S^{2}MILE: Semantic-and-Structure-Aware Music-Driven Lyric Generation

Mu You[1,2], **Fang Zhang**[1,2], **Shuai Zhang**[1], **Linli Xu**[1,2*]

[1]School of Computer Science and Technology, University of Science and Technology of China
[2]State Key Laboratory of Cognitive Intelligence
{youmu1998, fangzhang, shuaizhang}@mail.ustc.edu.cn, linlixu@ustc.edu.cn

Abstract

The task of music-to-lyric generation aims to create lyrics that can be sung in harmony with the music while capturing the music's intrinsic meaning. Previous efforts in this area have struggled to effectively handle both the structural and semantic alignments of music and lyrics, often relying on rigid, manually crafted rules or overlooking the semantic essence of music, which deviates from the natural lyric-writing process of humans. In this paper, we bridge the structural and semantic gap between music and lyrics by proposing an end-to-end model for music-driven lyric generation. Our model aims at generating well-formatted lyrics based solely on the music while capturing its inherent semantic essence. In the music processing phase, we introduce a hierarchical music information extractor, which operates at both the song and sentence levels. The song-level extractor focuses on discerning the overall semantic content of the music, such as themes and emotions. Simultaneously, the sentence-level extractor captures the local semantic and structural details from note sequences. Additionally, we propose a lyric length predictor that determines the optimal length for the generated lyrics. During the lyric generation phase, the information gathered by the above modules is integrated, providing essential guidance for the downstream lyric generation module to produce coherent and meaningful lyrics. Experimental results on objective and subjective benchmarks demonstrate the capabilities of our proposed model in capturing semantics and generating well-formatted lyrics.

Introduction

Automatic music-to-lyric generation is an interesting and challenging task in both academic research and industrial domains. This task seeks to emulate the human process of lyric composition, generating lyrics that are both semantically coherent and structurally aligned with the accompanying music.

In traditional lyric composition, lyricists first determine the theme and content of lyrics by analyzing semantic elements embedded in the music, such as the emotional tone, intensity, and associated imagery. Then, based on the music's structural attributes, lyrics are completed with a specific format or style (Culler 2017). However, due to the intricate relationship between music and lyrics (North, Krause, and Ritchie 2021; Johnson, Huron, and Collister 2014; Suharto 2004; Barradas and Sakka 2022; Mesaros and Virtanen 2008), previous works simplify this task to melody-to-lyric generation. Melodies, with their single-channel and monophonic characteristics (Pinkerton 1956), have a simpler structure compared to complete music compositions, making them more manageable for this purpose. Despite this simplification, melodies play an important role in the semantic expression of music (Wallace 1994), which is why these approaches have still achieved impressive performance. Some of these works focus on establishing structural correlations between melody and lyrics. For example, studies from (Watanabe et al. 2018) and (Lee, Fang, and Ma 2019) are based on the assumption that each musical note corresponds to a syllable in the lyrics. The methodology is further extended in SongMASS (Sheng et al. 2021) by leveraging well-annotated datasets that align notes with syllables and words. Additionally, Ding et al. (2024) develop an automatic procedure to extract the melody from music and fully fine-tune a large language model (LLM) to generate lyrics based on the extracted melody. Other works introduce predefined conditions to control the generation of lyrics. Among them, Ma et al. (2021) combine the structural details of the melody and predefined keywords to generate lyrics from a constrained vocabulary. Tian et al. (2023) design a template with predetermined titles, genres, and keywords, which is then filled with melodic structural details to create lyrics. Ou, Ma, and Wang (2023) use predefined sentence lengths and melodies as inputs for lyric generation.

While these methods have achieved certain success in melody-to-lyric generation, they exhibit specific limitations compared to human lyric writing:

Oversimplified structural alignment and lack of annotated data. Both the one-to-one note-syllable mapping and the artificial mappings based on note sequence lengths and syllable stresses oversimplify the structural relationship between music and lyrics. This simplification restricts the ability to create diverse and natural lyrical compositions. Moreover, the lack of data annotated by experts presents a significant challenge to this task.

Inconsistency between assigned semantic conditions and the inherent semantics of music. In traditional lyric composition, lyricists derive content from the inherent semantics of

[*]Corresponding author

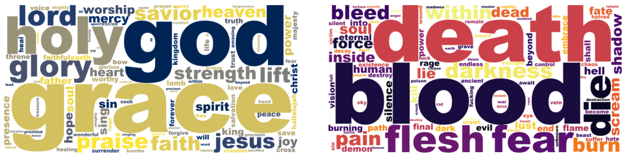

Figure 1: Top 100 lyric words in religious music (left) and metal music (right) from dataset (Fell and Sporleder 2014).

music, ensuring coherence between the lyrics and the music. Figure 1 partially illustrates this coherence. However, previous studies either overlook these semantics or impose semantic conditions manually, focusing solely on structural alignments. This results in a semantic misalignment between the generated lyrics and the music.

To resolve the aforementioned issues and extend melody-to-lyric generation to music-to-lyric generation, we propose S²MILE, an end-to-end model for lyric generation that integrates both the structural characteristics and semantic content of music. Unlike prior approaches focusing solely on melody, our method captures the richness of complete music compositions. Specifically, our model contains a hierarchical music information extractor, a lyric length predictor, and a lyric generator. Given the complex, multi-instrumental, and polyphonic nature of music (Stefan Kostka 1995), we employ a hierarchical music information extractor. This module captures the semantic and structural elements embedded in music at both the song and sentence levels. At the song level, it first encodes the entire song and then aligns the encoded representation with a summary of the lyrics. Following this alignment, the extractor produces a unified representation that conveys the abstract semantics of the song, such as its theme and emotions. At the sentence level, the extractor embeds the multi-instrument note sequence corresponding to each lyric line. It analyzes various note attributes, including pitch, velocity, instrument type, and timing details, such as the onset of each note and the rests between them. This detailed analysis enables the extractor to identify the music's fine-grained structural and semantic information necessary for accurate lyric generation. To complement the music information extractor, we develop a lyric length predictor that estimates the optimal length for the generated lyrics. This module considers the influence of various musical instruments on lyric length by analyzing features such as pitch variance and velocity centroid within each instrument (Burt 2016). This ensures that the lyrics are well-aligned with the music's structure. It is trained on an extensive collection of songs with timestamped lyrics, which is easily accessible online. During the lyric generation phase, we leverage a large language model (LLM) as the core architecture. The lyric generation module integrates the predicted lyric length with the semantic and structural information extracted from the music to generate lyrics. The LLM's outstanding data comprehension and integration capabilities ensure that the lyrics produced are contextually relevant and structurally coherent.

