Youling: an AI-Assisted Lyrics Creation System

Rongsheng Zhang[1]*, **Xiaoxi Mao**[1]†*, **Le Li**[1], **Lin Jiang**[1],
Lin Chen[1], **Zhiwei Hu**[1], **Yadong Xi**[1], **Changjie Fan**[1], **Minlie Huang**[2]

[1] Fuxi AI Lab, NetEase Inc., Hangzhou, China
[2] Department of Computer Science and Technology, Institute for Artifical Intelligence, State Key Lab of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China.
{zhangrongsheng, maoxiaoxi, lile, jianglin02}@corp.netease.com,
aihuang@tsinghua.edu.cn

Abstract

Recently, a variety of neural models have been proposed for lyrics generation. However, most previous work completes the generation process in a single pass with little human intervention. We believe that lyrics creation is a creative process with human intelligence centered. AI should play a role as an assistant in the lyrics creation process, where human interactions are crucial for high-quality creation. This paper demonstrates *Youling*, an AI-assisted lyrics creation system, designed to collaborate with music creators. In the lyrics generation process, *Youling* supports traditional one pass full-text generation mode as well as an interactive generation mode, which allows users to select the satisfactory sentences from generated candidates conditioned on preceding context. The system also provides a revision module which enables users to revise undesired sentences or words of lyrics repeatedly. Besides, *Youling* allows users to use multifaceted attributes to control the content and format of generated lyrics. The demo video of the system is available at https://youtu.be/DFeNpHk0pm4.

1 Introduction

Lyrics Generation has been a prevalent task in Natural Language Generation (NLG), due to the easy availability of training data and the value of the application. However, despite the popularity of lyrics generation, there still lacks a comprehensive lyrics creation assistant system for music creators. Previous researches (Castro and Attarian, 2018; Saeed et al., 2019; Lu et al., 2019; Manjavacas et al., 2019; Watanabe et al., 2018; Potash et al., 2018; Fan et al., 2019; Li et al., 2020) and systems (Potash et al., 2015; Lee et al., 2019; Shen et al., 2019), are mostly model-oriented, utilizing neural networks including GAN, RNN-based or Transformer-based (Vaswani et al., 2017) sequence to sequence (Seq2Seq) models for sentence-wise lyrics generation. They complete the lyrics generation process in a single pass with specific keywords or content controlling attributes as input, involving little human intervention. However, we believe the lyrics creation process should be human intelligence centered, and AI systems shall serve as assistants, providing inspiration and embellishing the wording of lyrics.

Therefore, we demonstrate *Youling*, an AI-assisted lyrics creation system, which is designed to collaborate with music creators, help them efficiently create and polish draft lyrics. To fulfill the goal, *Youling* supports interactive lyrics generation, in addition to the traditional one pass full-text generation. Interactive lyrics generation allows users to carefully choose desirable sentences from generated candidates conditioned on preceding context line by line. Preceding context can be either pre-generated, written by users, or a mix. *Youling* also has a revision module, which supports users to revise any unsatisfied sentences or words of draft lyrics repeatedly.

To ensure the controllability of generated lyrics, *Youling* supports multifaceted controlling attributes to guide the model to generate lyrics. These controlling attributes can be divided into two categories, content controlling attributes and format controlling attributes. Content controlling attributes include the lyrics' text style, the emotion or sentiment expressed in the lyrics, the theme described in the lyrics, and the keywords expected to appear in the lyrics. Format controlling attributes include the acrostic characters(letters), the rhymes of the lyrics, the number of sentences, and the number of words per sentence.

To ensure the quality and relevance of generated lyrics with controlling attributes, we implement

[*] Equal contribution
[†] Corresponding Author

Proceedings of the 2020 EMNLP (Systems Demonstrations), pages 85–91
November 16-20, 2020.

