Syllable-level lyrics generation from melody exploiting character-level language model

Zhe Zhang[1], Karol Lasocki [2†], Yi Yu[1*], Atsuhiro Takasu[1]
National Institute of Informatics, SOKENDAI[1]
Aalto University[2]
{zhe, yiyu, takasu}@nii.ac.jp, karolasocki@gmail.com

Abstract

The generation of lyrics tightly connected to accompanying melodies involves establishing a mapping between musical notes and syllables of lyrics. This process requires a deep understanding of music constraints and semantic patterns at syllable-level, word-level, and sentence-level semantic meanings. However, pre-trained language models specifically designed at the syllable level are publicly unavailable. To solve these challenging issues, we propose to exploit fine-tuning character-level language models for syllable-level lyrics generation from symbolic melody. In particular, our method endeavors to incorporate linguistic knowledge of the language model into the beam search process of a syllable-level Transformer generator network. Additionally, by exploring ChatGPT-based evaluation for generated lyrics, along with human subjective evaluation, we demonstrate that our approach enhances the coherence and correctness of the generated lyrics, eliminating the need to train expensive new language models.

1 Introduction

Generating lyrics from a given melody is a subjective and creativity-driven process that does not have a definitive correct answer. Recognizing the importance of subjective and creativity-driven generation processes is essential for advancing the development of AI. By embracing and enabling such processes, we can pave the way for more nuanced and expressive AI-generated lyrics. Accordingly, evaluating the quality of subjectively and creativity-driven generated lyrics has become a fascinating topic. Our system focuses on generating lyrics from symbolic melodies and could serve as a valuable creative aid, collaborating with artists throughout the entire songwriting process. The use of symbolic melodies allows for effortless and frequent modifications, facilitating iterative creative exploration.

In this work, we explore the generation of lyrics from simplified symbolic melodies consisting of 20 notes. Our aim is to maintain the alignment between the syllables of the lyrics and the corresponding melody notes during the inference stage. To achieve this, we propose a melody-encoder-syllable-decoder Transformer architecture, which generates syllables sequentially in accordance with the melody. However, due to the scarcity of paired lyrics-melody data available for training, this approach could lead to producing lyrics that are not coherent and grammatically not correct, such as "*you gotta o in what the you used to life*".

The dataset we are using is described in (Yu et al., 2021), and it only contains approximately 10,000 paired lyrics-melody sequences. Each lyrics sequence in the dataset contains 20 syllables in length, and there may be samples where syllables are occasionally missing due to misalignment, or lack of corresponding notes. These problems significantly hinder the training of a model to comprehend and generate coherent language.

On the other hand, due to the constraint of syllable-level generation, it is difficult to directly apply pre-trained language models that already have an understanding of linguistic knowledge, due to the scarcity of syllable-level language models. The utilization of the widely popular word-piece encoding is not feasible in our task because one word consists of different numbers of syllables. This would potentially affect the probabilities of generating multi-syllable words. A possible alternative approach to train a custom language model at the syllable level is using a large, clean text corpus that has been segmented into syllable-level texts, which can then be fine-tuned specifically for the task of generating lyrics, but it is also difficult to construct such kind of dataset. Another solution is to fine-

*Yi Yu is the corresponding author.
†Karol was involved in this work during the internship at National Institute of Informatics (NII), Tokyo.

Findings of the Association for Computational Linguistics: EACL 2024, pages 1336–1346
March 17-22, 2024

tune a character-level language model, refining it to generate syllable sequences. In this work, we focus on the latter approach, which aims to fine-tune a character-level language model for re-ranking the candidates generated by a melody-encoder-syllable-decoder Transformer (Vaswani et al., 2017).

We take inspiration from the usage of language models in re-ranking speech recognition token candidates (Bühler et al., 2005). Considering the sentence "*Last* ***x*** *was windy*", and the speech recognition system candidates *knight* and *night*. Due to the pronunciation similarities, the word *knight* could be given a higher probability when recognizing speech. However, a language model would easily fix the mistake, assigning a higher probability to the word *night* instead.

Another inspiring work by Wang et al. (2021) focused on video comment generation tasks, In this work, the probability of previous text token, the probability of future text token, and the mutual dependency between comment texts and video are modeled by three separately trained neural networks. The probabilities from all three models are then combined and the best candidate from the main comment generation Transformer model is selected, improving coherence and relation between comments and video.

In our study, using a real example from our models, given the sentence "*you gotta*", the lyrics generation model could predict possible next tokens as *o* rather than *treat* because of the limited training data it learned from, but it is neither grammatically correct nor semantically meaningful. In this case, a powerful language model would know that the latter is more likely to form a coherent sentence. Using a fine-tuned language model to refine the semantic meanings within generated syllable-level lyrics, we are able to improve the generated sequence from "*you gotta o in what the you used to life*" to "*you gotta treat me to maybe understand you*". As one phrase of lyrics, the revised sequence is much more coherent and interesting than the original version.

The main contributions of this work can be summarized as follows:

1. Training a melody-encoder-syllable-decoder Transformer model to generate lyrics syllable by syllable, ensuring semantic correlation with individual notes in the melody.