Due to the complex relationship between music and lyrics, which differs significantly from the direct mappings in machine translation, assessing the quality of generated lyrics is particularly challenging. Traditional metrics like BLEU (Papineni et al. 2002), NIST (Doddington 2002), and METEOR (Banerjee and Lavie 2005) are limited and biased in this context, as evidenced by various studies (North, Krause, and Ritchie 2021; Johnson, Huron, and Collister 2014; Suharto 2004; Barradas and Sakka 2022; Mesaros and Virtanen 2008). To address these limitations, we employ CLAP (Elizalde et al. 2023) and CLaMP (Wu et al. 2023) to measure the correlation between music and generated lyrics. These methods leverage contrastive learning on extensive music-text pairs to ensure precise semantic alignment, providing a more accurate evaluation of lyric quality.

Our method presents significant advancements over previous efforts in the following aspects:

- We propose an end-to-end method for generating lyrics directly from music. This approach includes a hierarchical music information extractor that captures both song-level and sentence-level music representations, ensuring that the generated lyrics are semantically and structurally aligned with the music. Extensive experiments on standard datasets demonstrate the superior performance of our method compared to previous models.
- The lyric length predictor addresses the limitations of rigid, manually defined rules for note-to-syllable correspondence and the challenges of the scarcity of well-annotated datasets. By leveraging extensive amounts of weakly supervised data, it accurately predicts the appropriate length of generated lyrics, providing a more flexible, context-aware method for achieving structural alignment between music and lyrics.
- We evaluate the quality of the generated lyrics using the CLAP and CLaMP scores, which directly measure the correlation between lyrics and music. Unlike traditional metrics such as BLEU, NIST, and METEOR, which focus on lyric-to-lyric similarity akin to translation tasks, our approach emphasizes the complex relationship between music and lyrics, providing a more reasonable measure of their semantic alignment.

Related Work

Lyric Generation

Lyric generation aims to imitate the human process of composing lyrics in response to music. Existing works can be categorized based on whether music is present. Specifically, works on lyric generation without music rely on predefined textual structures and semantic information. For instance, ChipSong (Liu and Han 2022) employs sentence and phrase lengths, trigger words, and rhythmic patterns to generate lyrics. Similarly, Youling (Zhang et al. 2020) utilizes content-controlling attributes (style, emotion, theme, expected keywords) and format-controlling attributes (acrostic, rhyme, line and word counts). In comparison, lyric generation with music aligns closer with the traditional lyric-writing process and can be further divided into the following:

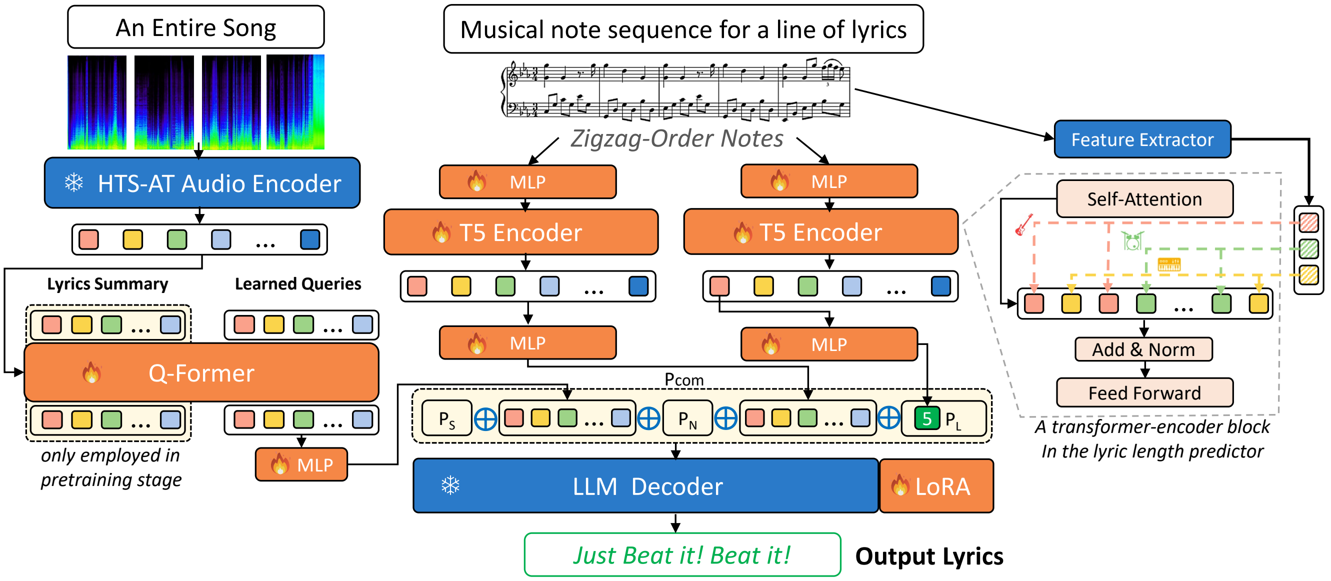

Figure 2: Model architecture. The sentence-level extractor and the lyric length predictor process note sequences in a zigzag order. The song-level extractor encodes the entire song in waveform. The embeddings and predicted lyric length from these modules are concatenated with prompts and fed into the lyric generator to produce lyrics.

Generation Based on Melody: Early methods (Watanabe et al. 2018; Lee, Fang, and Ma 2019; Chen and Lerch 2020) align notes with lyrics based on a one-to-one correspondence between syllables and notes. However, this assumption is not always valid (North, Krause, and Ritchie 2021). Subsequent works (Sheng et al. 2021; Ou, Ma, and Wang 2023) leverage data with syllable-note level annotations, but the scarcity of such annotated data makes it difficult to train models with strong generalization capabilities. Chen and Lerch (2020) attempt to extract semantic information from music by clustering it into five keyword-represented categories.

