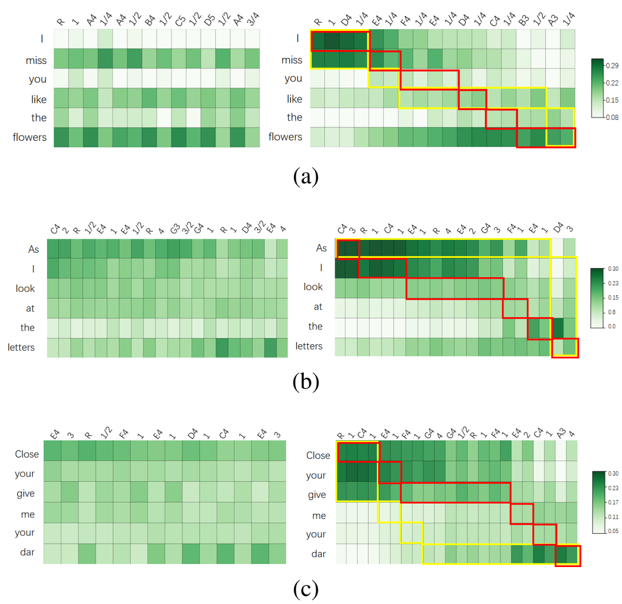


Figure 7: Attention visualization. The meaning of red and yellow blocks are same as Figure 7.