2. Proposing exploiting the fine-tuned character-level pre-trained language models for refining candidate syllables generated by the Transformer decoder to ensure the coherence and correctness in the generated lyrics, overcoming the difficulty of unavailable pre-trained syllable-level language models.

3. Designing a beam search and re-ranking technique to integrate the fine-tuned language model with the Transformer decoder to predict re-ranked lyrics candidates.

2 Proposed methods

By exploiting fine-tuning a pre-trained language model, we have successfully designed syllable-level lyrics generation architecture from symbolic melody exploiting character-level language model depicted in Figure 1. In this section, we will introduce the details of the proposed methods.

2.1 Syllable-level lyrics generation from melody

As shown in Figure 1, the Transformer on the right side generates the candidate syllable tokens based on the encoded melody latent representations M and previously generated lyrics. The fine-tuned language model on the left evaluates the probability of the candidates based on the given lyrics generated, which aims to improve the coherence and correctness of the generated lyrics.

As an example shown in Figure 1, the proposed model has generated a sequence of lyrics tokens *don't get any big* in previous time steps. In the current time step, the Transformer decoder predicts syllable *ger* with a probability of 0.3 and predicts syllable *ideas* with a probability of 0.2. Considering the Transformer is trained on a limited amount of data, it might assign a higher probability to *ger* because the syllables can construct a word *bigger*. However, the language model, which is trained on a large amount of corpus, can predict *ideas* with a higher probability of 0.6 because the sentence *don't get any big ideas* is more meaningful in natural language. Then, in the re-ranking stage, token *ideas* can be assigned the highest probability after weighting the two probabilities. In such a way, the language model can help the Transformer generator predict better lyrics in terms of grammar and meaning.

We focus on exploiting the language model in Figure 1, hoping to improve the coherence and correctness of the lyrics generated by the main model

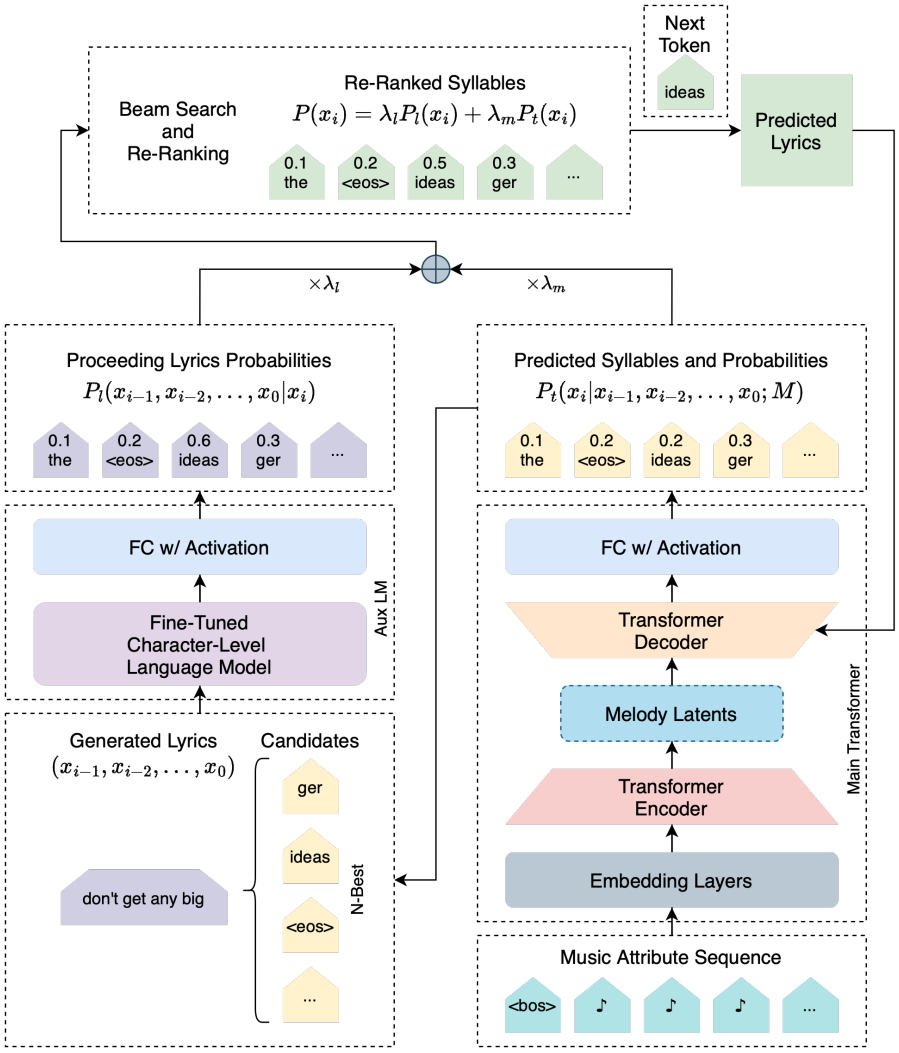


Figure 1: Transformer-based melody-encoder-syllable-decoder architecture exploiting character-level language model.

by using the knowledge of a pre-trained character-level language model to re-evaluate the token probabilities during beam search generation. It could improve the results and generated lyrics quality as opposed to using solely the baseline encoder-decoder Transformer model.