Generation Based on Melody and Hand-Crafted Semantics: These works aim to intentionally integrate predefined semantics into the generation process to enrich the semantic content of the generated lyrics. For example, Ma et al. (2021) utilize a constrained vocabulary and predefined keywords. Tian et al. (2023) create templates with hand-crafted semantics like keywords, titles, and genres. Ou, Ma, and Wang (2023) set word count constraints per sentence. However, they overlook the semantic relevance between music and lyrics, which could lead to discordant outputs.

Modality Alignment

CLIP (Radford et al. 2021), a well-known image-text alignment model, embeds images and texts into a shared latent space with contrastive learning. In the audio-text domain, models like CLAP (Elizalde et al. 2023) and CLaMP (Wu et al. 2023) employ a similar approach for coarse-grained alignment on large audio-text datasets. However, these models are designed for classification and are unsuitable for generation. BLIP-2 (Li et al. 2023), proposed for image-to-text generation, includes a Querying Transformer (Q-Former) module that captures modality-specific information at variable granularities and fuses information from both modalities into query embeddings, facilitating cross-modal generation tasks.

Problem Definition

The process of generating lyrics corresponding to the music involves creating a text sequence $L = \{w_1, w_2, \ldots, w_j, \ldots, w_x\}$ based on a sequence of musical notes $N = \{n_1, n_2, \ldots, n_k, \ldots, n_y\}$. In this context, w_j denotes the j^{th} word in the text, while n_k indicates the k^{th} note in the music, represented as a five-dimensional vector $n_k = (p_k, v_k, s_k, d_k, i_k)$. The components of this vector include p_k for pitch (ranging from 0 to 127); v_k for velocity (also within the range of 0 to 127); s_k, for the start time of the note (measured in seconds from the onset of the piece); d_k for the duration (measured in seconds capturing the time span from the note's start to its end), and i_k represents the instrument ID number in the MIDI file.

Additionally, for each group of instruments, we allow multiple notes to be triggered simultaneously, which means the model can receive polyphonous note sequences as s_{k_1} can be equal to s_{k_2} when k_1 is not equal to k_2. Inspired by BANDNET (Zhou et al. 2019), we organize the notes in a zigzag pattern, sorting simultaneous notes first by ascending instrument number and then by descending pitch, while sequencing notes from different moments chronologically.

Method

System Overview

Our model for Semantic-and-Structure-Aware Music-driven Lyrics Generation (S²MILE) is illustrated in Figure 2. It consists of three modules:

- Hierarchical Music Information Extractor: Extracts music representations at both the song and sentence levels.
- Lyric Length Predictor: Predicts the appropriate length for the generated lyrics.
- Lyric Generator: Generates lyrics based on the extracted music information and predicted length.

Hierarchical Music Information Extractor

This module includes a song-level extractor and a sentence-level extractor. The song-level extractor is designed to capture the overall semantic representations of a song, including themes and emotions. Meanwhile, the sentence-level extractor focuses on deriving fine-grained music representations from individual note sequences.

Song-level Extractor Ensuring semantic coherence between music and lyrics is crucial in music-driven lyric generation. Our song-level extractor is designed to achieve that by integrating musical compositions with lyrical content through advanced encoding and querying techniques. It harnesses the HTS-AT encoder (Chen et al. 2022) to transform complete note sequences of songs into waveform-encoded semantic information. Central to the extractor is the Querying Transformer (Q-Former) (Li et al. 2023), which translates these musical encodings into the text space, enabling the effective semantic extraction from the music. The Q-Former operates with a set of learnable query embeddings that interact with music encodings and lyric summaries. This interaction is facilitated through cross-attention and self-attention mechanisms derived from BERT (Devlin et al. 2019), an encoder-only transformer. Separate feed-forward networks are employed for each modality, music and text respectively, ensuring robust feature extraction.

Our pretraining process is designed to enhance the Q-Former's capability to extract and integrate song features with lyric summaries accurately. This is achieved through three pretraining tasks: Music-Summary Contrasting, Music-Summary Matching, and Music-Grounded Summary Generation. These tasks collectively enhance the model's ability to maintain semantic coherence between the generated lyrics and the input music. For more details, please refer to the Appendix.

Sentence-level Extractor To capture fine-grained structural and semantic details from musical phrases corresponding to each line of lyrics, we employ a sentence-level extractor(illustrated in Figure 3) based on the encoder-only transformer architecture (Vaswani et al. 2017). While the song-level extractor ensures high-level semantic coherence between lyrics and music, our approach generates lyrics sequentially, one sentence at a time. Therefore, the detailed content and structure of each lyric sentence are primarily influenced by its corresponding musical segment, a sequence of notes linked to the lyric. Specifically, we employ a twelve-layer T5 encoder (Raffel et al. 2020) and pretrain it on 436, 631 MIDI files from the MetaMIDI dataset (Ens and Pasquier 2021). It is modelled as follows:

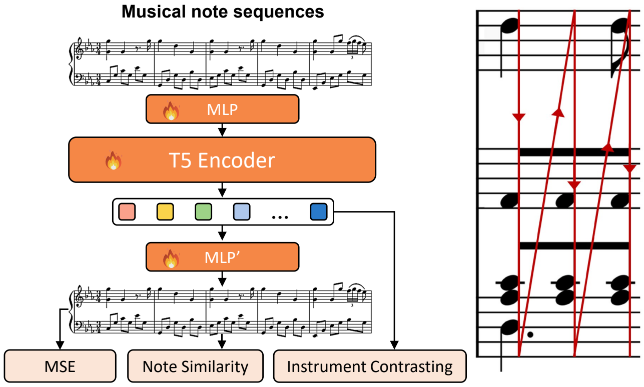

Figure 3: Sentence-level extractor and zigzag pattern. MLP' participates exclusively in the pretraining stage.

$$\text{Enc}_\text{S}(N) = H \in \mathbb{R}^{|N|\times d_S} \tag{1}$$

Here, N represents the note sequence $\{n_1, n_2, ..., n_k, ...\}$, and $|N|$ indicates its total number of notes. The extractor, Enc_S, transforms these notes into a hidden state matrix H, with d_S defining the dimensionality of the hidden states.