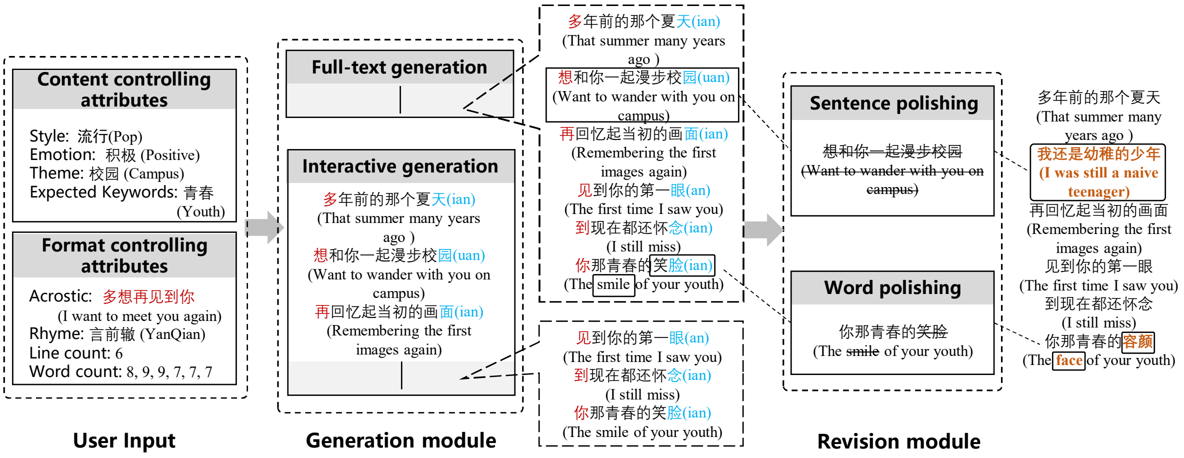

Content controlling attributes
Style: 流行(Pop)
Emotion: 积极 (Positive)
Theme: 校园 (Campus)
Expected Keywords: 青春 (Youth)
Format controlling attributes
Acrostic: 多想再见到你 (I want to meet you again)
Rhyme: 言前辙 (YanQian)
Line count: 6
Word count: 8, 9, 9, 7, 7, 7
User Input
Full-text generation
Interactive generation
多年前的那个夏天(ian) (That summer many years ago)
想和你一起漫步校园(uan) (Want to wander with you on campus)
再回忆起当初的画面(ian) (Remembering the first images again)
Generation module
多年前的那个夏天(ian) (That summer many years ago)
想和你一起漫步校园(uan) (Want to wander with you on campus)
再回忆起当初的画面(ian) (Remembering the first images again)
见到你的第一眼(an) (The first time I saw you)
到现在都还怀念(ian) (I still miss)
你那青春的笑脸(ian) (The smile of your youth)
见到你的第一眼(an) (The first time I saw you)
到现在都还怀念(ian) (I still miss)
你那青春的笑脸(ian) (The smile of your youth)
Sentence polishing
~~想和你一起漫步校园~~ ~~(Want to wander with you on campus)~~
Word polishing
你那青春的笑脸 (The ~~smile~~ of your youth)
Revision module
多年前的那个夏天 (That summer many years ago)
我还是幼稚的少年 (I was still a naive teenager)
再回忆起当初的画面 (Remembering the first images again)
见到你的第一眼 (The first time I saw you)
到现在都还怀念 (I still miss)
你那青春的容颜 (The face of your youth)

Figure 1: Architecture of *Youling*. The system supports multifaceted controlling attributes in user input to control the content and format of lyrics. The generation module provides two modes for draft lyrics creation: full-text generation and interactive generation. The former generates a full lyrics while the latter generates following sentences conditioned on the preceding context. Besides, a revision module is introduced to polish undesirable sentences or words.

Youling basing on a GPT-2 (Radford et al., 2019) based language model with 210M parameters, pretrained on around 30 gigabytes of Chinese books corpus. We further finetune *Youling* on a corpus of 300K lyrics collected online.

The contributions of the *Youling* system are summarized as follows:

1. *Youling* provides multiple modes to assist users in lyrics creation. It supports both the traditional one pass full-text generation and the interactive lyrics generation. It also provides a revision module for users to revise undesirable sentences or words of draft lyrics repeatedly.

2. To the best of our knowledge, *Youling* supports the largest variety of content controlling attributes and format controlling attributes to date.

3. *Youling* is implemented on top of GPT-2 model to ensure the quality and relevance of generated lyrics with controlling attributes.

We believe that *Youling* [1] can assist music creators in lyrics creation and inspire other developers to make practical solutions for real-world problems. The 2-minute demonstration video can be available at https://youtu.be/DFeNpHk0pm4.

2 Architecture

The framework of *Youling* is shown in Figure 1. The system mainly contains three parts: **user input**, **generation module** and **revision module**. We will describe them in detail in the following subsections.