The probability that the language model computes would be $P_l(x_{i-1}, x_{i-2}, \ldots, x_0|x_i)$, where x_i is the *ith* syllable of the lyrics. We only start using the language model from the second generation step, ensuring that x_0 is known. The probability modelled by the Transformer model would be $P_m(x_i|x_{i-1}, x_{i-2}, \ldots, x_0, f_n, f_{n-1}, \ldots, f_0)$, where f_i are the melody features at time i.

The total probability for a given token is then:

$$P(x_i) = \lambda_l * P_l(x_i) + \lambda_t * P_t(x_i),$$

where $\lambda_l + \lambda_t = 1$ are weights indicating which model we prioritize.

In our work, we fine-tune the pre-trained Google CANINE (Clark et al., 2022) model using our dataset. We chose CANINE as it is a widely recognized open-source character-level language model. We use the task of Next Sentence Prediction (NSP), i.e., given a syllable s and a lyric l, predicting the probability $P(s|l)$ that s follows l. Note that in the case of character-level language models, both s and l are sequences, hence the NSP approach can work well. Fine-tuning is essential since the word distribution of lyrics differs significantly from that of resources typically used in training the language model, such as books or Wikipedia. For instance, lyrics contain the words *love*, *hate*, and *gotta* more frequently, and have more lenient grammar.

2.2 Dataset for fine-tuning the language model

We have created the dataset for fine-tuning CANINE based on our lyrics dataset (Yu et al., 2021).

As each syllable of lyrics with its preceding sequence in our original dataset can be thought of as a data point, we are able to obtain a fine-tuning dataset of a considerable size of over 2 million examples.

An example of constructing data samples can be seen in Table 1. For negative examples (label 0), we select a random syllable from the same lyrics sequence that is not the correct continuation of the input sequence. We believe that using syllables from the same lyrics sequence poses a bigger challenge to the language model compared with selecting from the whole vocabulary since the syllables in the same sequence are more plausible candidates than unrelated ones from the vocabulary.

Since the syllables are separated by blank spaces in the melody-lyrics dataset, the lyrics it generates are different from the correctly formatted language that CANINE is used to. Therefore, to enable the pre-trained CANINE model to learn the blank space distribution, we introduce negative data samples with incorrect spacing, i.e., some without the space like "*example*" and some with it like "*_example*". The more probable variant is selected and used to form the context for the next generation step. This allows us to use the language model for connecting the syllables generated by the Transformer into full words. Specifically, for the first three predictions of the NSP task, we introduce negative examples with incorrect spacing, and in the following predictions, we set an incorrect spacing probability of 60%, to avoid significantly increasing the size of the dataset. Negative examples with random syllables selected as the candidate have the spacing information preserved from the original location of the candidate. For instance, in the example *"i know why your mean to me when i call on the", "_the"*, the candidate syllable *the* has a space in front of it, since this is how it originally appeared in the lyric.

Moreover, in order to improve the robustness of the model and its ability to recover from mistakes, in 40% cases we also include examples where one syllable from the preceding lyrics is randomly switched to a different syllable from the same lyric. For instance, in *"i know whytel mean to me when i", "_call"*, the syllable *tel* has randomly replaced the syllable *_your*, making it a negative data sample. Since we are aiming to simulate mistakes, we randomly insert a space before the syllable with a probability 50%.

The dataset used for training the model is imbalanced, with a higher proportion of negative examples compared to positive examples. The reason for such construction is that it reflects the real-world scenario, where the model performs a beam search with multiple candidates, out of which only one is expected to be correct. The model is able to perform well despite the imbalances, achieving convergence after 5 epochs of training.

2.3 Beam search and re-ranking

At each beam search step excluding the first, we have $n = \text{beam size}$ candidate syllables for each of the n beam sequences with the highest probabilities: $S = s_1, ..., s_n$, in total $n \times n$ candidate sequences to consider. The generated candidate syllables are then

$$G = g_{1,1}, g_{1,2}, ..., g_{1,n}, g_{2,1}, ..., g_{n,n}.$$

At the first beam search step, we start with a single <*BOS*> (beginning of sentence) special token, and generate the n best candidates for it, which become $s^0 = s_1, ..., s_n$.

Each generated candidate is associated with the probability assigned by the main transformer model $\mathbf{M} \in \mathbb{R}^{n \times n}$. We also compute the fine-tuned language model probabilities for the sequences

$$L_{i,j} = lm(s_i, g_{i,j}).$$

The final combined probabilities are then

$$C^t_{i,j} = \lambda_m * M_{i,j} + \lambda_l * L_{i,j},$$

for $0 < i, j \leq n$ at each timestep $t \in T$, where $\lambda_m + \lambda_l = 1$ are weights assigned to the predictions of each model. We then select the n best sequences, and continue the process using them as the new $s^{t+1} = s_1, ..., s_n$.

However, this does not take into account the probabilities at previous timesteps. If we consider text generation, the sequence "*I am co ming home*" might receive a low score, since *home* is just one of the possible continuations where one can be *coming*. However, the sequence "*the the could ath lete*", despite making less sense, could score higher, this is because having predicted syllable "*ath*", the model would be highly confident that the next syllable is "*lete*".

