IEICE TRANS. INF. & SYST., VOL.E108–D, NO.5 MAY 2025

PAPER

A Text-to-Lyrics Generation Method Leveraging Image-based Semantics and Reducing Plagiarism Risk*

Kento WATANABE[†a)], ***Nonmember and*** **Masataka GOTO**[†b)], ***Fellow***

SUMMARY This paper proposes a text-to-lyrics generation method, aiming to provide lyric writing support by suggesting the generated lyrics to users who struggle to find the right words to convey their message. Previous studies on lyrics generation have focused on generating lyrics based on semantic constraints such as specific keywords, lyric styles, and topics. However, these methods had limitations because users could not freely input their intentions as text. Even if such intentions can be given as input text, the lyrics generated from the input tend to contain similar wording, making it difficult to inspire the user. Our method is therefore developed to generate lyrics that (1) convey a message similar to the input text and (2) contain wording different from the input text. A straightforward approach of training a text-to-lyrics encoder-decoder is not feasible since there is no text-lyric paired data for this purpose. To overcome this issue, we divide the text-to-lyrics generation process into a two-step pipeline, eliminating the need for text-lyric paired data. (a) First, we use an existing text-to-image generation technique as a text analyzer to obtain an image that captures the meaning of the input text, ignoring the wording. (b) Next, we use our proposed image-to-lyrics encoder-decoder (I2L) to generate lyrics from the obtained image while preserving its meaning. The training of this I2L model only requires pairs of "lyrics" and "images generated from lyrics", which are readily prepared. In addition, we propose for the first time a lyrics generation method that reduces the risk of plagiarism by prohibiting the generation of uncommon phrases in the training data. Experimental results show that the proposed method can generate lyrics with phrasing different from the input text but conveying a message similar to that conveyed by the input text.

key words: *lyrics information processing, natural language processing, lyrics generation*

1. Introduction

Automatic lyrics generation methods are an important research topic in lyrics information processing [2]. With the aim of supporting users who already know what they want to convey in their lyrics but struggle to find the appropriate words, the methods are used in writing support systems providing them with generated lyrics as a source of new inspiration [3]–[9]. Most previous studies have focused on lyrics generation that is conditioned by semantic constraints, including specific keywords, lyric styles, and topics. Watanabe et al.'s system, for example, generates lyrics based on pre-defined topics selected by the user, but its limited range of topics results in similar styles of generated lyrics [3]. Oliveira et al.'s system generates poems based on keywords entered by the user, but it cannot generate poems based on sentences or paragraphs representing the user's intention [4], [5].

To provide more flexible lyric writing support, we propose generating lyrics based on freely formatted text entered by the user. We believe this approach surpasses the use of semantic constraints such as topics and keywords in terms of flexibility. While existing paraphrase systems [10] can be considered useful for this approach, the paraphrased lyrics may not provide sufficient inspiration because their wording tends to be similar to the input text. For example, even if a similar phrase "*Driving a car along the coastline*" is generated from the input text "*Driving a car on the seaside*", the user is unlikely to get new inspiration.

Therefore, the aim of this study is to develop a method for generating lyrics that not only have meanings similar to the input text but also use wording different from the input text. For example, if a user freely enters text that represents the content of the lyrics, such as "*Driving a car on the seaside*", our method generates lyrics with different wording, such as "*I'm driving in my car. But there's a beach of sand and the sea*". A simple way to achieve this aim would be to use Transformer-based encoder-decoders [11] for generating lyrics from text, but they require large amounts of text-lyric paired data for training, which are currently unavailable. To address this issue, we could use text summarization and machine translation to generate text from lyrics and obtain paired data automatically. However, since the generated text and lyric pairs have similar wording, an encoder-decoder trained using those paired data may generate lyrics with wording similar to the input text.

To achieve text-to-lyrics generation without using any paired text data for training, we propose a two-step pipeline framework: (a) using an existing text analyzer to obtain only the semantic representation from the input text, and (b) generating lyrics from the obtained representation. The core idea of this framework is to leverage a text-to-image generation technique such as Stable Diffusion XL [12] as the text analyzer. An image generated from the input text can serve as a reasonable intermediate representation that captures the meaning of the text while ignoring the details of its wording (Fig. 1 (a)). Using the generated image, our image-to-lyrics encoder-decoder generates semantically related lyrics (Fig. 1 (b)). It needs many image-lyric pairs as training data, but we can readily prepare those pairs by generating images

Manuscript received April 26, 2024.
Manuscript revised September 18, 2024.
Manuscript publicized November 13, 2024.

[†]National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba-shi, 305–8568 Japan.

*An earlier version of this paper was published at a conference [1].

a) E-mail: kento.watanabe@aist.go.jp
b) E-mail: m.goto@aist.go.jp

DOI: 10.1587/transinf.2024EDP7104

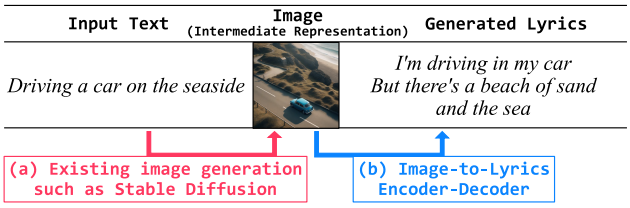


Fig. 1 Overview of the proposed text-to-lyrics generation method.

from lyrics of many songs. This is an advantage of using text-to-image generation. Another advantage is that it can generate images without regard to the input text's format, i.e., regardless of whether it is a word, phrase, sentence, or paragraph. We can thus provide flexible lyric writing support that is not constrained by the format of the input text.

Machine learning-based generation methods may inadvertently output portions of the training data directly without modification. Because such output can be considered plagiarism in some cases [13], [14], we also propose an anti-plagiarism method to reduce this risk. We assume that generating common phrases (word sequences having high commonness [15]) used in many songs is not plagiarism and reduce the risk of plagiarism by prohibiting the generation of uncommon phrases used in only a few songs. To the best of our knowledge, this is the first study to include such an anti-plagiarism method in lyrics generation.