2.1 User Input

The input includes a wide variety of controlling attributes provided by users. They can be divided into two categories: content controlling attributes and format controlling attributes. Content controlling attributes consist of the lyrics' text style, the emotion expressed in the lyrics, the theme described in the lyrics, and the keywords expected to appear in the lyrics. Our system supports four kinds of text styles, including Pop, Hip-hop, Chinese Neo-traditional and Folk; three kinds of emotion (positive, negative, and neutral); 14 kinds of themes such as college life, unrequited love, reminiscence, friendship, and so on. Format controlling attributes consist of the acrostic characters (letters), the rhymes of the lyrics, the number of lines of lyrics, and the number of words per line. Users can choose rhyme from 13 Chinese traditional rhyming groups (十三辙).

2.2 Generation Module

Once users have prepared the controlling attributes, the generation module can generate lyrics in full-text generation mode or interactive generation mode. Below we will explain in detail how we

[1] Our system is available at https://yl.fuxi.netease.com/, visitors can log in with the public account (*youlingtest@163.com*) and password (*youling666*).

implement the lyrics generation conditioned on so many controlling attributes.

2.2.1 Full-Text Generation

Model and Pre-training: We use a Transformer-based sequence to sequence model for the generation of lyrics. To ensure the performance, we use a pre-trained language model based on GPT-2 to initialize the weights of the Transformer encoder and decoder. Our encoder uses a unidirectional self-attention similar to GPT-2; in addition, GPT-2 has only one self-attention block per layer, so the two self-attention blocks in each decoder layer share the same weights. For saving memory, the encoder and decoder share the same weights (Zheng et al., 2020). Our pre-trained language model has 16 layers, 1,024 hidden dimensions, 16 self-attention heads, and 210 million parameters. It is pre-trained on around 30 gigabytes of Chinese Internet novels collected online, which is tokenized with Chinese character. The vocabulary size is 11,400 and the context size is 512.

Training: Here we describe how we train the sequence to sequence model. We collected 300K lyrics from the Internet as training data, including 60M tokens in total. To achieve controllable generation, we need to annotate the style and mood tags corresponding to each song's lyrics and extract the keywords in the lyrics. The style tags corresponding to the lyrics were also obtained as we crawled the lyrics, so no additional processing is required. To get emotion labels, we used a ternary emotion classifier to classify emotion for each song's lyrics. The emotion classifier was trained on 20k labeled data and achieved 80% accuracy on the validation set. To get the keywords contained in each song's lyrics, we extracted all the nouns, verbs, and adjectives in the lyrics.

After the pre-processing described above, we have the style tags, emotion tags, and keyword lists corresponding to each song's lyrics and can start building the training data. The encoder input is a concatenation of the style tag, emotion tag and keywords corresponding to the song lyrics with the *[SEP]* special character. Since there are too many keywords extracted from a song's lyrics, we augment training examples by sampling different numbers of keywords multiple times. This approach is to allow the model to better generalize to the number of keywords. To construct the decoder output, we use a special token *[SEP]* to concatenate every line in a song lyrics, where the last character of each line is placed at the beginning for rhyming control. Finally, we append a special token *[EOS]* to the end of the decoder output. Kindly note that the constraints on format attributes, as well as the theme tag, are imposed during inference, so they will not be included in the training phase.

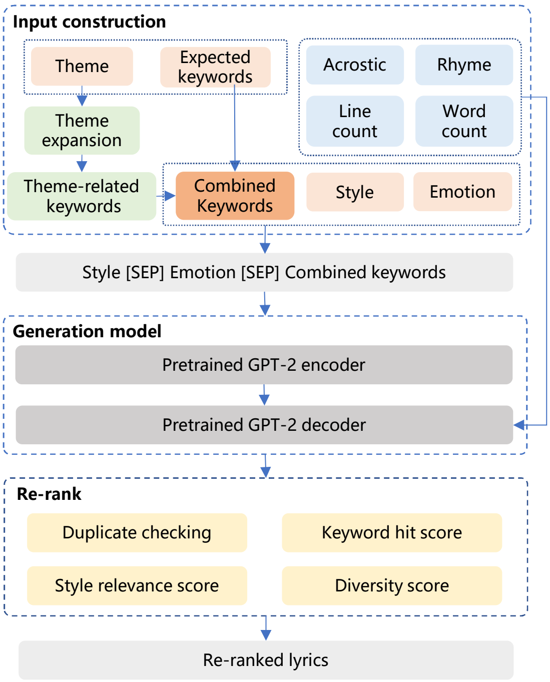


Figure 2: The inference process of full-text generation.