To prevent that, a standard technique is to compare the candidates using cumulative probabilities, given by

$$C^t_{i,j} = \lambda_m * M^t_{i,j} + \lambda_l * L^t_{i,j} + C^{t-1}_{i,j},$$

where $C^0_{i,j} = M^0_{i,j}$, since we do not engage the language model in the first beam search step.

lyrics input	candidate syllable	label
i know why your mean to me when	_i	1
i know why your mean to me when	_the	0
i e why your mean to me when	_i	0
i know why your mean to me when	i	0
i know why your mean to me when i	_call	1
i know why your mean to me when i	_on	0
i know whytel mean to me when i	_call	0
i know why your mean to me when i	call	0
...		
i know why your mean to me when i call on the	_tel	1
i know why your mean to me when i call on the	_the	0
i know why your mean to<eos>when i call on the	_tel	0
i know why your mean to me when i call on the	tel	0
i know why your mean to me when i call on the tel	e	1
i know why your mean to me when i call on the tel	_when	0
i know why your mean to me when i call one tel	e	0
i know why your mean to me when i call on the tel	_e	0
i know why your mean to me when i call on the tele	phone	1
i know why your mean to me when i call on the tele	<eos>	0
i know why your mean to me when i call on the tele	_phone	0
i know why your mean to me when i call on the telephone	<eos>	1
i know why your mean to me when i call on the telephone	_phone	0

Table 1: An example of how the fine-tuning dataset is built from sequences of lyrics. The reasons for negative labels are marked in red, while correct spaces are highlighted in green.

3 Experiments

3.1 Experiment setup

We trained a melody-to-lyrics Transformer model as a strong baseline and the basis of our methods. To leverage the ability of the language model, we set the weight of the fine-tuned language model to 75%, leaving 25% for the Transformer. Although the use of the language model noticeably slows down the beam search procedure, a complete evaluation on a validation set containing approximately 1000 examples can still be done in less than 3 hours on an A100 GPU. The fine-tuning of the language model was performed using default hyperparameters from the huggingface library (Wolf et al., 2019), and lasts less than one day on an A100 GPU, despite the size of the fine-tuning dataset.

3.2 Objective metrics

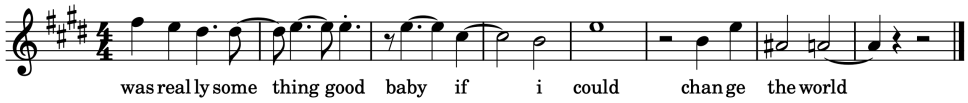
Evaluating creative text objectively is an exceedingly challenging task. Sequence evaluation metrics such as ROUGE and BLEU have limited utility when evaluating creative text because they mainly focus on measuring n-gram similarities between generated sequences and reference sequences. When evaluating creative text, it is crucial to understand that the goal is not to replicate a single ground truth reference. In some cases, an outstanding lyric may be unfairly penalized simply because it deviates from the ground truth, despite effectively fitting the melody and showcasing artistic excellence.

To the best of our knowledge, there exists no objective metrics that can comprehensively capture the quality of the generated lyrics. Therefore, we only use the objective metrics as a means to validate the reconstruction ability of the proposed model.

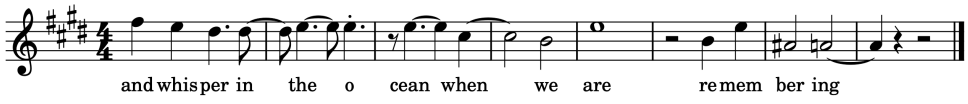
Table 2 shows the evaluation results of our model (Transformer + LM) and the baselines. We selected the recently published semantic dependency network (SDN) as a strong baseline, which already surpassed some methods like LSTM-GAN, SeqGAN, and RelGAN (Duan et al., 2023a). We also implemented the original Transformer as another baseline. The BLEU and ROUGE metrics are slightly worse for the proposed model, however, the difference is insignificant enough to judge that our approach stays relatively close to ground truth in terms of the modeled syllable distribution. In the subjective evaluation in the following sections, and in the generated lyrics from Appendix A, we show that objective metrics can be misleading when evaluating models on a creative task. Examples of generated lyrics accompanied by the input melody are shown in Figure 2, which show that the lyrics generated by our model can better capture the characteristics of musical lyrics. More generated lyrics by using the proposed methods compared with the baseline model can be seen in Appendix A.

Metric	SDN(Duan et al., 2023a)	Transformer	Transformer + LM
ROUGE F score (1,2,L)	0.1301, 0.0008, 0.0981	0.1476, 0.0354, 0.1248	0.1439, 0.0289, 0.1186
Sentence BLEU (2,3,4-gram)	0.0171, 0.0074, 0.0049,	0.0637, 0.0454, 0.0374	0.0576, 0.0386, 0.0308
BERT Scores (Precision, Recall, F1)	0.8771, 0.8870, 0.8819	0.967, 0.968, 0.967	0.967, 0.969, 0.968

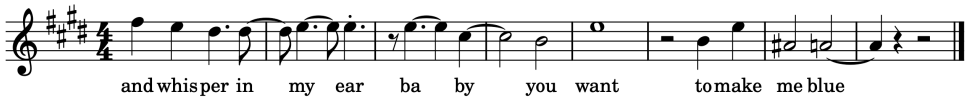
Table 2: Objective metrics on the validation dataset



(a) Ground-truth lyrics.