We commence the pretraining phase with a masked language modeling (MLM) task, where 15% of the note vectors are intentionally masked as $(-1, -1, -1, -1, -1)$ to represent missing data, enhancing the model's ability to understand the contextual information of the music. The primary objective is initially designed with a hybrid MSE-CE loss, which is a combination of mean squared error (MSE) and cross-entropy (CE) loss:

$$\begin{aligned} L_\text{MSE-CE}(N) = \sum_{n_m \in [\text{mask}]} \Big[&\text{MSE}(\text{Rec}_\text{S}(n_{m_{v,s,d}}), n_{gt_{v,s,d}}) \\ &+ \text{CE}(\text{Rec}_\text{S}(n_{m_{p,i}}), n_{gt_{p,i}})\Big] \end{aligned} \tag{2}$$

where $\text{Rec}_\text{S}(n_{m_*})$ denotes the masked note n_m recovered by Enc_S, n_{gt} is its corresponding original note, and $* \in \{p, v, s, d, i\}$ represents different attributes of the note: pitch (p), velocity (v), start time (s), duration (d), and instrument index (i). For continuous attributes like v, s, and d, the mean squared error is employed. Conversely, for the discrete attributes such as p and i, the cross-entropy loss is utilized.

Additionally, we introduce the note similarity loss, which enhances the extractor's ability to capture the detailed distributions of note sequences:

$$L_\text{NS}(N) = \sum_{n_m \in [\text{mask}]} \left(1 - \frac{\text{Rec}_\text{S}(n_m) \cdot n_{gt}}{||\text{Rec}_\text{S}(n_m)|| \cdot ||n_{gt}||}\right) \tag{3}$$

Previous studies (Kartomi 1990; Wicaksana, Hartono, and Wei 2006; Eronen 2001; Herrera-Boyer, Peeters, and Dubnov 2003; Howle and Trefethen 2001) suggest that different instruments possess distinct sonic signatures and musical characteristics. Consequently, notes from the same instrument are expected to be more similar to notes from other instruments. Building on this insight, we propose the instrument contrastive loss:

$$L_{\text{IC}}(H) = \sum_{\text{Enc}_S(n_k)\in H}\left[\sum_{i_k=i_j}\left(\tau - \frac{\text{Enc}_S(n_k)\cdot\text{Enc}_S(n_j)}{\|\text{Enc}_S(n_k)\|\cdot\|\text{Enc}_S(n_j)\|}\right)\right.$$
$$\left. + \sum_{i_k\neq i_l}\frac{\text{Enc}_S(n_k)\cdot\text{Enc}_S(n_l)}{\|\text{Enc}_S(n_k)\|\cdot\|\text{Enc}_S(n_l)\|}\right] \tag{4}$$

Here, n_j, n_k, and n_l represent the j^{th}, k^{th}, and l^{th} notes respectively, with their corresponding instrument indices i_j, i_k, and i_l. The objective aims to minimize the difference between hidden states of notes from the same instrument (with τ between 0 and 1, here is 0.85) while maximizing the difference between notes from different instruments. This strategy helps the extractor distinguish note vectors more effectively, enriching the amount of information in the hidden states and alleviating the risk of extractor degeneration.

Finally, we add up these losses as the objective to guide the sentence-level extractor in capturing detailed structural and semantic representations from the music.

Lyric Length Predictor

Our previous experiments indicate that while the sentence-level extractor can control the length of the generated lyrics, its effectiveness is limited. To address this, we introduce a lyric length predictor(illustrated in Figure 4) to guide the lyric generator more precisely in producing lyrics with appropriate length. The core of our predictor is an encoder-only transformer, specifically a pretrained sentence-level extractor that processes note sequences. Inspired by Burt's theory on lyrics (Burt 2016), we integrate various musical features to determine the optimal lyric length measured in syllable count. Key features such as pitch variance, pitch contour, velocity centroid, and velocity contour are calculated for each instrument track. The features are normalized to a range between 0 and 1 and then combined to serve as coefficients that modulate the outputs of the self-attention layers within the transformer. The first logit produced by the transformer is then processed through a dual-layer MLP to predict the lyric length.

During the pretraining phase, the predictor receives the note sequence as input and processes it through the transformer and the music feature extractor. The music feature extractor computes the music features $F \in \mathbb{R}^{|F|\times 1}$, where $|F| = 4$ represents the number of the designed features for each instrument's corresponding note sequence. For more details on feature calculation, please refer to the Appendix. For the feature $f^k_{i_j} \in F$ from each instrument i_j (where i_j denotes the j^{th} instrument in the note sequence and f^k indicates k^{th} feature in F), they are normalized among instruments to compute the weighted feature $\hat{f}^k_{i_j}$:

$$\hat{f}^k_{i_j} = \frac{f^k_{i_j}}{\sum_{i_l\in I} f^k_{i_l}} \tag{5}$$

Here, I denotes the set of instruments in the note sequence. Subsequently, all the weighted features are combined across different feature types to compute the coefficients c_{i_j}:

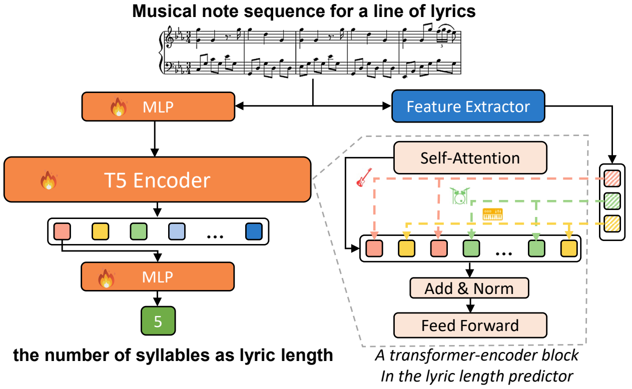
Musical note sequence for a line of lyrics
MLP
Feature Extractor
T5 Encoder
...
MLP
5
the number of syllables as lyric length
Self-Attention
...
Add & Norm
Feed Forward
A transformer-encoder block
In the lyric length predictor

Figure 4: Lyric length predictor

$$c_{i_j} = \sum_{k\in\{1,2,\ldots,|F|\}} \alpha_k \hat{f}^k_{i_j} \tag{6}$$

In this equation, α denotes a set of hyperparameters that adjust the influence of each weighted feature. The coefficient c_{i_j} scales the self-attention outputs for notes corresponding to the instrument i_j. The model's performance is evaluated using the MSE loss, which compares the predicted length to the actual length. For more details, please see the Appendix.