Inference: Here we introduce the inference process, as shown in Figure 2. Under full-text generation mode, the source sequence is a concatenation of the user-entered style tag, emotion tag, and keywords. The keywords include the expected keywords, as well as keywords related to the theme selected by the user. The keywords related to different themes are obtained through offline computation. We calculated PMI (Pointwise Mutual Information) for all word pairs in the lyrics corpus after removing low-frequency words. The PMI of word pair w_i, w_j is calculated as

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i) * p(w_j)}, \quad (1)$$

where $p(w_i)$ and $p(w_i, w_j)$ are the word frequency and co-occurrence frequency. We keep all word pairs with PMI above a specific threshold, which gives us the lists of keywords corresponding to specific themes. At inference time, we randomly sample the keywords list corresponding to the theme selected by the user to get the input keywords, which are then concatenated with user-entered keywords, style tag, and emotion tag to form the final source sequence.

Format control in decoding: We describe the details of format control in decoding. To keep the number of lines and words per line in accordance to the user's requirements, we record the number of lines and words of the generated lyrics at every decoding step and adjust the logits of *[SEP]* and *[EOS]* in the decoder output accordingly. To achieve rhyming control, we always generate the last character of a line first and then generate the rest from left to right. We adjust the training examples accordingly, as mentioned before. To achieve the acrostic control, we simply amplify the corresponding logit in the decoder output to a very large value when generating the acrostic character of each line of lyrics.

Re-rank: We adopt the top-k sampling method at decoding to generate candidate results. Then we re-rank the candidates according to four rules. (1) Duplicate checking: Due to the strong copy ability of Transformer (Lioutas and Drozdyuk, 2019), the generated lyrics may contain original pieces of text in the training corpus, which will introduce copyright issues. To avoid that, we remove any candidate result containing three or more lines overlapping with the training corpus. (2) Keyword hit (kh) score: For each candidate, we compute the keyword hit score as $S_{kh} = n/n_{max}$, where n is the number of keywords appearing in the current candidate, n_{max} is the number of of keywords in the one with the most hits in all candidates. (3) Style relevance (sr) score: This score measures how well each candidate matches its target style $style_t$. To compute the score, we train a style classifier g on the collected lyrics corpus, and take the classification probability of the target style of the generated lyrics as $S_{sm} = g(style_t|lyric)$. (4) Diversity (div) score: As mentioned before, the Transformer model is likely to copy original lyrics in the training data. Besides, repetition is also common in lyrics; thus, the learned model may constantly generate repeated pieces of text. Sometimes repetition is good, but too much repetition needs to be avoided. We count the number of repeated sentences in each candidate and calculate the diversity score as $S_{div} = 1 - n_{rep}/n_{tot}$, where n_{rep} and n_{tot} denotes the number of repeated sentences and all sentences respectively. The final ranking score of each candidate is computed as

$$S_{rank} = \lambda_1 S_{kh} + \lambda_2 S_{sm} + \lambda_3 S_{div}, \qquad (2)$$

where λ_1, λ_2, λ_3 are weights and default to 1.0.

2.2.2 Interactive Generation

For the interactive generation, we use the same model used for full-text generation. The differences exist at decoding. The first difference is that under the interactive generation mode, generation is conditioned on both the encoder input and the preceding context. In other words, the interactive generation can be formulated as

$$s_{i+1}, ..., s_{i+k} = \text{Model}(X, s_0, s_1, ..., s_i), \qquad (3)$$

where the s_i means the *i*-th line of the lyrics text Y, and k is the number of lines to be generated. In comparison the full-text generation is just formulated as $Y = \text{Model}(X)$. The second difference is that the interactive generation mode generates only a few lines $s_{i+1}, .., s_{i+k}$ rather than the full lyrics Y. Hence, under the interactive generation mode, the preceding context must be provided, which can either be pre-generated by *Youling*, written by the user, or a mix of them.