Experimental results show that our text-to-lyrics generation method can generate lyrics with meaning similar to the input text but expressed differently. Another experiment shows that lyrics generated without using our anti-plagiarism method end up plagiarizing uncommon phrases from the training data, but those undesirable phrases can successfully be removed by our method.

2. Related Work

While natural language generation methods such as machine translations and chat systems have been actively studied and their performance greatly improved by deep neural networks (DNNs), automatic lyrics generation has also attracted attention as a research topic [2]. Most studies of lyrics generation have focused on lyric-specific musical constraints such as melody [16]–[21], rhyme [7], [9], [22]–[26], and audio signal [27]–[29]. While these lyric-specific musical constraints are an important aspect of lyrics generation, the main focus of this study is on the controllability of the semantic content of the generated lyrics.

Other studies have focused on lyrics generation that is conditioned by semantic constraints such as input keywords, styles, and topics [3]–[6], [30]–[33]. However, although these constraints allow some control over the semantic content of the generated lyrics, there may be differences between the user's intentions and the semantic content of the generated lyrics. To improve the usability of the lyrics generation method as a creative tool, we believe that users should be able to enter freely formatted text (words, phrases, sentences, paragraphs, etc.). Our proposed method therefore allows any text format, giving users greater control over the semantic content of the generated lyrics.

Some studies have proposed methods for generating lyrics that are semantically related to the input text [7], [8]. Ram et al. proposed a fine-tuned T5 model [10] that generates a single line of lyrics that comes after several lines of input lyrics [7]. Their method allows the user not only to enter sentences but also to control the rhyme and syllable count of the generated lyrics by adding special tokens at the end of the input sentence. In contrast to that method, in which the generated lyrics are a continuation of the input lyrics, ours generates lyrics that capture the semantic content of the input text. Zhang et al.'s research motivation is similar to ours, as they have also proposed a method for generating lyrics that capture the semantic content of the input text (which they refer to as passage-level text) [8]. To overcome the problem of the lack of text-lyric paired data for training the text-to-lyrics encoder-decoder, they collected lyrics data and passage-level text data (such as short novels and essays) separately and utilized an unsupervised machine translation framework. Specifically, they prepared two encoder-decoders, one for lyric text and one for passage-level text. They then aligned the latent representation space of these two encoder-decoders to build a text-to-lyrics encoder-decoder. In this paper, we propose a novel approach to develop a text-to-lyrics generation method that requires only lyrics data. While Zhang et al.'s method requires the collection of both lyrics and input texts, ours does not require additional text data, thus simplifying the development of the lyrics generation method.

3. Text-to-Lyrics Generation with Image-Based Semantics

As described in Sect. 1, the proposed text-to-lyrics generation method first generates an image from the input text by leveraging an existing text-to-image generation method. It then generates lyrics from the generated image by using our own image-to-lyrics encoder-decoder that we call *I2L*. Since the image serves as an intermediate representation of the input text's meaning, the generated lyrics can have a similar meaning but different wording. The network structure of the I2L is illustrated in Fig. 2. Assuming that one paragraph of lyrics can be represented in a single image, we set the unit of the generated lyrics to a paragraph.

For the text-to-image generation, we used the pretrained Emi 2.5[†] model, which is based on Stable Diffusion XL [12]. We selected this model because its developers state that it was trained on image data after excluding unauthorized images. We utilized Emi 2.5 to generate images using the prompt template, "*No text, visual representation of the following scene: [input text].*" To ensure that the generated images do not contain any textual elements, we used negative prompts such as "*text, signature, name, logo, transcription, words,*

[†]https://huggingface.co/aipicasso/emi-2-5

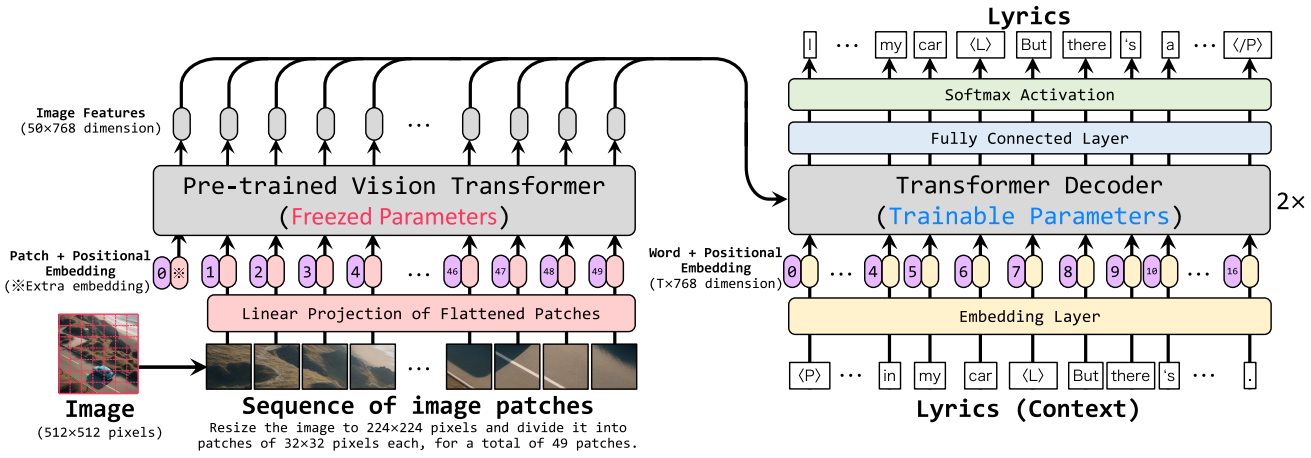


Fig. 2 Image-to-lyrics encoder-decoder (I2L) for generating lyrics from an image that is generated from the input text.

sentence, language, lyrics, phrase, paragraph, document." This exclusion of textual content helps maintain the focus on the visual representation of the input. Each image has a resolution of 512×512 and corresponds to a paragraph of the English lyrics.