Lyric Generator

We employ Mistral (Jiang et al. 2023) as the backbone of the lyric generator, a decoder-only transformer pretrained extensively. It is noted for the superior performance in commonsense reasoning, world knowledge, and reading comprehension, which is ideal for music-to-lyrics generation. This task involves the intricate integration of music representations and contextual information during the lyric generation process, aiming to establish a deep connection between musical notes and the corresponding lyrics. During training, we fine-tune Mistral for lyrics generation using the LoRA technique (Hu et al. 2022) to efficiently adapt it for lyrics generation while preserving its pretrained weights. We map the music representations extracted by the hierarchical music information extractor to the hidden space of the lyric generator and concatenate them with the predicted lyric length to form a comprehensive prompt. Based on this prompt, the lyric generator produces lyrics autoregressively. For further details on this process, please refer to the Appendix.

Experiments

In this section, we conduct our experiments on the MetaMIDI dataset (Ens and Pasquier 2021), which comprises 436,631 tracks in MIDI format. From this dataset, we extract 53,367 tracks with time-stamped lyrics sourced online. The lyrics are segmented and aligned with their corresponding MIDI tracks at the sentence level, yielding 1,013,497 pairs of note sequences and lyrics.

Model Configuration The song-level extractor employs HTS-AT (Chen et al. 2022) as its audio encoder with a window size of 1024. For text encoding, we use a pretrained BERT model (12 layers, hidden size 768) with 32 query embeddings.

For the sentence-level extractor and lyric length predictor, we adopt the T5-base encoder (12 layers, hidden size 768). A two-layer MLP maps musical note sequences into the encoder's hidden space: (5, 768) → (768, 768). During pretraining, another MLP maps hidden representations back to the musical note space, while a similar MLP predicts syllable counts in lyric length predictor.

For lyric generation, we leverage Mistral-7B, a transformer decoder (32 layers, hidden size 4096). To align the song-level and sentence-level encoders with Mistral-7B, a two-layer MLP is employed. Mistral-7B's parameters remain frozen, and only the adaptive MLPs and LoRA components are trainable, reducing trainable parameters to approximately 40 million (5.71‰ of 7B).

We optimize modules with Adam (Kingma and Ba 2015), employing the following learning rates and batch sizes:

- Song-level Extractor: 1×10^{-5}, batch size 42, processing up to 24 audio tokens and 108 text tokens per batch.
- Sentence-level Extractor: 1×10^{-4}, batch size 12, processing 256 musical notes and 32 text tokens per batch.
- Lyric Length Predictor: 1×10^{-4}, batch size 12, processing up to 256 musical notes per batch.
- Lyric Generator: 2×10^{-4}, batch size 2, with 84 gradient accumulation steps.

Training is conducted on a single NVIDIA RTX 3090 GPU, using an 8:1:1 split for the training, validation, and test sets.

Methods in Comparison

We conduct comparisons with the following models. To ensure fairness, all melody-to-lyric generation models take the melody track from the music as their input.

SongMASS: An open-source model which employs transformer architecture to generate lyrics from melodies, utilizing a closed-source dataset with crafted syllable-note alignments.

GPT-3.5: A large language model which is known for its effective in-context learning capabilities. In our experiments, we utilize GPT-3.5 with two input types: (1) a sequence of musical notes (*GPT-3.5 w/ N*), and (2) a combination of keywords and musical notes (*GPT-3.5 w/ KN*) to generate lyrics. Each input type includes two pairs of note sequences and corresponding lyrics to facilitate its in-context learning.

SongComposer (Ding et al. 2024): The latest open-source model which is designed for melody-to-lyric generation based on large language models (LLMs). This model, using instruction tuning to generate lyrics, is nearly twice the size of our model and has been fully fine-tuned.

S²MILE: Our proposed model that integrates both semantic and structural music representations to enhance lyric generation. Additionally, to assess the impact of each component, we evaluate it through ablation studies: S²MILE without song-level information (*S²MILE w/o S*), S²MILE without sentence-level information (*S²MILE w/o N*) and S²MILE without predicted lyric length (*S²MILE w/o L*).

Evaluation Metrics

This subsection introduces objective and subjective metrics to analyze and compare each model's performance.

Objective Evaluation The objective evaluation measures the textual quality of lyrics and their semantic and structural relevance to the music.

The evaluation of the textual quality of the generated lyrics takes into account the following metrics:

Fluency: To measure the fluency of the generated lyrics, we calculate their perplexity (PPL) with GPT-2 Large (Radford et al. 2019). Additionally, we evaluate the integrity (Li et al. 2020) (INT) of each lyric line. This is done by analyzing the logits of GPT-2 Large at the end of each line. After applying the softmax function, we compute the probabilities of [End of Sentence (EOS)] and various punctuation marks (i.e., '.', '?', '!', ',', ';', ':'). A lyric line is considered complete if one of these tokens has the highest probability, suggesting a natural end to the line. It evaluates both the completeness and grammatical correctness of lyrics.

$$\text{Integrity} = 2^{-\frac{1}{|L|}\sum_{j=1}^{|L|} \log \max_{w_n^j \in \text{punc}} P\left(w_n^j | w_0^j, w_1^j, \ldots, w_{n-1}\right)} \tag{7}$$

Here, $|L|$ denotes the total number of generated lyrics. The term **punc** includes punctuation marks and the `[EOS]` token. w_n^j is the predicted word after the j^{th} lyric line from GPT-2 Large. The formula determines the integrity score by calculating the highest probability that the predicted word falls within the **punc** category.

Diversity: We calculate lyric diversity by randomly sampling one lyric line from each song to mitigate potential biases caused by high intra-song similarity. Following (Li et al. 2016), we measure diversity with distinct-1, distinct-2, and distinct-n metrics, which represent the proportion of unique unigrams, bigrams, and multigrams (up to six terms) relative to the total word count in the sample set.