For the example of interactive generation in Figure 1, the system generates the following three lines "*The first time I saw you [SEP] I still miss [SEP] The smile of your youth*" based on the user input and the preceding context "*That summer many years ago [SEP] Want to wander with you on campus [SEP] Remembering the first images again*".

2.3 Revision Module

The revision module provides useful features allowing users to further polish draft lyrics at the sentence or word level. The framework of the revision module is shown in Figure 3.

The model of revision module follows the same sequence to sequence framework used in the full-text generation model, initialized with weights of the same pre-trained language model.

To build training examples for the model, we simply randomly replace a sentence or a word of lyrics in the training corpus with a special token *[MASK]*. The result is concatenated with the corresponding style tag as the final source sequence, with the form "*Style [SEP] Masked Lyrics*." The sentence or word replaced becomes the target sequence. We use an example to illustrate this idea, given the lyrics "*The snow glows white on the mountain tonight [SEP] Not a footprint to be seen [SEP] A kingdom of isolation [SEP] And it looks like I'm the queen ...*", we replace the sentence "*Not*

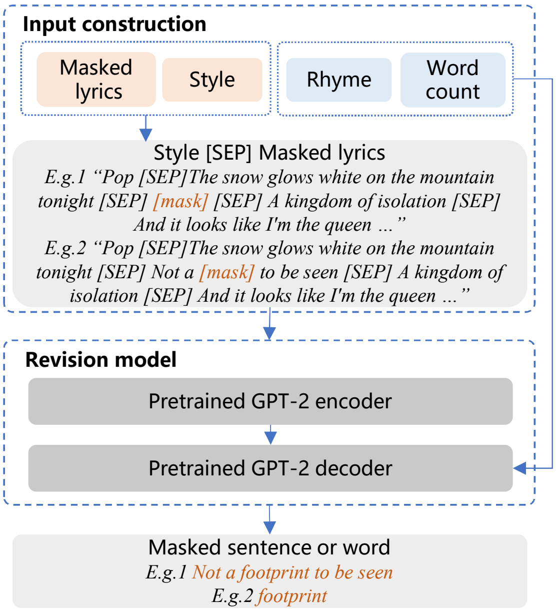

Input construction
Masked lyrics
Style
Rhyme
Word count
Style [SEP] Masked lyrics
E.g.1 “Pop [SEP]The snow glows white on the mountain tonight [SEP] [mask] [SEP] A kingdom of isolation [SEP] And it looks like I'm the queen ...”
E.g.2 “Pop [SEP]The snow glows white on the mountain tonight [SEP] Not a [mask] to be seen [SEP] A kingdom of isolation [SEP] And it looks like I'm the queen ...”
Revision model
Pretrained GPT-2 encoder
Pretrained GPT-2 decoder
Masked sentence or word
E.g.1 Not a footprint to be seen
E.g.2 footprint

Figure 3: The process of the revision module polishing lyrics.

a footprint to be seen” or the word “*footprint*” with the masking token *[MASK]*, and take the masked contents as the target sequence, as shown in Figure 3. Note that we don't treat word-level and sentence-level replacement differently, so the revision is executed with the same model.

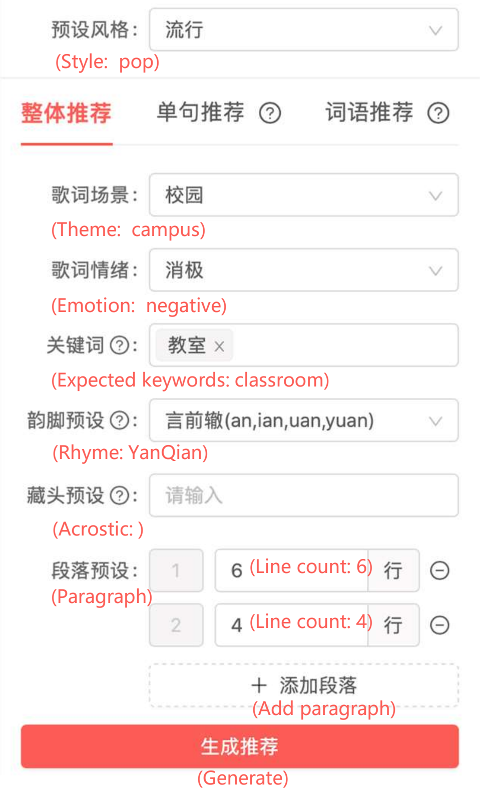
3 Demonstration

In this section, we demonstrate how *Youling* assists music creators to create lyrics conveniently.