(b) Generated lyrics by Transformer.



(c) Generated lyrics by Transformer + LM.

Figure 2: Generated sheet music.

3.3 ChatGPT evaluation

Due to the above-mentioned limitations of objective metrics, we proposed to evaluate the quality and correctness of generated lyrics via Large Language Models (LLMs), since they are objective and have a vast linguistic knowledge. It should be noted that our method only evaluates the texts of lyrics, without considering how well they fit the given melodies. Although feeding symbolic melodies could potentially strain the capabilities of LLMs, it is an approach worth exploring in future work.

We asked the GPT-3 (Brown et al., 2020) to evaluate our generated lyrics. After experimenting with the prompts, we proposed the following prompts to let ChatGPT do the evaluation tasks.

> *I will send you three sets of generated candidate lyrics for 20-note melodies. I want you to evaluate them in terms of naturality, correctness, coherence (staying on topic), originality, and poetic value. Try to give numerical scores to all three candidate methods of lyric generation. I will send them in separate messages, please evaluate them after the third message. Is it clear?*

By clarifying the task by the prompts, we hope to exploit the well-known strong language ability of ChatGPT. The conversation is available online[1].

In addition to the aforementioned evaluation session, we informed ChatGPT that the lyrics are syllable-split, lowercase, and without punctuation. This additional information made ChatGPT more aware of the characteristics of our input beyond natural language. The conversation of the second version evaluation can be seen at [2].

We show the results from both runs in Table 3. In both cases, the proposed method is able to outperform the baseline, and in the second evaluation, it also outperforms the ground truth data. During the first evaluation, the ground truth has the highest values in all the categories, while the proposed method is equal to the baseline in two, and outperforms the baseline in 3 of the categories as indicated in bold.

[1] https://chat.openai.com/share/46166c1e-5505-4f74-af3d-3627c905b66c

[2] https://chat.openai.com/share/bcfdcac3-b63c-44e2-bb29-c93699eae8f2

Metrics	Ground-truth		Transformer		Trans.+LM	
	1st	2nd	1st	2nd	1st	2nd
Naturality	6	6	3	5	**4**	**7**
Correctness	7	7	4	6	**5**	**8**
Coherence	5	5	3	4	3	**6**
Originality	4	4	2	3	**3**	**5**
Poetic Value	4	5	2	4	2	**6**
Overall	5.2	5.4	2.8	4.4	**3.4**	**6.4**

Table 3: Results of the ChatGPT evaluation of generated lyrics on a scale from 1 to 10.

During the second evaluation, the proposed method has the highest values in all of the categories. We argue that by clarifying the characteristics of our text input, ChatGPT focuses more on the correctness and quality of the syllable-level split lyrics, hence giving higher scores on our model. This also verified the effectiveness of our proposed methods with language models.

3.4 Subjective evaluation

Subjective evaluation is an important metric for evaluating creative text generation systems, especially for evaluating the fitness between the generated lyrics and input melodies.

3.4.1 Evaluation of generated lyrics

We conduct a subjective experiment with the same questions in subsection 3.3 on 11 participants with different levels of musical knowledge to compare human and ChatGPT-based evaluation of texts of generated lyrics. The evaluation results of human participants are visualized via boxplots in Figure 3, where we also annotated the ChatGPT-based evaluation results in subsection 3.3 for comparison. We found that human evaluation and ChatGPT-based evaluation show the same general trends among the three methods despite the difference in the numerical scales, where the ground-truth lyrics are rated highest and our model surpasses the Transformer baseline. Moreover, by comparing two sets of ChatGPT-based evaluation results in subsection 3.3, we found that a more detailed description for ChatGPT about the lyrics to be evaluated is helpful to get the results that are more similar with human evaluation results. However, due to the limited number of participants in our evaluation, it is difficult to perform a thorough correlation analysis. We leave it as future work to conduct a comprehensive analysis with a large number of participants to study the correlation between human and ChatGPT evaluation.

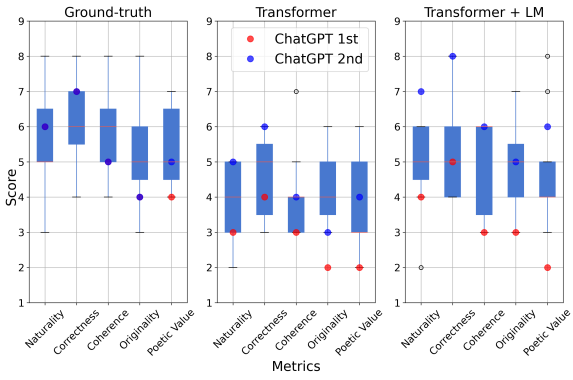


Figure 3: Correlation between ChatGPT-based evaluation and human evaluation of generated lyrics.

3.4.2 Evaluation of synthesized music with lyrics and melody

In addition to the above text-based evaluation of generated lyrics, we performed a subjective evaluation by synthesizing audible samples of our generated lyrics with input melodies and distributing a questionnaire including the audio samples to 11 participants with different levels of musical knowledge. The questionnaire and samples are available at Google Form[3]. We have tried to exclude highly famous songs in the form, to prevent participants from identifying the ground truth hidden reference. The questions used in the subjective evaluation are listed as follows.