To evaluate the alignment between the generated lyrics and the music, we consider the following metrics:

Format: The alignment between the length of a musical phrase and its lyrics is more accurately represented by syllable count than word count due to the syllabic nature of speech in vocal music. We measure deviations between generated and original lyrics with the Syllable Discrepancy Distance (SDD), defined as SDD $= |S(L_{gen}) - S(L_{ori})|$ where $S(L)$ denotes the syllable count in lyrics L. A minimal SDD indicates a closer match to the original lyrics, suggesting a more precise alignment with the musical phrase's length. Due to variations in both musical phrase and lyric lengths, we normalize the SDD metric and propose SDD-N and SDD-S, with note length and original syllable count as denominators, respectively.

Semantic similarity: We employ the CLAP and CLaMP scores to evaluate the semantic similarity between lyrics and music. Both audio-text alignment models derive their scores by calculating the cosine similarity between the generated lyrics and the original audio.

Models	Diversity			Fluency		Format			Semantic Similarity	
	Dist-1↑	Dist-2↑	Dist-N↑	PPL↓	INT↓	SDD↓	SDD-S↓	SDD-N↓	CLAP Score↑	CLaMP Score↑
SongMASS	0.46±0.02	0.85±0.03	0.96±0.01	911.22±233.33	272.30±27.78	**3.48±3.25**	**0.32±0.23**	**0.14±0.20**	0.042±0.077	0.163±0.048
GPT-3.5 w/ KN	0.31±0.02	0.52±0.03	0.75±0.03	53.06±9.91	2.21±0.09	20.54±6.11	2.30±1.17	1.35±2.98	0.025±0.103	0.167±0.048
GPT-3.5 w/ N	0.19±0.02	0.38±0.03	0.85±0.04	**38.35±3.81**	**2.09±0.06**	9.56±4.99	1.12±0.78	0.54±0.91	-0.013±0.109	0.152±0.035
SongComposer	0.21±0.02	0.49±0.06	**0.99±0.01**	213.22±98.14	35.30±11.67	32.65±40.26	3.59±4.70	1.71±3.47	0.023±0.082	0.145±0.044
S^{2}MILE	0.62±0.02	0.85±0.01	0.94±0.01	90.09±21.97	3.70±0.40	4.58±4.59	0.49±0.76	0.31±1.14	0.057±0.101	**0.203±0.041**
S^{2}MILE w/o L	**0.68±0.02**	**0.90±0.02**	0.95±0.01	74.38±12.40	2.69±1.37	7.34±5.09	0.87±0.76	0.51±1.16	**0.058±0.101**	0.201±0.050
S^{2}MILE w/o S	0.54±0.02	0.83±0.02	0.92±0.02	49.20±23.79	13.32±1.95	7.11±4.93	0.85±0.76	0.48±0.99	0.041±0.101	0.186±0.045
S^{2}MILE w/o N	0.52±0.03	0.89±0.03	0.94±0.01	38.50±15.63	10.38±1.64	7.71±4.83	0.92±0.77	0.50±0.97	0.052±0.099	0.199±0.046
Ground Truth	0.62±0.03	0.85±0.04	0.90±0.03	93.51±44.35	6.12±0.59	0	0	0	0.034±0.096	0.170±0.047

Table 1: Performance comparison on automatic evaluation. The best scores are bolded, and the second-best ones are underlined. ↑ denotes a preference for higher metric values, and ↓ for lower ones.

Models	Coherence↑	Meaningfulness↑	Structure relevance↑	Semantic relevance↑	Diversity↑
SongMASS	2.70±0.64	2.66±0.76	2.68±0.20	2.43±1.01	1.35±0.99
GPT-3.5 w/ KN	3.56±0.64	3.62±0.88	2.04±1.06	3.20±0.55	2.33±1.22
GPT-3.5 w/ N	3.12±0.62	3.10±0.80	**3.53±0.98**	2.88±1.20	1.37±1.09
SongComposer	2.89±0.96	2.42±0.93	2.38±0.97	2.54±1.06	1.67±0.98
S^{2}MILE	**3.60±0.62**	**3.75±0.87**	3.31±0.72	3.26±0.86	**2.82±0.96**
S^{2}MILE w/o L	3.39±0.75	3.67±0.65	3.19±1.08	**3.60±0.86**	2.79±1.00
S^{2}MILE w/o S	3.14±1.11	2.94±0.93	2.81±0.94	2.46±0.91	2.43±0.95
S^{2}MILE w/o N	3.13±0.92	2.99±0.98	2.96±1.06	2.40±0.88	2.14±1.01
Ground Truth	3.78±0.44	3.70±0.74	3.18±0.54	3.51±0.83	3.16±0.86

Table 2: Performance comparison on subjective evaluation. The best scores are bolded, and the second-best ones are underlined. ↑ denotes a preference for higher metric values, and ↓ for lower ones.

Subjective Evaluation We invite 17 participants with musical knowledge to evaluate 50 randomly selected pairs of generated lyric sentences and music phrases from our test set. Each participant uses a five-point scale, ranging from 1 (Poor) to 5 (Perfect), to assess the following criteria: 1) *Coherence*: Examines whether the lyrics are grammatically correct and logically structured. 2) *Meaningfulness*: Evaluates the degree to which the lyrics deliver a meaningful message. 3) *Structural Relevance*: Assesses the alignment between the lyrics' structure (e.g. length and rhythm) and the musical phrases. 4) *Semantic Relevance*: Checks whether the emotional tone of the lyrics matches the mood of the music. 5) *Diversity*: Evaluates the variety of meanings and expressions across the generated lyrics. Further details of the testing procedure and specifics are provided in the Appendix.

Results

Main Results The main results of the objective evaluation, summarized in Table 1, clearly show that S^{2}MILE outperforms all other models in overall performance. While SongMASS benefits from annotated data, it only slightly surpasses S^{2}MILE in formatting, falling significantly behind in all other metrics. GPT-3.5 demonstrates slightly better text fluency, but it struggles to grasp the meaning and structure of the music, leading to lower scores in semantic similarity and SDDs compared to S^{2}MILE. Furthermore, S^{2}MILE surpasses SongComposer in almost every metric, possibly due to the repetitive word patterns observed in the endings of SongComposer's generated lyrics, as shown in Table 3. The subjective evaluations are shown in Table 2, indicating that S^{2}MILE achieved higher average scores across most subjective metrics. Additionally, the subjective and objective diversity scores of S^{2}MILE closely align with the ground truth. It suggests that the model successfully extracts and translates musical elements into diverse lyrical expressions, leading to high-quality lyric generation.