As shown in Fig. 2, we then segment the generated image into 49 patches and compute the features of the image patches by using a pre-trained Vision Transformer† [34] to obtain 50 features (each with 768 dimensions) per image. These 50 image features are fed into the multi-head attention layer of the Transformer decoder [11]. We feed each word in a paragraph into the word embedding and positional embedding layers to compute the word vectors and feed each word vector into the masked multi-head attention layer of the Transformer decoder. The output of the Transformer decoder is fed into the fully connected layer FC to obtain a vector of vocabulary size dimensions. Finally, we apply the softmax activation function to this vector to calculate the word probability distribution.

3.1 Parameters

We use 768 as the number of embedding dimensions, 6 as the number of multi-heads, 2 as the number of decoder layers, 1024 as the number of feedforward layer dimensions, and GELU as the activation function. For optimization we use AdamW [35] with a mini-batch size of 8, a learning rate of 0.001, and a warm-up step of one epoch. Training is run for 40 epochs, and the I2L used for testing is the one that achieves the best loss on the development set.

We dare to train our Transformer decoder from scratch using only the lyrics data we have, without reusing available pre-trained large-scale language models (LLMs) such as BERT [36] or GPT-2 [37]. This is because when the training data of LLMs contain copyrighted literary works such as novels, poems, or essays, reusing pre-trained LLMs can result in plagiarizing those works. Since we would like to reduce the risk of plagiarism as described in Sect. 3.4, we cannot leverage pre-trained LLMs.

3.2 Training Data

We sample 129,747 English songs from the Music Lyrics Database V.1.2.7†† so that each song contains at least three paragraphs. The resulting dataset contains 927,535 paragraphs. This means that we can obtain 927,535 images by using Emi 2.5. We then split these songs into training (90%) and development (10%) sets. We use the top 52,832 words with the highest document-frequency as the vocabulary for training and convert the other words into a special symbol ⟨unknown⟩. This vocabulary includes ⟨L⟩ tags for line breaks, ⟨P⟩ tags for the beginnings of paragraphs, and ⟨/P⟩ tags for the ends of paragraphs.

We apply the same procedure not only to the lyrics of English songs but also to the lyrics of 142,772 Japanese songs sourced from a private dataset. This Japanese dataset contains 1,078,500 paragraphs, and the vocabulary size is 50,989 words. To extract word boundaries for Japanese lyrics, we apply the CaboCha parser [38]. Japanese lyrics are pre-translated into English by a Japanese-English translator††† for use with Emi 2.5. We use these English and Japanese lyrics datasets to train two I2Ls (one for each language).

3.3 Decoding Algorithm

We expect that generating and suggesting different variations of lyrics can give users new ideas for writing lyrics. To generate such different variations, we use a sampling method rather than a beam search method. In the sampling method, we sample each word according to the probability distribution calculated by the Transformer decoder. Sampling words according to a probability distribution allows a

†https://huggingface.co/google/vit-base-patch32-224-in21k

††https://www.odditysoftware.com/page-datasales1.htm

†††https://huggingface.co/staka/fugumt-en-ja

wide variety of words to be included in the generated lyrics, although some words that make the generated lyrics meaningless may be included. To avoid generating such meaningless lyrics, we use a Top-p sampling method that prohibits sampling words with low generation probabilities [39]. We can generate several lyrics simultaneously by running Top-p sampling in parallel. The probability distribution for word sampling in Top-p sampling is calculated using the formula $\mathrm{softmax}(\mathbf{z}/\tau)$, where $\mathbf{z}$ is the output of the fully connected layer FC and τ is the temperature parameter. If τ is less than 1, common words with high probability values are more likely to be sampled. In model training we set τ to 1, while in lyrics generation the user can set τ freely.

3.4 Anti-Plagiarism Method for Lyrics Generation

One of concerns with lyrics generation based on machine learning is the risk of plagiarism since the generated lyrics may contain phrases that are identical to existing lyrics phrases in training data, potentially leading to copyright infringement issues. To address these issues, we propose a method to reduce the risk of plagiarism in machine learning-based lyrics generation. This method not only allows the generation of new phrases that are not present in the training data but also permits the use of commonly used phrases such as "*I love you*" in the generated lyrics. In contrast, it prohibits the use of uncommon phrases that we consider to be a form of plagiarism. To achieve this, we create a list of uncommon phrases, $UncommonPhrase$, and prohibit the generation of phrases that are included in this list.

First, we define the uncommon phrases included in $UncommonPhrase$, as well as the new phrases and common phrases that are allowed to be generated. A phrase is defined by a word n-gram denoted by $\{w_1, \ldots, w_n\}$, where w is a word. We categorize a phrase as "new", "common", or "uncommon" according to $SN(\{w_1, \ldots, w_n\})$ defined as the number of songs in which the n-gram occurs in the training data:

- If $SN(\{w_1, \ldots, w_n\}) = 0$, this n-gram is a new phrase (i.e., it does not appear in the training data).
- If $3 < SN(\{w_1, \ldots, w_n\})$, this n-gram is a common phrase (i.e., it appears frequently in the training data).
- If $1 \leq SN(\{w_1, \ldots, w_n\}) \leq 3$, this n-gram is an uncommon phrase (i.e., it appears infrequently in the training data)[†].

Note that there is a possibility of mistaking uncommon phrases for common phrases when duplicate lyrics are contained in the training data, which results in SN values larger than they should be. It could happen when different artists sing the same lyrics, the same lyrics are repeatedly registered, and so on. We therefore identify duplicate lyrics according to the following two criteria: (1) we assume that pairs of lyrics with the same 20-grams are duplicates, and (2) we assume that pairs of lyrics with a normalized edit distance [40] of less than 0.5 are duplicates. To calculate SN accurately, we then concatenate the identified duplicate lyrics and replace those lyrics with the single concatenated lyrics. When lyrics that do not duplicate are mistaken for duplicate lyrics, a common phrase can be mistaken for an uncommon phrase, but this is better than vice versa from the anti-plagiarism viewpoint. This reduced the number of English songs in our lyrics data from 129,747 to 108,497. For Japanese lyrics, the number of songs was reduced from 142,772 to 119,595.

