Sudowoodo: a Chinese Lyric Imitation System with Source Lyrics

Yongzhu Chang[1], Rongsheng Zhang[12]*, Lin Jiang[3], Qihang Chen [3]
Le Zhang [1], Jiashu Pu [1]
[1] Fuxi AI Lab, NetEase Inc., Hangzhou, China
[2] Zhejiang University
[3] Music AV Lab, NetEase Inc., Hangzhou, China
{changyongzhu, zhangrongsheng}@corp.netease.com

Abstract

Lyrics generation is a well-known application in natural language generation research, with several previous studies focusing on generating accurate lyrics using precise control such as keywords, rhymes, etc. However, lyrics imitation, which involves writing new lyrics by imitating the style and content of the source lyrics, remains a challenging task due to the lack of a parallel corpus. In this paper, we introduce ***Sudowoodo***, a Chinese lyrics imitation system that can generate new lyrics based on the text of source lyrics. To address the issue of lacking a parallel training corpus for lyrics imitation, we propose a novel framework to construct a parallel corpus based on a keyword-based lyrics model from source lyrics. Then the pairs *(new lyrics, source lyrics)* are used to train the lyrics imitation model. During the inference process, we utilize a post-processing module to filter and rank the generated lyrics, selecting the highest-quality ones. We incorporated audio information and aligned the lyrics with the audio to form the songs as a bonus. The human evaluation results show that our framework can perform better lyric imitation. Meanwhile, the *Sudowoodo* system and demo video of the system is available at Sudowoodo and https://youtu.be/u5BBT_j1L5M.

1 Introduction

AI creative assistants are artificial intelligence systems that can learn from large amounts of text data to understand human language and culture and use this knowledge to create content such as story generation (Alabdulkarim et al., 2021; Zhu et al., 2020), poetry writing (Guo et al., 2019; Liu et al., 2019b; Yang et al., 2019), grammar and spelling checking (Patil et al., 2021), etc. In addition, AI creative assistants can also assist in songwriting (Potash et al., 2015; Zhang et al., 2020; Shen et al., 2019) by learning from numerous songs, understanding human emotional expression, and creating music in a similar writing style to humans. Previous research (Castro and Attarian, 2018; Watanabe et al., 2018; Manjavacas et al., 2019; Fan et al., 2019; Li et al., 2020; Zhang et al., 2020, 2022) has focused on generating lyrics based on specified keywords (e.g., *Snow*), lyrics styles, themes, or user input passages, which generate new lyrics with limited control over the content. However, in actual music production, users sometimes adapt excellent songs by adding their own creativity while remaining the original lyrical structure, resulting in new lyrics. This requires stronger control over the source lyrics such as text content, emotion, and fine-grained writing styles.

To address this issue, this paper demonstrates ***Sudowoodo*** [1] (a Pokémon with the ability to imitate) a Chinese lyrics imitation generation system based on source lyrics. *Sudowoodo* is typically based on the Encoder-Decoder framework, where the encoder encodes the text and attributes of the source lyrics, and the decoder generates the imitated lyrics. However, since we only have the source lyrics and not the target ones, the parallel corpus is lacking to train the imitation model. To solve the problem, we also propose a method for constructing aligned training samples, which generated the target lyrics from the extracted keywords of source lyrics using a keywords-based lyrics generation model.

Specifically, we first collect the source lyrics corpus D_k from the Internet [2] and utilize the keyword extraction method described in Section 2.1 to extract keywords from source lyrics. And we train a keywords-based model, named Model$_{K2L}$, which can generate lyrics from given keywords. Then, we generate the target lyrics $D_{k'}$ using the Model$_{K2L}$. Finally, we train a lyrics imitation model with the aligned lyrics corpus ($D_{k'}$, D_k) based on the encoder-decoder framework. In addition, to improve the quality of generated lyrics