1. Assess the correctness and coherence of the provided lyrics as natural language, without considering the melody.
2. What do you think about the creativity and poetic value of the text as song lyrics?
3. How well do the generated lyrics fit the input melody in terms of rhythm?
4. How well do the generated lyrics fit the input melody in terms of atmosphere?

The rating scores are on a 5-point scale (very bad, bad, okay, good, very good). After the subjects finished their questionnaire, we collected the results and calculated the average scores rated for each model. The human evaluation results are shown in Figure 4.

Evaluation results show that our proposed model achieves an improvement based on the Transformer

[3]https://forms.gle/RN88Exw3D7H8DjvN7

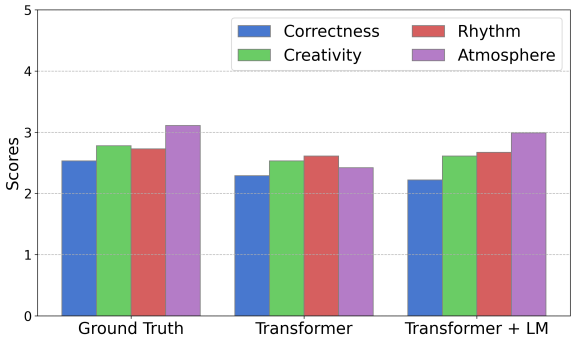


Figure 4: Results of subjective evaluation of lyrics generation from melody.

baseline. Also, it is worth mentioning that the potential consistency between human evaluation and ChatGPT evaluation observed in the experiments of 3.4.1 makes it promising for future research on ChatGPT-based evaluation, which could be an effective way to improve evaluation efficiency and reduce human resource costs, leveraging the linguistic power of the pre-trained LLMs.

4 Background and related works

Lyrics generation has been an active area of research, with various methodologies being proposed over the years. Early efforts in lyrics generation predominantly utilized traditional machine learning methods. For instance, Ramakrishnan A et al. (2009) focused on the automatic generation of Tamil lyrics for melodies by predicting the syllable patterns from melodies and subsequently filling the pattern using a corpus.

With the advent of deep learning, there has been a surge in models tailored for automatic lyrics generation. Generating lyrics conditioned on symbolic melody can be thought of as the intersection of creative text generation, and computer music modeling. In both of these areas, recent years have been dominated by deep learning (Brown et al., 2020; Agostinelli et al., 2023), leading us to primarily research deep neural networks. Fan et al. (2019) proposed a hierarchical attention-based Seq2Seq model for Chinese lyrics generation that emphasized both word-level and sentence-level contextual information. Lu et al. (2019) employed RNN encoders for encoding syllable structures and semantic encoding with contextual sentences or input keywords. Wu et al. (2019) introduced a Chinese lyric generation system using an LSTM network to capture the patterns and styles of lyricists. Wang and Zhao (2019) presented a theme-aware language generation model to enhance the theme-connectivity and coherence of generated paragraphs. Furthermore, Nikolov et al. (2020) developed Rapformer, a method that utilizes a Transformer-based denoising autoencoder to reconstruct rap lyrics from extracted content words.

A subset of research has delved deeper into the relationship between lyrics and melodies. Watanabe et al. (2018) proposed a data-driven language model that crafts lyrics for a given input melody. Vechtomova et al. (2020) utilized a bimodal neural network to generate lyrics lines based on short audio clips. Chen and Lerch (2020) employed SeqGAN models for syllable-level lyrics generation conditioned on lyrics. Sheng et al. (2020) leveraged unsupervised learning to discern the relationship between lyrics and melodies. Chang et al. (2021) introduced a singability-enhanced lyric generator with music style transfer capabilities. Huang and You (2021) proposed an emotion-based lyrics generation system combining a support vector regression model with a sequence-to-sequence model. Ma et al. (2021) presented AI-Lyricist, a system designed to generate vocabulary-constrained lyrics given a MIDI file. Zhang et al. (2022a) and Liu et al. (2022) explored methods to enhance the harmony between lyrics and melodies, with the latter focusing on system controllability and interactivity. Lastly, large-scale pre-trained models have also been explored by (Rodrigues et al., 2022) and Zhang et al. (Zhang et al., 2022b).

Many above existing works of lyrics generation are based on word-level sequence generation. In (Yu et al., 2021), a syllable-level lyrics-melody paired dataset was proposed with an LSTM-GAN model addressing the lyrics-conditioned melody generation problem. Some following works also explored lyrics-to-melody generation problems based on this dataset (Yu et al., 2020; Srivastava et al., 2022; Duan et al., 2022, 2023b; Yu et al., 2023; Zhang et al., 2023). However, melody-to-lyrics generation on syllable level is a more difficult task in predicting semantic dependencies among syllable-level, word-level, and sentence-level meaning. A semantic dependency network is proposed in (Duan et al., 2023a) to address the degraded text quality in the syllable-level lyrics generation task. In our work, fine-tuning a pre-trained character-level language model is proposed to help the syllable-level melody-to-lyrics Transformer to generate lyrics with better grammar correctness and semantic meaning.