Using these SN criteria, we collect uncommon phrases from our training data. However, it is important to note that even if a word 3-gram is a common phrase, it may become an uncommon phrase when it becomes a word 4-gram. For instance, "*I love you*" is a common 3-gram with a large SN, while "*I love you darling*" is an uncommon 4-gram with a small SN. Therefore we do not use a single value of n but instead consider all values of n within a range from 1 to sufficiently large values. However, it is difficult to store all uncommon phrases in memory because the number of n-grams that have to be listed increases with n. To overcome this memory limitation problem, we propose to use the following procedure to minimize the number of uncommon phrases we need to store in memory: (1) we start by examining 1-grams, then move on to 2-grams, 3-grams, and so on until we have looked at all possible n-grams in the training data. (2) For each target n-gram, we generate all possible sub-n-grams of length $1, 2, \ldots, n-1$. If any of these sub-n-grams are already in $UncommonPhrase$, we can skip adding the target n-gram to $UncommonPhrase$ because we know it is uncommon. Otherwise, we add the target n-gram to $UncommonPhrase$. Following this procedure, we collected approximately 22.3M uncommon n-grams with n ranging from 1 to 21 for English lyrics. For Japanese lyrics, we collected approximately 18.2M uncommon n-grams with n ranging from 1 to 19.

After creating $UncommonPhrase$ using the above procedure, we prohibit their generation during Top-p sampling by the following two steps: (1) During word generation, we check whether any sub-n-grams derived from the word sequence $\{w_1, \ldots, w_t\}$ are included in $UncommonPhrase$. (2) If any of these sub-n-grams are found in $UncommonPhrase$, we prohibit the generation of word w_t by setting its generation probability $P(w_t|\{w_1, \ldots, w_{t-1}\})$ to zero.

4. Quantitative Evaluation

The proposed text-to-lyrics generation method was quantitatively evaluated using three metrics:

Test-set perplexity (PPL): This is a standard evaluation measure for encoder-decoders. It measures the degree of predictability of the phrasing in the original text in the test set [41]. A smaller PPL value is better because it indicates that the encoder-decoder has a higher ability to generate lyrics that capture the meaning of the input text.

[†]In this study, we tentatively set the threshold for SN at 3. Since there is no established legal rule, we believe that this threshold will be determined by social consensus in the future. Providing the technical basis for discussions establishing such a consensus is also a contribution of this study.

Normalized edit distance (NED): The normalized edit distance [40] between the generated lyrics and the input text is calculated to evaluate whether the proposed method generates lyrics that differ in wording from the input text. A larger NED is better because it indicates that the generated lyrics have wording that is more different from the input text.

BERTScore difference (DiffBS): BERTScore is an effective metric for measuring semantic similarity between input text and generated lyrics. However, when the wording of the generated lyrics closely resembles that of the input text, the BERTScore tends to be artificially high. This is not consistent with our goal of generating lyrics that are semantically related to the input text but differ from it in wording. Therefore, we employ DiffBS = |BS_real−BS_gen|, calculated as the absolute difference between two BERTScores, BS_real and BS_gen. BS_real is the BERTScore between the original (actual) lyrics and the input text, and BS_gen is the BERTScore between the generated lyrics and the input text. A small DiffBS value indicates that the semantic similarity between the generated lyrics and the input text is close to that between the original lyrics and the input text. This novel metric is designed to assess how well the generated lyrics capture the essence of the input text without merely replicating its wording.

4.1 Experimental Dataset

To evaluate the proposed lyrics generation method, we constructed a test dataset consisting of pairs of a lyric paragraph and input text representing the semantic content of the lyrics. For English songs, we randomly selected 20 Disney animated films and used their plot summaries taken from Wikipedia as the input text, along with their corresponding theme song lyrics, resulting in a total of 125 lyric paragraphs. We here assume that the lyrics of each theme song are based on the content of the corresponding film.

For Japanese songs, the test dataset included plot summaries and theme song lyrics from 51 Japanese animated series, totaling 620 lyric paragraphs. These were selected from a well-known Japanese website that hosts user-generated novels[†]. We chose novels that were popular (widely viewed) and had later been adapted into animated series. We used each novel's plot summary as the input text, assuming that the summary is reflected in the theme song lyrics of its corresponding animated series.

4.2 Methods Compared

To compare the proposed method with possible different methods, we prepared the following encoder-decoders trained on paired data created in different suitable ways.

Image-to-Lyrics encoder-decoder (I2L) This is the proposed encoder-decoder trained on image-lyric paired data.

Summary-to-Lyrics encoder-decoder (S2L) We converted each lyric paragraph in the training data into a summary using a text summarization method[††] to create summary-lyric paired data. The data was then used to train a Transformer-based summary-to-lyric encoder-decoder.

Back-translated-lyrics-to-Lyrics encoder-decoder (B2L) We translated each lyric paragraph in the training data from English to Japanese to English by using English-Japanese and Japanese-English translation methods[†††] to create paired data of the back-translated lyrics and the original lyrics. The data was then used to train a Transformer-based back-translated-lyrics-to-lyrics encoder-decoder.

Half-to-Half encoder-decoder (H2H) The H2H model was developed to tackle the difficulty of training a text-to-lyrics model without pairs of input text and corresponding lyrics. Following the approach in [7], we split each lyric paragraph into its first and second halves. Although this split does not mean that the halves convey exactly the same content, they are related to the same topic. By training a Transformer-based encoder-decoder to generate the second half from the first half, we aim to achieve our goal of generating lyrics that are topically related to the input but distinct in wording from it.

ChatGPT4o-mini As a cutting-edge comparative method, we adapted the ChatGPT4o-mini for the text-to-lyrics task by using the following prompt: "*You are a creative assistant tasked with generating song lyrics. The lyrics should be 2–5 lines long, forming a single paragraph. Please generate English (or Japanese) lyrics that are imaginative and reflective of the user's prompts.*"

Since the above S2L, B2L, and H2H are also Transformer-based encoder-decoders, their parameter settings are the same as for the proposed I2L. Given one input text, five lyrics were generated by each method. The parameter p for Top-p sampling was set to 0.9 and τ was set to 0.4. The generation process stops when the symbol ⟨/P⟩ (end of paragraph) is generated. In this comparison we did not use the proposed anti-plagiarism method.