* Corresponding Author

[1] https://en.wikipedia.org/wiki/Talk%3ASudowoodo
[2] https://music.163.com/

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 99–105
December 6-10, 2023

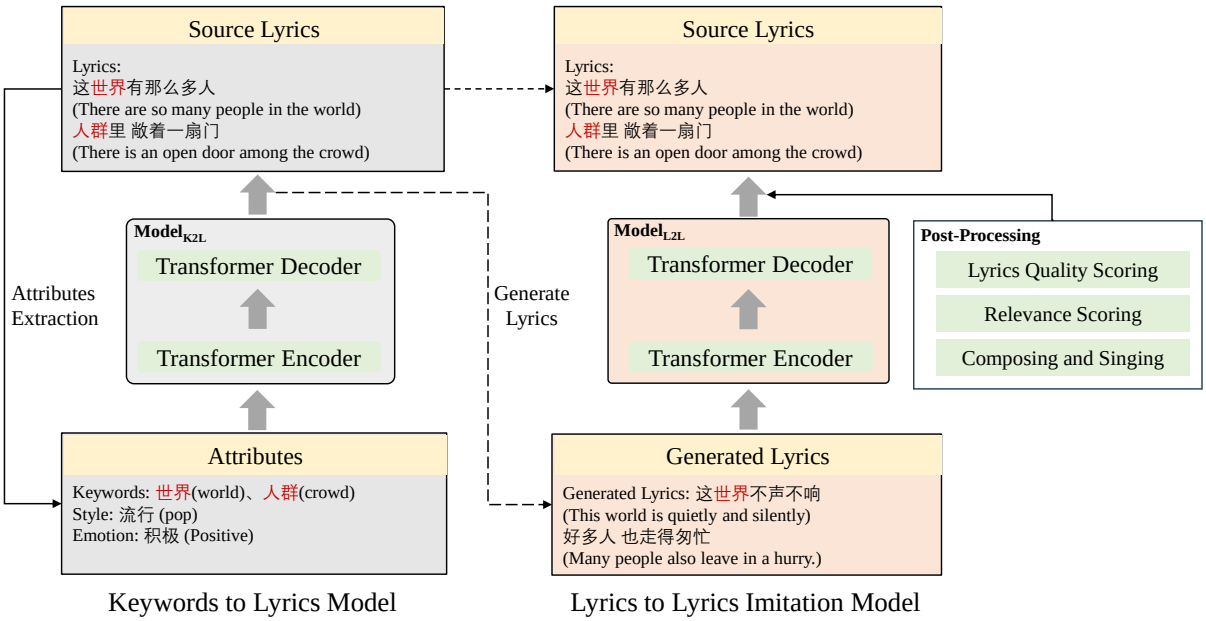

Figure 1: The framework of *Sudowoodo* system proposes in this paper. **Model**$_{K2L}$ denotes a model for generating lyrics based on keywords, while **Model**$_{L2L}$ represents the generation model from source lyrics to imitation lyrics.**Encoder** refers to the encoding portion of the Encoder-decoder architecture, while **Decoder** represents the decoding portion. **Post-processing** is mainly aimed at the imitation lyrics generated based on the Model$_{L2L}$.

and better showcase the results, we also employ post-processing modules including lyrics quality scoring and relevance scoring. Meanwhile, to provide a more intuitive understanding of the generated lyrics through imitation, we incorporate audio information (the vocals and melody of the source song) and align the lyrics with the audio to produce a complete song.

The main contributions of the *Sudowoodo* system are summarized as follows:

- We present a lyric imitation tool that generates new lyrics end-to-end based on source lyrics. Furthermore, we explore the addition of musical information to the generated lyrics in order to create songs. Sample songs can be heard at the songs of Sudowoodo.
- We propose a novel framework for constructing a parallel lyrics corpus for imitation based on the keyword-based model. The results of the human evaluation show the efficacy of the imitation model trained on the basis of this parallel lyrics corpus.
- The *Sudowoodo* system and demo video can be available at Sudowoodo and https://youtu.be/u5BBT_j1L5M.