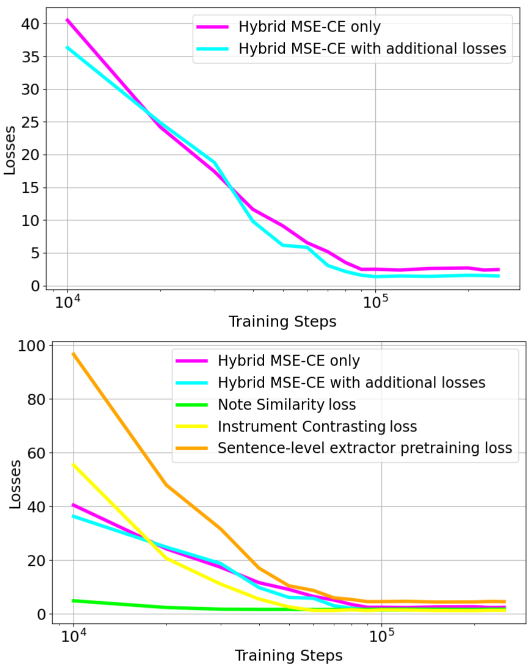

Figure 5: Ablation experiment on the pretraining stage of the sentence-level extractor. The addition of note similarity loss and instrument contrastive loss leads to a noticeable decrease in hybrid MSE-CE loss.

Ablation Study From the lower sections of Table 1 and Table 2, it is clear that the lyrics generated by S^{2}MILE consistently maintain high quality. Compared to the full S^{2}MILE model, the variant without the song-level extractor (S^{2}MILE w/o S) shows a decrease in semantic rele-

Ground Truth	S²MILE	SongMASS	GPT-3.5 w/ N	GPT-3.5 w/ KN	SongComposer
So strong against the hard winds as the years go by	Prepared for any turn of events, I trust you're aware	people begin to be jungle	Walking through the melody, feeling the rhythm in the air	In the timeless mountains, under the night sky, our true love will endure for eternity with enduring strength	that i will never say she is gone is gone is gone
And at his word we will rise and sing	Ignite my soul, I'll blaze through this agony	joni said to me	In the rhythm of the notes, we find harmony and melody entwined	The King of majesty conquered death, awake to the beauty, let us sing	you are so good to get it right for once in your life
Those stars have been there shinning though eternity	You ignite my passion, kindling my soul each time we meet	wonnin along the said meigh comes	Floating through the melody, we dance in harmony	In the timeless mountains, under the night sky, our true love shines with enduring strength and eternity	baby i will be there there there there
We can make this love go on forever	Always at the ready, my unwavering presence for you	lovely voice was born with our	Through the rhythm and flow, we find our way to the melody's embrace	Longing for a love so intense, it's crazy, but forever dreaming to realize	i have friends heart on you just is sweet

Table 3: Case study on lyrics generated from different models.

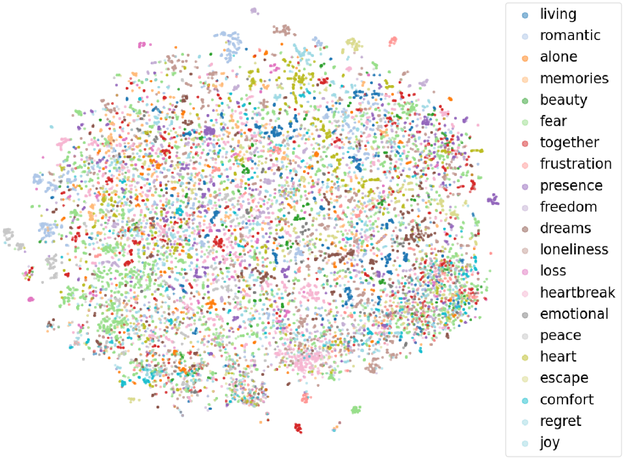

Figure 6: The song-level extractor's output (with only music as input) aligns closely with the lyrics' semantics, demonstrated by t-SNE with keyword labeling

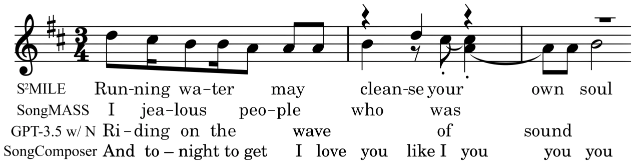
S²MILE Run-ning wa-ter may clean-se your own soul
SongMASS I jea-lous peo-ple who was
GPT-3.5 w/ N Ri-ding on the wave of sound
SongComposer And to – night to get I love you like I you you you

Figure 7: Case study on the coherence between note sequence and generated lyrics.

vance, while the variant without the sentence-level extractor (S²MILE w/o N) exhibits reduced structural relevance and lower SDD scores. These results highlight the crucial role of the hierarchical music information extractor in effectively capturing both song-level semantics and sentence-level details. When the lyric length predictor is removed (S²MILE w/o L), structural alignment significantly declines, particularly in SDD scores. This indicates that the lyric length predictor supports the lyric generator in producing lyrics of appropriate length. However, removing the predictor slightly improves text quality, suggesting that while it enhances structural alignment, it may cause the lyric generator to overfocus on length while compromising text quality. Figure 5 shows that incorporating the note similarity loss and instrument contrastive loss during the pretraining phase of the sentence-level extractor significantly reduces the hybrid MSE-CE loss from 2.4 to 1.4. This result supports previous research (Kartomi 1990; Eronen 2001) and indicates that introducing these two additional loss functions during pretraining is reasonable.

Case Study Table 3 showcases lyric examples generated by different models. The lyrics produced by S²MILE align closely with the ground truth in both semantics and structure. In contrast, lyrics generated by SongMASS often exhibit grammatical errors and logical inconsistencies. When prompted solely with musical notes, GPT-3.5 often repeats specific words such as "dance", "night", "melody", "rhythm", and "harmony". This issue persists even with additional keywords in the prompt. Furthermore, GPT-3.5 tends to produce lyrics that exceed the appropriate length. SongComposer, on the other hand, tends to conclude sentences with repeated phrases. Figure 6 visualizes the outputs of the song-level extractor and their corresponding keywords with t-SNE (Van der Maaten and Hinton 2008). The figure illustrates that outputs with the same keyword label consistently form distinct clusters, indicating that the song-level extractor effectively captures the semantic content of music and maps it into the lyric domain. Finally, Figure 7 shows that S²MILE achieves superior structural alignment between the generated lyrics and the note sequence compared to other models.