4.3 Experimental Results

Table 1 shows the evaluation results for the proposed I2L method and several comparative methods, including ChatGPT4o-mini, which because of API limitations cannot compute PPL. The proposed I2L method showed the lowest PPL values in both English and Japanese experiments ($p < 0.05$ based on t-tests). This indicates its superior ability to generate original lyrics that are not only grammatically correct and readable but are also semantically related to the input text. In contrast, the B2L method had the highest PPL values, indicating a strong tendency to generate lyrics that were grammatically incorrect and semantically incoherent.

[†]https://syosetu.com

[††]https://huggingface.co/google/pegasus-xsum for the English summarization. https://huggingface.co/tsmatz/mt5_summarize_japanese for the Japanese summarization.

[†††]https://huggingface.co/staka/fugumt-en-ja for the English to Japanese translation. https://huggingface.co/staka/fugumt-ja-en for the Japanese to English translation.

Table 1 Results of quantitative evaluation. The arrow (↓) for PPL and DiffBS indicates that smaller values are better, while the arrow (↑) for NED indicates that larger values are better. The arrow (→) for BERTScore shows that values closer to the Real value are more desirable.

Method	English				Japanese			
	PPL ↓	NED ↑	BERTScore →	DiffBS ↓	PPL ↓	NED ↑	BERTScore →	DiffBS ↓
Real	-	-	0.338	-	-	-	0.426	-
I2L w/o the anti-plagiarism method	**73.83**	**0.78**	0.326	0.012	**198.9**	**0.93**	**0.442**	**0.016**
S2L	346.73	0.69	0.353	0.015	306.19	0.86	0.480	0.054
B2L	544.21	0.71	**0.349**	**0.011**	1051.58	0.66	0.519	0.093
H2H	163.98	0.69	0.390	0.012	583.13	0.90	0.490	0.064
ChatGPT4o-mini	-	0.58	0.477	0.139	-	0.83	0.567	0.141

Furthermore, the NED between the lyrics generated by the I2L method and the input text was the largest ($p < 0.05$ based on t-tests), suggesting that the I2L method is more likely to generate lyrics with wording different from the input text. On the other hand, lyrics generated by ChatGPT4o-mini had the smallest NED, indicating a tendency to generate lyrics with wording similar to the input text.

Regarding BERTScore and DiffBS, the BERTScore between the input text and the human-created lyrics (BS_real) was 0.338 for English and 0.426 for Japanese. The BERTScore between the generated lyrics and the input text (BS_gen) from the I2L method was very close to BS_real, at 0.326 for English and 0.442 for Japanese. The resulting DiffBS values were 0.012 for English and 0.016 for Japanese, indicating that the semantic similarity between the generated lyrics and the input text was close to that of the original lyrics and the input text. When comparing the proposed I2L method with other methods, the Japanese lyrics from I2L exhibited the smallest DiffBS, while in English, S2L, B2L, and H2H showed competitive results.

These results confirm that image-lyric pairs, as utilized in the I2L method, are more effective than other paired data sets as training data for encoder-decoders generating lyrics that are semantically related to the input text but differ from it in wording.

Additionally, the performance of ChatGPT4o-mini showed low NED and high BERTScore, indicating that the lyrics often replicated the wording of the input text, resulting in higher BERTScores. This result confirms that the proposed method is more suitable for our goal than ChatGPT4o-mini.

5. Effectiveness of the Proposed Anti-Plagiarism Method

We examined whether the absence of the anti-plagiarism method proposed in Sect. 3.4 results in plagiarizing uncommon phrases found in existing lyrics. In the lyrics generated by the I2L method in Sect. 4, we calculated the percentage of n-grams included in $UncommonPhrase$.

The results with n ranging from 1 to 18 are shown in Fig. 3. The percentage of uncommon 1-grams and 2-grams in the generated lyrics is almost 0%. This indicates that almost all of the generated 1-grams and 2-grams are common phrases used in many existing lyrics, even without the use of the anti-plagiarism method. On the other hand, the percentage of uncommon 3-grams to 8-grams ranged between 3% and 16%. This suggests that many phrases in the generated lyrics are likely to plagiarize if the proposed anti-plagiarism method is not used. Furthermore, as n increases beyond 9, the n-gram combinations become so numerous that the generated n-grams are rarely included in $UncommonPhrase$. These results confirm that our machine learning-based lyrics generation method tends to sample common words, but the generated 3- to 8-gram phrases, even though they are composed of common words, may be uncommon enough to raise suspicion of plagiarism. Using the proposed anti-plagiarism method, in contrast, ensures that uncommon phrases contained in $UncommonPhrase$ are never generated, thereby reducing the risk of plagiarism.

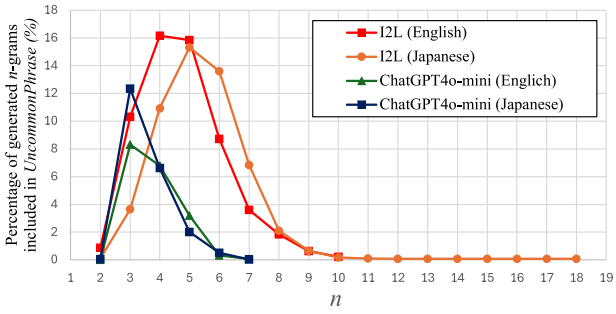


Fig. 3 The percentage of generated lyric n-grams that are included in $UncommonPhrase$, a list of phrases that should not be generated (plagiarized). For example, 16.2% at English 4-grams means that among all 4-gram phrases in the generated lyrics, 16.2% are uncommon phrases, though 83.8% are new or common phrases.

To support our results, we conducted a similar analysis on lyrics generated by ChatGPT4o-mini. We found that 19% of the n-grams in the English lyrics generated by ChatGPT4o-mini and 22% of those in the Japanese lyrics generated by ChatGPT4o-mini were also in the $UncommonPhrase$. These findings, shown in Fig. 3, reveal that even large language models can generate lyrics that might include potentially plagiaristic phrases.