2 Framework

The *Sudowoodo* system consists of two models and a post-processing module, as illustrated in Figure 1: **Model**$_{K2L}$, **Model**$_{L2L}$, and **Post-Processing**. These modules will be described in greater detail below.

2.1 Data Preparation

In this study, we obtain a dataset of $800k$ Chinese lyrics of various styles from the Internet, including pop, hip-hop, rap, etc. After filtering out lyrics less than 100 characters in length and removing duplicates, we are left with $600k$ unique lyrics. We denote the processed lyrics corpus as D_k.

As depicted in the attribute extraction section of Figure 1, when conducting attributes extraction for the source lyrics, we extract not only the keywords of the source lyrics but other attributes such as style and emotion. To extract keywords from the source lyrics, we first segment the lyrics into multiple bars. We then apply KBERT (Liu et al., 2019a) based on distiluse-base-multilingual-cased-v1 [3] to extract a subset of the keywords from each bar. We extract 5 keywords for each bar. In addition, we rank the

[3]Multilingual knowledge distilled version of multilingual Universal Sentence Encoder. Supports 15 languages: Arabic, Chinese, Dutch, English, French, German, Italian, Korean, Polish, Portuguese, Russian, Spanish, and Turkish. https://www.sbert.net/docs/pretrained_models.html

keywords according to their scores and select the top 10% scoring keywords as the keywords for the whole song. In this process, we utilize the Jieba [4] as a word separation tool. For other information, we train a classifier model to acquire attributes such as emotion and style from source lyrics. Finally, we construct a parallel corpus dataset by extracting keywords, style, and emotion from the lyrics and aligning these attributes with source lyrics to form paired data (D_A, D_K), where D_A represents the corpus composed of the extracted attributes of the corresponding source lyrics. The size of this dataset is $600k$.

2.2 Models

We first train a model, named Model$_{K2L}$, using the paired data (D_A, D_K) to generate lyrics based on keywords and their associated attributes such as emotion and style. Then, we acquire three new lyrics through Model$_{K2L}$ for each source lyric with random keywords extracted from the source lyric. The new lyrics are aligned with the source lyric and keywords to form paired data. All the lyrics generated by Model$_{K2L}$ are collected as D'_K. Consequently, we construct a parallel corpus dataset (D'_k, D_k) with a size of $1800k$. Meanwhile, during the training of Model$_{L2L}$, we encoder D'_k and the write styles of D_k, while the decoding side targets D_k.

Initialization: To improve the model's performance and generate more fluent text, we initialize the model with a self-developed transformers-based pre-training model. Note that the structure of the pre-trained model is consistent with GPT-2 [5], containing 210 million parameters with 16 layers, 1024 hidden dimensions, and 16 self-attention heads. The model is pre-trained on 30G of Chinese novels collected from the internet, using a vocabulary of 11400 words and a maximum sequence length of 512.

Training: Due to the lack of direct alignment corpus from lyrics to lyrics, we cannot train a seq2seq encoding and decoding model directly. Therefore, we propose a novel training strategy, as shown in Figure 1. The framework comprises two models for training. Firstly, a keyword-to-lyrics model, named Model$_{K2L}$, is used to generate aligned lyrics from source lyrics, with keywords and attributes such as style and emotion encoded into a latent semantic space and then decoded into source lyrics. The Model$_{K2L}$ utilizes an encoder-decoder architecture with the keywords, style, and emotion serving as encoder inputs and the source lyrics as decoder outputs, with training loss as shown in Equation 1. Secondly, an end-to-end lyrics imitation model, called Model$_{L2L}$, is trained using the aligned corpus (D'_k, D_k) constructed from Model$_{K2L}$ and also utilizes the encoder-decoder architecture. The Model$_{L2L}$ encodes D'_k and the attributes of the source lyrics into the encoder, with the source lyrics serving as the decoder output and training loss as shown in Equation 2.

$$L_{K2L} = -\sum_{D_k} logP(y_i|D(E(k_i, W_i))) \quad (1)$$

$$L_{L2L} = -\sum_{(D_{k'}, D_k)} logP(y_i|D(E(x_i, k_i, W_i))) \quad (2)$$

Where E encodes lyrics, keywords, and writing styles into latent representation, and D decodes the latent representation into lyrics. k_i means the keywords and the W_i represents the writing styles such as emotion and style in source lyrics. x_i indicates the lyrics in D'_k. D_k is the dataset of source lyrics, and D'_k is lyrics generated from Model$_{K2L}$.