5 Conclusion

In this work, we proposed a method to enhance the predictions of a syllable-level melody-conditioned lyrics generation Transformer, which utilizes pre-trained character-level language models fine-tuned on lyrics data. We propose a method for creating a dataset tailored to fine-tune the character-level language model for refining syllable-level semantic meanings. Moreover, we present an algorithm for re-ranking candidate tokens during the beam search procedure.

We prove that our syllable-level refinement leads to improved naturality, correctness, and coherence of lyrics, while maintaining them tightly related to the conditioning melodies via the use of the encoder-decoder architecture. In future work, we plan to work on pre-training a syllable-level language model on a large data corpus, and then fine-tuning it, as well as exploring fine-tuning character-level language models for the task of lyrics-conditioned melody generation.

6 Limitations

There are several limitations in the current work and directions for future research:

1. Incorporating melody information for ChatGPT evaluation: While our current ChatGPT-based evaluation focuses on the linguistic quality of the generated lyrics, future work could explore ways to provide melody context to ChatGPT, allowing it to evaluate the fit between lyrics and melody.

2. Expanding the dataset: Our current dataset, though substantial, is limited in its diversity. Gathering more diverse melody-lyrics pairs can further enhance the generalization capabilities of the model.

3. Exploring other pre-trained models: While we used the CANINE model in our experiments, other character-level or subword-level models could be explored to see if they offer any advantages in this task.

4. End-to-end training: Instead of a two-step process (Transformer generation followed by language model re-ranking), an end-to-end training approach where both models are jointly trained could be explored.

5. Risks: It is possible that our method can be utilized to predict lyrics when given melodies. Therefore, it could potentially be leveraged for fake music generation. We will restrict the usage of our method and share our model with the AI community to contribute to the reliability of AI music generation.

References

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. Musiclm: Generating music from text.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Dirk Bühler, Wolfgang Minker, and Artha Elciyanti. 2005. Using language modelling to integrate speech recognition with a flat semantic analysis. In *SIGDIAL Conferences*.

Jia-Wei Chang, Jason C. Hung, and Kuan-Cheng Lin. 2021. Singability-enhanced lyric generator with music style transfer. *Computer Communications*, 168:33–53.

Yihao Chen and Alexander Lerch. 2020. Melody-Conditioned Lyrics Generation with SeqGANs. In *2020 IEEE International Symposium on Multimedia (ISM)*, pages 189–196.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Wei Duan, Yi Yu, and Keizo Oyama. 2023a. Semantic dependency network for lyrics generation from melody. *Neural Computing and Applications*.

Wei Duan, Yi Yu, Xulong Zhang, Suhua Tang, Wei Li, and Keizo Oyama. 2023b. Melody Generation from Lyrics with Local Interpretability. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 19(3):124:1–124:21.

Wei Duan, Zhe Zhang, Yi Yu, and Keizo Oyama. 2022. Interpretable Melody Generation from Lyrics with

Discrete-Valued Adversarial Training. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6973–6975.

Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A Hierarchical Attention Based Seq2Seq Model for Chinese Lyrics Generation. In *PRICAI 2019: Trends in Artificial Intelligence*, pages 279–288.

Yin-Fu Huang and Kai-Cheng You. 2021. Automated Generation of Chinese Lyrics Based on Melody Emotions. *IEEE Access*, 9:98060–98071.

Nayu Liu, Wenjing Han, Guangcan Liu, Da Peng, Ran Zhang, Xiaorui Wang, and Huabin Ruan. 2022. ChipSong: A Controllable Lyric Generation System for Chinese Popular Song. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 85–95.

Xu Lu, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A Syllable-Structured, Contextually-Based Conditionally Generation of Chinese Lyrics. In *PRICAI 2019: Trends in Artificial Intelligence*, pages 257–265.

Xichu Ma, Ye Wang, Min-Yen Kan, and Wee Sun Lee. 2021. AI-Lyricist: Generating Music and Vocabulary Constrained Lyrics. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1002–1011.

Nikola I. Nikolov, Eric Malmi, Curtis G. Northcutt, and Loreto Parisi. 2020. Rapformer: Conditional Rap Lyrics Generation with Denoising Autoencoders.

Ananth Ramakrishnan A, Sankar Kuppan, and Sobha Lalitha Devi. 2009. Automatic Generation of Tamil Lyrics for Melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 40–46.

Matheus Augusto Rodrigues, Alcione Oliveira, Alexandra Moreira, and Maurilio Possi. 2022. Lyrics Generation supported by Pre-trained Models. *The International FLAIRS Conference Proceedings*, 35.

Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. 2020. SongMASS: Automatic Song Writing with Pre-training and Alignment Constraint.

Abhishek Srivastava, Wei Duan, Rajiv Ratn Shah, Jianming Wu, Suhua Tang, Wei Li, and Yi Yu. 2022. Melody Generation from Lyrics Using Three Branch Conditional LSTM-GAN. In *MultiMedia Modeling*, pages 569–581.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Olga Vechtomova, Gaurav Sahu, and Dhruv Kumar. 2020. Generation of lyrics lines conditioned on music audio clips.

Jie Wang and Xinyan Zhao. 2019. Theme-aware generation model for chinese lyrics.

Shuhe Wang, Yuxian Meng, Xiaofei Sun, Fei Wu, Rongbin Ouyang, Rui Yan, Tianwei Zhang, and Jiwei Li. 2021. Modeling text-visual mutual dependency for multi-modal dialog generation.

Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. 2018. A Melody-Conditioned Lyrics Language Model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 163–172.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Xing Wu, Zhikang Du, Yike Guo, and Hamido Fujita. 2019. Hierarchical attention based long short-term memory for Chinese lyric generation. *Applied Intelligence*, 49(1):44–52.

Yi Yu, Florian Harscoët, Simon Canales, Gurunath Reddy M, Suhua Tang, and Junjun Jiang. 2020. Lyrics-Conditioned Neural Melody Generation. In *MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II*, pages 709–714.

Yi Yu, Abhishek Srivastava, and Simon Canales. 2021. Conditional LSTM-GAN for Melody Generation from Lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 17(1):35:1–35:20.

Yi Yu, Zhe Zhang, Wei Duan, Abhishek Srivastava, Rajiv Shah, and Yi Ren. 2023. Conditional hybrid GAN for melody generation from lyrics. *Neural Computing and Applications*, 35(4):3191–3202.

Chen Zhang, Luchin Chang, Songruoyao Wu, Xu Tan, Tao Qin, Tie-Yan Liu, and Kejun Zhang. 2022a. ReLyMe: Improving Lyric-to-Melody Generation by Incorporating Lyric-Melody Relationships. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1047–1056.

Rongsheng Zhang, Xiaoxi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2022b. Youling: An AI-Assisted Lyrics Creation System.

Zhe Zhang, Yi Yu, and Atsuhiro Takasu. 2023. Controllable lyrics-to-melody generation. *Neural Computing and Applications*, 35(27):19805–19819.

A Generated lyrics

Ground truth	Transformer	Transformer + LM
how minus cule is any light if it light you breaking up the fold for your love	when it takes more than i met you the sub way the pow er of the stars	not the way you bet ter than you ev er seen it when you need some thing
i need to know the way to feel to keep me sat is fied	i know i be lieve i can give you the way it got to	i know i believe in love with you to mor row bless me soon
in their mas que rade no the out to get you	you got ta o in what the you used to life	you got ta treat me to may be un der stand you
and i touched her on the sle e ve she rec og nize the face at first	and i be lieve i can fol low you know i must have known it ea sy	but i be lieve i can fol low you know i have to face it of us
da la da da la da da drift a way fade a way lit tle tin god dess	da da da la da da da da da la da da da da da da da	we can give this world to ge ther and we are not so da da da da da
from mem phis ten nes see her home is on the south side high up on a ridge	for get a no ther way you real ly need to know now when it feels like you	for get a no ther way you real ly need to know now when it feels so hard
went crash boom bang the whole rhy thm sec tion was the pur ple gang rock	must have been ran ing to the an swer to we got no thing no thing	must have been talk ing to the an swer i wan na live for some thing
you take mur der on the	in the wings of the ri ver	in the wings of the ri ver
with you and the lit tle days and party joints do now just miss ing you how i wish	a gain why i come a gain why i must be my su per to me smil ing like	a gain why i must re mem ber the sun shine fills my head with me and she stings
i want to break free i want i want i want i want to break free to break free	i ne ver on ly know i on ly know who i am i was born on a wall	i should be here i am i ne ver seen your horse and i know what i feel inside
and it it makes me me sad for the ly walked that road for so now i know that the	i stand the ground and i stand the fire my friend and i need a rai ny roads i need	i stand the ground and i stand the fire my friend some times i need a rai ny roads run
do ing do wop do we were in the with our blue suede shoes	an y li ons they say that you were a life of your life	they know what they think that they were six teen your world a bout
i got my first real bought it at the played it till my fin gers bled	and she looks so hard to un der stand that she comes the game and they	and she looks so hard to un der stand the word and they come to town
to it mad bur ning mad it it mad ni ght the beat to the beat to the beat	to you know gon na be a and i your to be doing the the the be oh the	to night gon na be out of the night babe cos i will be a called love grow when
get down and move it a round hey love need girl you tell if feel too in hour	what i hea ven no bod y no bod y wants what i heard you a ny thing	your bod y call me your bod y sis ter su per star hol low and too much
she rush es out to hold him thank ful a live but on the wind and rain	and by the way you come a lit tle bit more you get a lit tle clos	you can say a bout my love for you to day and you get a feel ing
must be how could so much love be in side of you whoa oh	on the run ning on the run ning to be with you to town	on the road got to be shin ing on the streets of the town
high out side your door late at night when not sleep ing and moon light falls a cross your floor	why do i have to die why we won der where it was the rain bow is fall ing down	why do i have to die why we won der where it was the rain bow is fall ing down
ma ha mm ma ha ha ha ha ha ha the world	she said got me love for me but each oth er day	she said got me love for me but each oth er day
love has tak en life time child girl you know you are the nic est thing love your rap	sex bomb and can you feel the one smile you know you smile you smile i want to cry	sex bomb and smile you take the mo ney sex bomb and smile you know talk to get back
the glo ries of his righ teous ness and won ders of his love and won ders of his love	un der stand why mark and if on ly i say this is ach ing you if i do this	un til this day i wear my heart and try to bring me out of mind if i should let

Table 4: Comparison of generated lyrics.