In additional experiments, we evaluated the impact of our anti-plagiarism method on the quantitative metrics NED and DiffBS. The results, as shown in Table 2, indicate that applying the anti-plagiarism method affects the performance on these metrics only slightly. This suggests that it has little impact on the wording differences or semantic similarity between the input text and the generated lyrics. PPL could not be calculated in this case, as the anti-plagiarism

Table 2 Impact of the proposed anti-plagiarism method on NED and DiffBS.

Method	English			Japanese		
	NED ↑	BERTScore →	DiffBS ↓	NED ↑	BERTScore →	DiffBS ↓
Real	-	0.338	-	-	0.426	-
I2L w/o the anti-plagiarism method	0.78	0.326	0.012	0.93	0.442	0.016
I2L with the anti-plagiarism method	0.76	0.325	0.013	0.92	0.446	0.020

Table 3 Examples of lyrics generated from the input text by using the proposed text-to-lyrics generation method with the anti-plagiarism method.

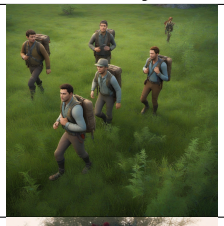
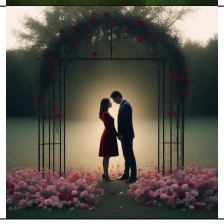
Input text	Image (intermediate representation)	Generated lyrics
A group of explorers are walking through the grass neutral.	⇒  ⇒	Running in the fields of grass We were running from a hill to thrill Where I was born on guard, but it's not easy They said that you're never gonna be sorry
We meet again I guess our love is forever.	⇒  ⇒	Love is a garden of Eden All alone in this world we live on forever I'm gonna live it all for you Is the love that it takes to get better

Table 4 Qualitative evaluation of semantic similarity for different pairs.

Metric	Our method	ChatGPT4o-mini
Percentage of evaluator judgments of input text and generated lyrics as similar	53/100 (53%)	72/100 (72%)
Percentage of evaluator judgments of input text and intermediate image as similar	85/100 (85%)	-
Percentage of evaluator judgments of intermediate image and generated lyrics as similar	63/100 (63%)	-

method sets the generation probability of any phrase in the *UncommonPhrase* to zero, resulting in an infinite PPL value.

While the proposed anti-plagiarism method is effective, it is important to note that it is not intended to be a foolproof solution that ensures legal compliance. Rather, it is designed to provide a helpful guideline for those who wish to generate original lyrics while reducing the risk of plagiarism. We hope that our approach will contribute to further discussions on a reasonable balance between encouraging creativity and respecting intellectual property rights.

6. Qualitative Evaluation

Table 3 shows two examples of lyrics generated using the proposed method with the anti-plagiarism method. Given the input text, our method can generate any number of lines of lyrics, but here four lines were generated by stopping the generation process when four ⟨L⟩ (line break) symbols and the ⟨/P⟩ (end of paragraph) symbol were generated. In the first example, the input text was taken from the SICK dataset [42], while in the second example the input text was taken from lyrics in the RWC Music Database [43]. In both examples, our method generated lyrics that reflected the content of the input text. In the first example, it generated an image that represented the scene described in the input text and generated corresponding lyrics that reflected the image. In contrast, in the second example, our method generated an image of a person with an emotional expression corresponding to the input text and generated lyrics that express the emotion depicted in the image. Other examples can be found in Appendix A.

In addition to the quantitative evaluation and the generated examples, we conducted a comprehensive human evaluation to assess the similarity not only between the input text and the generated lyrics but also between the input text and the intermediate image, as well as between the intermediate image and the generated lyrics. Furthermore, to compare the effectiveness of our proposed method, we also evaluated the similarity between the input text and the lyrics generated by ChatGPT4o-mini.

To prepare the input text in an objective way, we collected the 100 titles of the "Hot 100 Songs" in 2022 on the Billboard year-end charts[†], extracted the first verse from their lyrics, and summarized each verse into a short sentence using ChatGPT[††]. This input text was then used to generate an intermediate image, which was subsequently used to

[†]https://www.billboard.com/charts/year-end/2022/hot-100-songs/

[††]We entered a prompt like "*Rephrase the following text into a short sentence.*" into ChatGPT-3.5 (https://chat.openai.com/chat) on March 31, 2023.

generate the final lyrics using the proposed method with the anti-plagiarism method. For the evaluation, we showed three pairs to an evaluator: (1) the input text and the generated lyrics, (2) the input text and the intermediate image, and (3) the intermediate image and the generated lyrics. The evaluator was asked to determine whether the impressions from each pair were similar or not. This comprehensive evaluation aimed to assess not only the semantic consistency between the input and the generated lyrics but also the effectiveness of the image generation process and the image-to-lyrics generation process. Additionally, the evaluator compared the input text with the lyrics generated by ChatGPT4o-mini to evaluate its performance.

Table 4 shows that the impressions of the input text and the generated lyrics were judged to be similar in 53 of the 100 cases for our method. Although ChatGPT4o-mini achieved more similar impressions (72 out of 100 cases), as expected from its higher BERTScore in Table 1, our goal of generating lyrics that are semantically related to the input text but differ from it in wording could not be fulfilled. Despite the current similarity judgments, our method has the potential to be useful as a writing support tool in many situations where users' intentions can be represented as images, and it may also offer value in pioneering a novel approach to lyric generation.

With only 53 cases yielding similar impressions in our method, the result suggests the potential for semantic drift during the text-to-image-to-lyrics generation process. As shown in Table 4, 85 out of 100 cases had a high degree of similarity between the input text and the intermediate image, demonstrating strong image generation capabilities. However, only 63 cases had a high degree of similarity between the intermediate image and the generated lyrics. This gap highlights the need for improvements in the image-to-lyrics generation method in the future.