Conclusion

In this paper, we present S²MILE, an end-to-end automatic lyric generation model that processes multi-instrumental, polyphonic music and generates lyrics that are semantically and structurally aligned with the music. To achieve this, we propose a hierarchical music information extractor designed to capture both overarching song-level and detailed sentence-level music representations. Additionally, we integrate a lyric length predictor to precisely control the length of lyrics generated by the LLM-based lyric generator. Experimental results demonstrate that S²MILE significantly outperforms existing baseline models in music-to-lyric generation. In future work, we plan to enhance the model's capability to generate complete song lyrics in a single pass.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 62276245).

References

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.

Barradas, G. T.; and Sakka, L. S. 2022. When words matter: A cross-cultural perspective on lyrics and their relationship to musical emotions. *Psychology of Music*, 50(2): 650–669.

Burt, S. 2016. What Is This Thing Called Lyric? *Modern Philology*, 113(3): 422–440.

Chen, K.; Du, X.; Zhu, B.; Ma, Z.; Berg-Kirkpatrick, T.; and Dubnov, S. 2022. HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 646–650.

Chen, Y.; and Lerch, A. 2020. Melody-Conditioned Lyrics Generation with SeqGANs. *2020 IEEE International Symposium on Multimedia (ISM)*, 189–196.

Culler, J. 2017. Theory of the Lyric. *Nordisk poesi*, 2(2): 119–133.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Ding, S.; Liu, Z.; Dong, X.; Zhang, P.; Qian, R.; He, C.; Lin, D.; and Wang, J. 2024. SongComposer: A Large Language Model for Lyric and Melody Composition in Song Generation. arXiv:2402.17645.

Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, 138–145.

Elizalde, B.; Deshmukh, S.; Ismail, M. A.; and Wang, H. 2023. CLAP Learning Audio Concepts from Natural Language Supervision. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

Ens, J.; and Pasquier, P. 2021. Building the MetaMIDI Dataset: Linking Symbolic and Audio Musical Data. *International Society for Music Information Retrieval*, 182–188.

Eronen, A. 2001. Comparison of features for musical instrument recognition. In *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*, 19–22. IEEE.

Fell, M.; and Sporleder, C. 2014. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, 620–631.

Herrera-Boyer, P.; Peeters, G.; and Dubnov, S. 2003. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1): 3–21.

Howle, V.; and Trefethen, L. N. 2001. Eigenvalues and musical instruments. *Journal of computational and applied mathematics*, 135(1): 23–40.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. *International Conference on Learning Representations*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B.

Johnson, R. B.; Huron, D.; and Collister, L. 2014. 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.

Kartomi, M. J. 1990. *On concepts and classifications of musical instruments*. University of Chicago Press.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lee, H.-P.; Fang, J.-S.; and Ma, W.-Y. 2019. iComposer: An Automatic Songwriting System for Chinese Popular Music. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 84–88.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 110–119.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. C. H. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *International Conference on Machine Learning*.

Li, P.; Zhang, H.; Liu, X.; and Shi, S. 2020. Rigid Formats Controlled Text Generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 742–751.

Liu, N.; and Han, W. 2022. ChipSong A Controllable Lyric Generation System for Chinese Popular Song. *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, 85–95.

Ma, X.; Wang, Y.; Kan, M.-Y.; and Lee, W. S. 2021. AI-Lyricist: Generating Music and Vocabulary Constrained Lyrics. *Proceedings of the 29th ACM International Conference on Multimedia*, 1002–1011.

Mesaros, A.; and Virtanen, T. 2008. Automatic alignment of music audio and lyrics. In *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*.

North, A. C.; Krause, A. E.; and Ritchie, D. 2021. The relationship between pop music and lyrics: A computerized content analysis of the United Kingdom's weekly top five singles, 1999–2013. *Psychology of Music*, 49(4): 735–758.

Ou, L.; Ma, X.; and Wang, Y. 2023. Loaf-m2l: Joint learning of wording and formatting for singable melody-to-lyric generation. *arXiv preprint arXiv:2307.02146*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Pinkerton, R. C. 1956. Information theory and melody. *Scientific American*, 194(2): 77–87.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. *International Conference on Machine Learning*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Sheng, Z.; Song, K.; Tan, X.; Ren, Y.; Ye, W.; Zhang, S.; and Qin, T. 2021. Songmass: Automatic song writing with pre-training and alignment constraint. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15): 13798–13805.

Stefan Kostka, D. P. 1995. *Tonal Harmony with an Introduction to Twentieth-Century Music*. McGraw-Hill.

Suharto, S. 2004. Music and Language: A Stress Analysis of English Song Lyrics. *Harmonia: Journal of Arts Research and Education*, 5(3).

Tian, Y.; Narayan-Chen, A.; Oraby, S.; Cervone, A.; Sigurdsson, G.; Tao, C.; Zhao, W.; Chung, T.; Huang, J.; and Peng, N. 2023. Unsupervised Melody-to-Lyric Generation. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Watanabe, K.; Matsubayashi, Y.; Fukayama, S.; Goto, M.; Inui, K.; and Nakano, T. 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 (Long Papers)*, 163–172.

Wicaksana, H.; Hartono, S.; and Wei, F. S. 2006. Recognition of musical instruments. In *APCCAS 2006-2006 IEEE Asia Pacific Conference on Circuits and Systems*, 327–330. IEEE.

Wu, S.; Yu, D.; Tan, X.; and Sun, M. 2023. CLaMP: Contrastive Language-Music Pre-training for Cross-Modal Symbolic Music Information Retrieval. *International Society for Music Information Retrieval Conference*.

Zhang, R.; Mao, X.; Li, L.; Jiang, L.; Chen, L.; Hu, Z.; Xi, Y.; Fan, C.; and Huang, M. 2020. Youling: an AI-assisted Lyrics Creation System. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 85–91.

Zhou, Y.; Chu, W.; Young, S.; and Chen, X. 2019. BandNet: A Neural Network-based, Multi-Instrument Beatles-Style MIDI Music Composition Machine. 655–662.