Inference: During inference, the input to Model$_{K2L}$ is controlled by keywords and writing style and is typically less than 512 in length. In contrast, Model$_{L2L}$'s inputs include the source lyrics, which can easily exceed the length of 512. The most intuitive approach is to truncate the inputs after incorporating the keywords and writing style. However, this approach would be easy to obscure the controlling elements such as writing style and keywords. To address this issue, when the lyrics exceed 512 minus the length of the writing style and keywords, we truncate the last bar of the source lyrics to ensure that the input to the model does not exceed 512. It is worth noting that the last bar of the source lyrics often repeats the previous content, so this truncation does not significantly impact the generated lyrics.

Decoding Strategy: We use a top-k sampling strategy with a sampling temperature of 0.8 and a value of k of 10. Additionally, to prevent the model from easily generating duplicate words, we apply a sampling penalty technique proposed by Yadong et al. (2021), which only penalizes the first 200 words. In lyrics generation, although the model

[4] https://github.com/fxsjy/jieba

[5] https://openai.com/blog/gpt-2-1-5b-release/

can learn the specific format, which is the number of lines and a number of words per line based on the source lyrics, we perform format control decoding to ensure that the generated lyrics have the same format as the source lyrics. To do this, we record the number of lines and words in the generated lyrics and adjust the *[SEP]* and *[EOS]* logits in each decoding step.

2.3 Post-processing

After the model training is finished, we can use the source lyrics, provided keywords, and writing styles to generate limitation lyrics with Model$_{L2L}$. We utilize the top-k sampling method at decoding to generate candidate lyrics. For each input, the model generates 10 samples. Then we re-rank the samples according to the following scores.

Lyrics Quality Scoring: To filter high-quality lyrics, we train a classification model to determine whether a song lyric is a high-quality lyric and consider its confidence score as the Lyrics score, which is called S_{Lyric}, for re-rank. Inspired by QiuNiu (Zhang et al., 2022), we utilize popular and classic lyrics as positive samples, while lyrics with very few plays are negative samples. The experimental results indicate that the model gives a high confidence score when the lyrics contain beautiful sentences and rhetorical devices.

Relevance Scoring: In this paper, we introduce a method called $S_{relevance}$ to measure the semantic similarity between source lyrics and generated lyrics. To calculate $S_{relevance}$, we use the sentence transformer to obtain sentence vectors for both the source and generated lyrics, and then calculate the cosine similarity to rank the relevance. This method allows us to evaluate the quality of the generated lyrics in terms of their semantic similarity to the original lyrics.

Finally, we apply an anti-spam filter to the lyrics and use a combination of scores to sort them as shown in Equation 3. We then select the top 3 results as the final output. This post-process allows us to identify the most high-quality lyrics according to our criteria.

$$Score = w_1 * S_{Lyric} + w_2 * S_{relevance} \quad (3)$$

which the w_1 and w_2 denote the weights of the corresponding scores. In this paper, we set w_1 to 0.7 and w_2 to 0.3.

Composing and Singing: In order to evaluate the quality of lyrics generated from the Model$_{L2L}$, we annotate popular songs by extracting various musical features, including melody, chord progression, key, structure, and phrasing, using both general music theory [6] and more advanced analytical techniques. Based on these features, we then use intelligent composition (Song et al., 2009) techniques to generate melodies similar to those in the source style. Additionally, we use matching arrangement techniques, virtual vocal timbre selection, and mixing parameter adjustment to produce a fully synthesized song that includes accompaniment and singing. Finally, we incorporated audio information and aligned the lyrics with the audio to form the songs as shown in Figure 2. We can enjoy it in songs mode of Sudowoodo!