7. Conclusion

This paper has described a method for generating lyrics that are similar in meaning to the input text but expressed differently. The contributions of this study are as follows:

1. We proposed a novel two-step pipeline framework. First, we apply text-to-image generation as a text analyzer to extract only the semantic content from the input text. Next, we use our proposed image-to-lyrics encoder-decoder to generate lyrics that capture the semantics of the generated image.
2. We proposed a method to reduce the risk of plagiarism by prohibiting the generation of uncommon phrases in the training data and verified its effectiveness.
3. We quantitatively showed that our proposed method outperforms other methods generating lyrics.

Future work will develop a flexible lyric writing support system using the proposed lyrics generation method.

Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP20K19878, Japan.

References

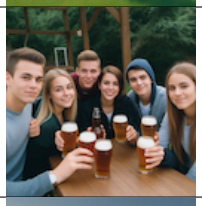
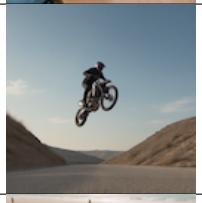
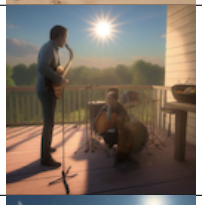
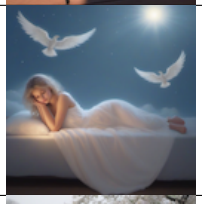
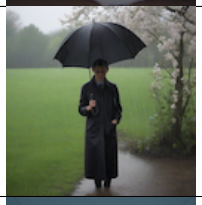
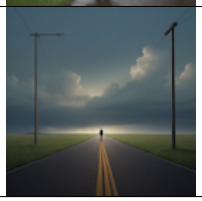
[1] K. Watanabe and M. Goto, "Text-to-lyrics generation with image-based semantics and reduced risk of plagiarism," Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR), pp.398–406, 2023.
[2] K. Watanabe and M. Goto, "Lyrics information processing: Analysis, generation, and applications," Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA), pp.6–12, 2020.
[3] K. Watanabe, Y. Matsubayashi, K. Inui, T. Nakano, S. Fukayama, and M. Goto, "LyriSys: An interactive support system for writing lyrics based on topic transition," Proceedings of the 22nd International Conference on Intelligent User Interfaces (ACM IUI), pp.559–563, 2017.
[4] H.G. Oliveira, T. Mendes, and A. Boavida, "Co-PoeTryMe: A co-creative interface for the composition of poetry," Proceedings of the 10th International Conference on Natural Language Generation (INLG), pp.70–71, 2017.
[5] H.G. Oliveira, T. Mendes, A. Boavida, A. Nakamura, and M. Ackerman, "Co-PoeTryMe: Interactive poetry generation," Cognitive Systems Research, vol.54, pp.199–216, 2019.
[6] R. Zhang, X. Mao, L. Li, L. Jiang, L. Chen, Z. Hu, Y. Xi, C. Fan, and M. Huang, "Youling: An AI-assisted lyrics creation system," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), pp.85–91, 2020.
[7] N. Ram, T. Gummadi, R. Bhethanabotla, R.J. Savery, and G. Weinberg, "Say what? collaborative pop lyric generation using multitask transfer learning," Proceedings of the 9th International Conference on Human-Agent Interaction (HAI), pp.165–173, 2021.
[8] L. Zhang, R. Zhang, X. Mao, and Y. Chang, "QiuNiu: A Chinese lyrics generation system with passage-level input," Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics - System Demonstrations (ACL), pp.76–82, 2022.
[9] N. Liu, W. Han, G. Liu, D. Peng, R. Zhang, X. Wang, and H. Ruan, "ChipSong: A controllable lyric generation system for Chinese popular song," Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing), pp.85–95, 2022.
[10] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P.J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," Journal of Machine Learning Research, vol.21, no.140, pp.1–67, 2020.
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol.30, pp.1–11, 2017.
[12] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "SDXL: improving latent diffusion models for high-resolution image synthesis," The Twelfth International Conference on Learning Representations (ICLR), 2024.
[13] A. Papadopoulos, P. Roy, and F. Pachet, "Avoiding plagiarism in markov sequence generation," Proceedings of the 28th AAAI Conference on Artificial Intelligence, vol.28, no.1, pp.2731–2737, 2014.
[14] Q. Feng, C. Guo, F. Benitez-Quiroz, and A.M. Martínez, "When do GANs replicate? on the choice of dataset size," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp.6681–6690, 2021.
[15] T. Nakano, K. Yoshii, and M. Goto, "Musical similarity and commonness estimation based on probabilistic generative models of mu-