First, we show how to generate draft lyrics based on multifaceted controlling attributes. Users are asked to specify the controlling attributes, as shown in Figure 4. After the controlling attributes have been prepared, we use the full-text generation mode to generate the draft lyrics, as shown in Figure 5(a).

After the draft lyrics are generated, we use the interactive generation mode to generate the following lines. Note that in real cases, users can directly write lyrics or modify pre-generated lyrics in the input box and generate the following lines with interactive generation mode. Here we use the unchanged generated draft lyrics for convenience of demonstration.

After completing the draft lyrics by carefully choosing the final line from generated candidates, we can further polish the undesired parts of the generated lyrics. Here we replace a flawed sentence with the best suggestion made by the revision module under sentence level, as seen in Figure 6(a).


预设风格：流行
(Style: pop)

歌词场景：校园
(Theme: campus)

歌词情绪：消极
(Emotion: negative)

关键词 ⓘ：教室 ×
(Expected keywords: classroom)

韵脚预设 ⓘ：言前辙(an,ian,uan,yuan)
(Rhyme: YanQian)

藏头预设 ⓘ：请输入
(Acrostic:)

段落预设：
(Paragraph)
1　6 (Line count: 6) 行
2　4 (Line count: 4) 行

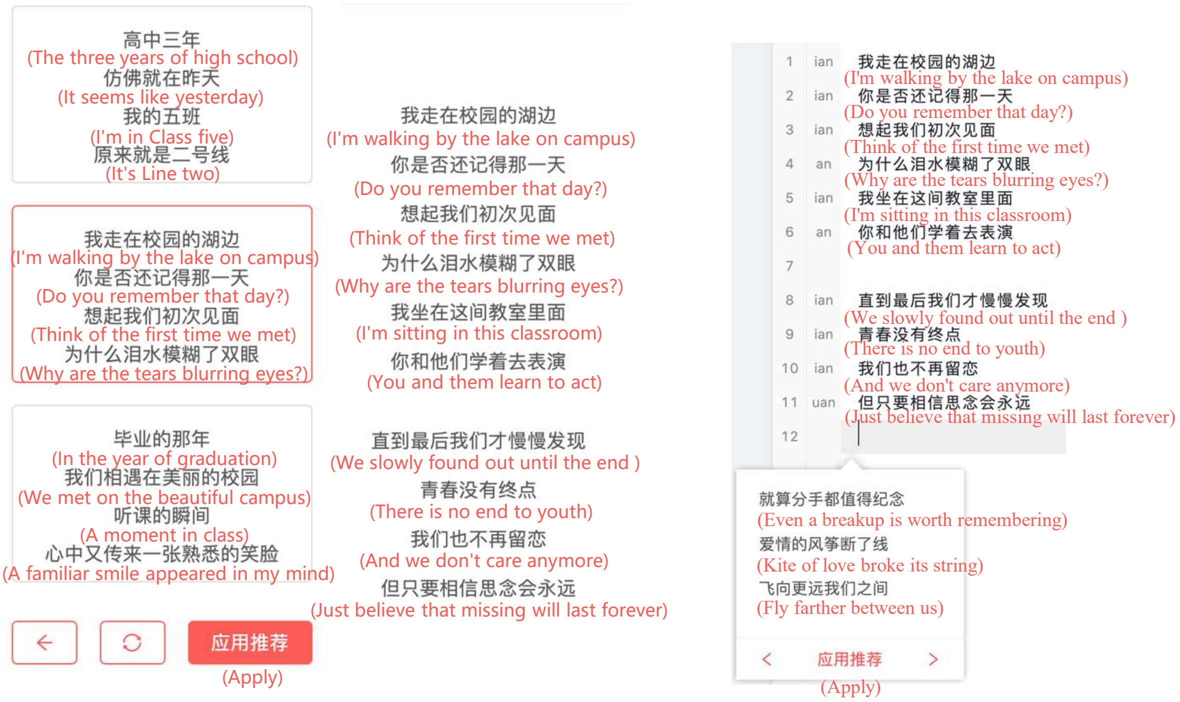
Figure 4: A case of the input page. Users can set content and format controlling attributes.

However, we are still not completely satisfied with the last word in the previous suggested sentence. We switch to word level and replace the last word with an appropriate word suggested by the revision model, as shown in Figure 6(b).

As described above, users can repeatedly revise the lyrics until desirable results are obtained. To facilitate the process, *Youling* provides version control so that users can create lyrics with peace of mind.

4 Conclusion

In this paper, we demonstrate *Youling*, an AI-assisted lyrics creation system. *Youling* can accept multifaceted controlling attributes to control the content and format of generated lyrics. In the lyrics generation process, *Youling* supports traditional one pass full-text generation mode as well as an interactive generation mode. Besides, the system also provides a revision module which enables users to revise the undesirable sentences or words of lyrics repeatedly. We hope our system can assist music creators in lyrics creation and inspire other developers to make better solutions for NLG applications.



高中三年
(The three years of high school)
仿佛就在昨天
(It seems like yesterday)
我的五班
(I'm in Class five)
原来就是二号线
(It's Line two)

我走在校园的湖边
(I'm walking by the lake on campus)
你是否还记得那一天
(Do you remember that day?)
想起我们初次见面
(Think of the first time we met)
为什么泪水模糊了双眼
(Why are the tears blurring eyes?)

毕业的那年
(In the year of graduation)
我们相遇在美丽的校园
(We met on the beautiful campus)
听课的瞬间
(A moment in class)
心中又传来一张熟悉的笑脸
(A familiar smile appeared in my mind)

我走在校园的湖边
(I'm walking by the lake on campus)
你是否还记得那一天
(Do you remember that day?)
想起我们初次见面
(Think of the first time we met)
为什么泪水模糊了双眼
(Why are the tears blurring eyes?)
我坐在这间教室里面
(I'm sitting in this classroom)
你和他们学着去表演
(You and them learn to act)

直到最后我们才慢慢发现
(We slowly found out until the end)
青春没有终点
(There is no end to youth)
我们也不再留恋
(And we don't care anymore)
但只要相信思念会永远
(Just believe that missing will last forever)

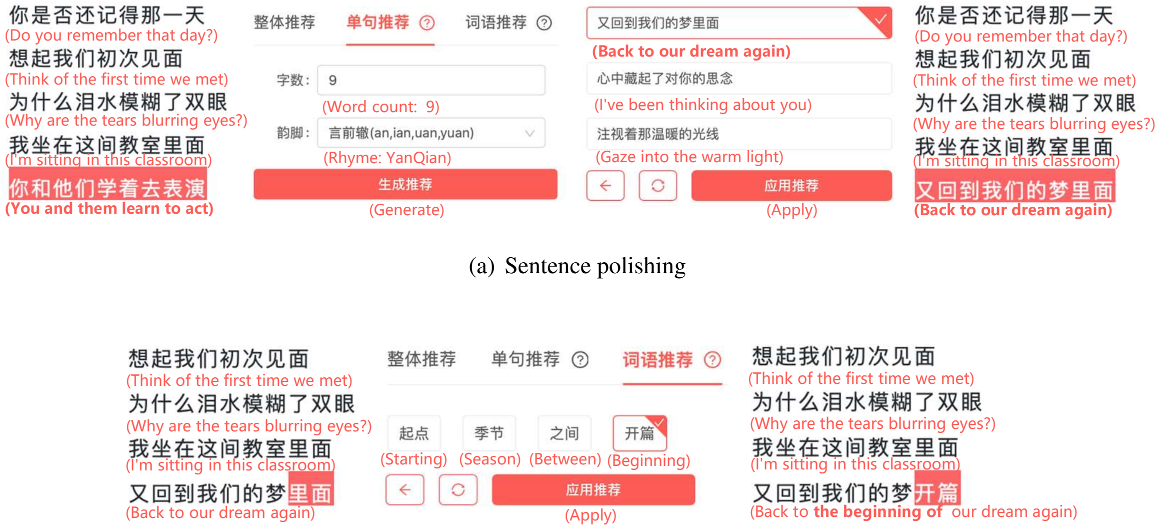
1 ian 我走在校园的湖边
(I'm walking by the lake on campus)
2 ian 你是否还记得那一天
(Do you remember that day?)
3 ian 想起我们初次见面
(Think of the first time we met)
4 an 为什么泪水模糊了双眼
(Why are the tears blurring eyes?)
5 ian 我坐在这间教室里面
(I'm sitting in this classroom)
6 an 你和他们学着去表演
(You and them learn to act)
7
8 ian 直到最后我们才慢慢发现
(We slowly found out until the end)
9 ian 青春没有终点
(There is no end to youth)
10 ian 我们也不再留恋
(And we don't care anymore)
11 uan 但只要相信思念会永远
(Just believe that missing will last forever)
12