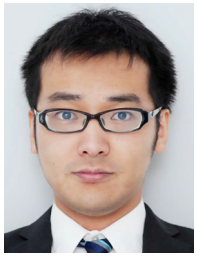
sical elements," International Journal of Semantic Computing (IJSC), vol.10, no.1, pp.27–52, 2016.
[16] K. Watanabe, Y. Matsubayashi, S. Fukayama, M. Goto, K. Inui, and T. Nakano, "A melody-conditioned lyrics language model," Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp.163–172, 2018.
[17] X. Lu, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A syllable-structured, contextually-based conditionally generation of Chinese lyrics," Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence (PRICAI), pp.257–265, 2019.
[18] Y. Chen and A. Lerch, "Melody-conditioned lyrics generation with SeqGANs," Proceedings of the 2020 IEEE International Symposium on Multimedia (ISM), pp.189–196, 2020.
[19] Y.-F. Huang and K.-C. You, "Automated generation of Chinese lyrics based on melody emotions," IEEE Access, vol.9, pp.98060–98071, 2021.
[20] X. Ma, Y. Wang, M.-Y. Kan, and W.S. Lee, "AI-Lyricist: Generating music and vocabulary constrained lyrics," Proceedings of the 29th ACM International Conference on Multimedia (ACM-MM), pp.1002–1011, 2021.
[21] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, and T. Qin, "SongMASS: Automatic song writing with pre-training and alignment constraint," Proceedings of the 35th AAAI Conference on Artificial Intelligence, vol.35, no.15, pp.13798–13805, 2021.
[22] G. Barbieri, F. Pachet, P. Roy, and M.D. Esposti, "Markov constraints for generating lyrics with style," Proceedings of the 20th European Conference on Artificial Intelligence (ECAI), pp.115–120, 2012.
[23] J. Hopkins and D. Kiela, "Automatically generating rhythmic verse with neural networks," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), pp.168–178, 2017.
[24] E. Manjavacas, M. Kestemont, and F. Karsdorp, "Generation of hip-hop lyrics with hierarchical modeling and conditional templates," Proceedings of the 12th International Conference on Natural Language Generation (INLG), pp.301–310, 2019.
[25] L. Xue, K. Song, D. Wu, X. Tan, N.L. Zhang, T. Qin, W.-Q. Zhang, and T.-Y. Liu, "DeepRapper: Neural rap generation with rhyme and rhythm modeling," Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP), pp.69–81, 2021.
[26] J.-W. Chang, J.C. Hung, and K.-C. Lin, "Singability-enhanced lyric generator with music style transfer," Computer Communications, vol.168, pp.33–53, 2021.
[27] O. Vechtomova, G. Sahu, and D. Kumar, "Generation of lyrics lines conditioned on music audio clips," Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA), pp.33–37, 2020.
[28] O. Vechtomova, G. Sahu, and D. Kumar, "LyricJam: A system for generating lyrics for live instrumental music," Proceedings of the 12th International Conference on Computational Creativity (ICCC), pp.122–130, 2021.
[29] K. Watanabe and M. Goto, "Atypical lyrics completion considering musical audio signals," Proceedings of the 27th International Conference on Multimedia Modeling (MMM), pp.174–186, 2021.
[30] K. Watanabe, Y. Matsubayashi, K. Inui, and M. Goto, "Modeling structural topic transitions for automatic lyrics generation," Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC), pp.422–431, 2014.
[31] P. Potash, A. Romanov, and A. Rumshisky, "GhostWriter: Using an LSTM for automatic rap lyric generation," Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1919–1924, 2015.
[32] M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight, "Generating topical poetry," Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1183–1191, 2016.
[33] H. Fan, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A hierarchical attention based seq2seq model for Chinese lyrics generation," Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence (PRICAI), pp.279–288, 2019.
[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," Proceedings of the 9th International Conference on Learning Representations (ICLR), 2021.
[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," Proceedings of the 7th International Conference on Learning Representations (ICLR), 2019.
[36] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp.4171–4186, 2019.
[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI blog, vol.1, no.8, p.9, 2019.
[38] T. Kudo and Y. Matsumoto, "Japanese dependency analysis using cascaded chunking," Proceedings of the 6th Conference on Natural Language Learning (CoNLL), vol.20, pp.1–7, 2002.
[39] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," Proceedings of the 8th International Conference on Learning Representations (ICLR), 2020.
[40] Y. Li and B. Liu, "A normalized Levenshtein distance metric," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.29, no.6, pp.1091–1095, 2007.
[41] C. Manning and H. Schutze, Foundations of statistical natural language processing, MIT press, 1999.
[42] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A SICK cure for the evaluation of compositional distributional semantic models," Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC), pp.216–223, 2014.
[43] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, classical and jazz music databases," Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR), pp.287–288, 2002.

Appendix A: Examples of Input Text, Intermediate Image, and Generated Lyrics

In addition to the two examples shown in Table 3, eight other examples of lyrics generated by the proposed text-to-lyrics generation method with the anti-plagiarism method are shown in Table A· 1. The top four input texts are sentences selected from the SICK dataset [42], and the bottom four input texts are parts of lyrics selected from the RWC Music Database [43]. Details on how to generate these examples are described in Sect. 6.

Table A·1 Examples of input text, intermediate image, and generated lyrics.

Input text	Image (intermediate representation)	Generated lyrics
The kids are playing outdoors near a man with a smile	⇒  ⇒	You're living in a world full of fun A boy, you can have it all come back home to everyone I'll be the one that did not come from anyone who's been? But if there was someone somewhere out there at your own house now!
Several young people are posing for a photo and holding beers	⇒  ⇒	Let's go, let's go out with the beer We got a lot of friends that can do without you? It's like I never felt so good at all But it's not that easy when your heart is breaking
A lone biker is jumping in the air	⇒  ⇒	It's a beautiful day... I'm gonna ride high, ride, die! for me to die! Where's the sky? My heart is racing down my spine
People are on a beach full of sand by the ocean and are enjoying a day full of sun	⇒  ⇒	There's a party in white sand And the people that we have found out of here It's an endless beach, it's not your secret We're just as young as they can be... oh yeah
The sun is up on a brand new day I've got to face the music	⇒  ⇒	When I was young, a heart full of music The sun would rise up and fall on me again But when my guitar sings the blues away They say it's true but they don't know how to play
Last night I dreamt of an angel Who flew over me, and I saw beauty	⇒  ⇒	I'm dreaming of an angel And a white dove and she's so far above the law But it doesn't matter how much love's been done She is my only one, oh yeah
In the Spring when gentle rains turn to showers	⇒  ⇒	Rain, rain come down and bring me pain Come wash away the pain from my love again... yeah! Oh baby please come with me now? Do I have to face these rainy days - yes it will be okay
Standing at the crossroads I hear a voice Asking about tomorrow and where I'm going	⇒  ⇒	If you're not alone, It would be a long long walk home? I'd rather go there in the wrong direction and leave me alone And if it takes all night long



Kento Watanabe received his Ph.D. from Tohoku University in 2018. He is currently a senior Researcher at National Institute of Advanced Industrial Science and Technology (AIST), Japan. His research interests include lyrics information processing, natural language processing, machine learning, and human-computer interaction.

Masataka Goto received the Doctor of Engineering degree from Waseda University in 1998. He is currently a Principal Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. Over the past 32 years he has published more than 350 papers in refereed journals and international conference proceedings and has received 69 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and the Tenth JSPS PRIZE. He has served as a committee member of over 140 scientific societies and conferences, including as the General Chair of ISMIR 2009 and 2014. As the research director, he began the OngaACCEL project in 2016 and the RecMus project in 2021, which are five-year JST-funded research projects (ACCEL and CREST) related to music technologies.