就算分手都值得纪念
(Even a breakup is worth remembering)
爱情的风筝断了线
(Kite of love broke its string)
飞向更远我们之间
(Fly farther between us)

(a) Full-text generation

(b) Interactive generation

Figure 5: Examples of the two generation modes. (a) Full-text generation: this mode will generates three full-text candidates for users to choose. (b) Interactive generation: the mode generates following three sentences conditioned on the preceding context.



你是否还记得那一天
(Do you remember that day?)
想起我们初次见面
(Think of the first time we met)
为什么泪水模糊了双眼
(Why are the tears blurring eyes?)
我坐在这间教室里面
(I'm sitting in this classroom)
你和他们学着去表演
(You and them learn to act)

字数： 9
(Word count: 9)
韵脚： 言前辙(an,ian,uan,yuan)
(Rhyme: YanQian)

又回到我们的梦里面 ✓
(Back to our dream again)
心中藏起了对你的思念
(I've been thinking about you)
注视着那温暖的光线
(Gaze into the warm light)

你是否还记得那一天
(Do you remember that day?)
想起我们初次见面
(Think of the first time we met)
为什么泪水模糊了双眼
(Why are the tears blurring eyes?)
我坐在这间教室里面
(I'm sitting in this classroom)
又回到我们的梦里面
(Back to our dream again)

(a) Sentence polishing

想起我们初次见面
(Think of the first time we met)
为什么泪水模糊了双眼
(Why are the tears blurring eyes?)
我坐在这间教室里面
(I'm sitting in this classroom)
又回到我们的梦里面
(Back to our dream again)

起点 (Starting)
季节 (Season)
之间 (Between)
开篇 ✓ (Beginning)

想起我们初次见面
(Think of the first time we met)
为什么泪水模糊了双眼
(Why are the tears blurring eyes?)
我坐在这间教室里面
(I'm sitting in this classroom)
又回到我们的梦开篇
(Back to **the beginning of** our dream again)

(b) Word polishing

Figure 6: Examples of revision module polishing sentences and words in lyrics. Users can select undesirable sentences or words, then ask the system to generate candidates for selected contents conditioned on the context.

References

Pablo Samuel Castro and Maria Attarian. 2018. Combining learned lyrical structures and vocabulary for improved lyric generation. *arXiv preprint arXiv:1811.04651*.

Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A hierarchical attention based seq2seq model for chinese lyrics generation. In *Pacific Rim International Conference on Artificial Intelligence*, pages 279–288. Springer.

Hsin-Pei Lee, Jhih-Sheng Fang, and Wei-Yun Ma. 2019. iComposer: An automatic songwriting system for Chinese popular music. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 84–88, Minneapolis, Minnesota. Association for Computational Linguistics.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. *arXiv preprint arXiv:2004.08022*.

Vasileios Lioutas and Andriy Drozdyuk. 2019. Copy this sentence. *arXiv preprint arXiv:1905.09856*.

Xu Lu, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A syllable-structured, contextually-based conditionally generation of chinese lyrics. *arXiv preprint arXiv:1906.09322*.

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

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. GhostWriter: Using an LSTM for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924, Lisbon, Portugal. Association for Computational Linguistics.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2018. Evaluating creative language generation: The case of rap lyric ghostwriting. pages 29–38.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Asir Saeed, Suzana Ilić, and Eva Zangerle. 2019. Creative gans for generating poems, lyrics, and metaphors. *arXiv preprint arXiv:1909.09534*.

Liang-Hsin Shen, Pei-Lun Tai, Chao-Chung Wu, and Shou-De Lin. 2019. Controlling sequence-to-sequence models - a demonstration on neural-based acrostic generator. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 43–48, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Yinhe Zheng, Rongsheng Zhang, Xiaoxi Mao, and Minlie Huang. 2020. A pre-training based personalized dialogue generation model with persona-sparse data. In *Proceedings of AAAI*.