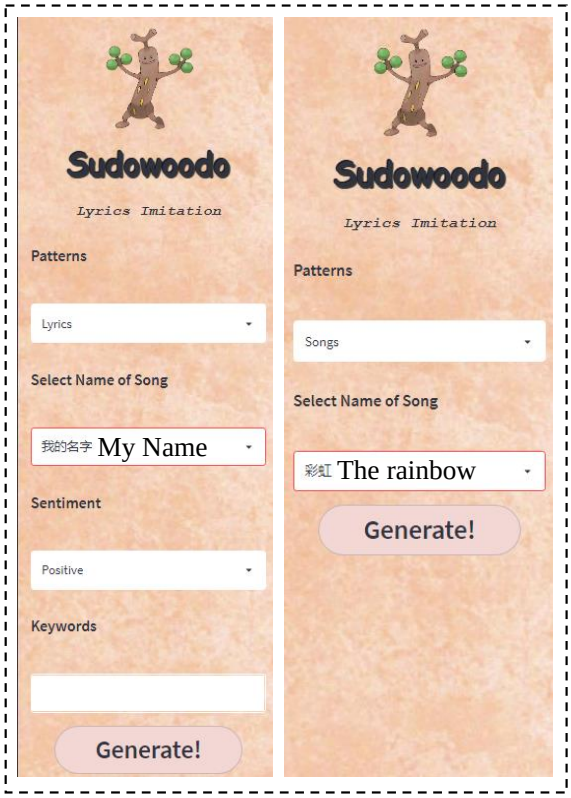

Figure 2: The interface of Sudowoodo.

3 Results of the Experiment

We conduct an ablation study to evaluate the framework proposed in this paper.

Metrics: We evaluate the generated lyrics from four perspectives: (1) *Thematic:* The relevance of the imitation lyrics to the theme of the source lyrics, including love, friendship, family inspiration, etc. (2) *Fluency:* It refers to the smoothness

[6] https://www.ipr.edu/blogs/audio-production/what-are-the-basics-of-music-theory/

	Theme (avg.)	Flu (avg.)	Logic (avg.)	Overall (avg.)	Best (%)
Model$_{K2L}$	4.168	4.103	**3.480**	4.078	32.75
Model$_{L2L}$	4.250	**4.160**	3.460	**4.153**	**34.25**
w/o WS	**4.275**	4.108	3.415	4.148	33

Table 1: Human evaluation results of Ablation. The scores in the table are the average scores of the three annotators. "Best" indicates that the model achieves Top-1 in the validation dataset for the same source lyric using three end-to-end lyrics imitation methods. *Flu* means *Fluency*, and *Theme* is *Thematic* in metrics. WS means the writing styles such as keywords, style, and emotion.

Figure 3: The instance of imitation lyrics in Lyrics mode. We enter the "爱情 (love)" and "自由 (freedom)" as keywords. As you can see from the picture, not all of the keywords entered are necessarily used. The Red color in Chinese and English indicates keywords.

and naturalness of the language used in the lyrics. In evaluating the fluency of a song's lyrics, we consider factors such as the fluency of the words and the rhythmic structure of the sentences. (3) *Logic:* It refers to the coherence and smoothness of scene transitions in the lyrics. To evaluate the logic of a song's lyrics, we consider whether consecutive sentences describe a single scene. If m consecutive sentences describe a scene, we argue that those sentences are reasonable within logic. If n consecutive groups of m sentences are found to exist within n different scenes, the lyrics are considered to have a high degree of smooth scene transitions overall. The number of scene jumps [7] can measure the logic of the song. (4) *Overall:* The overall scoring of a song's lyrics.

Results: We sample 100 lyrics from the source dataset and generate three imitation lyrics for each source lyric. We invite 3 professional lyricists to score each of the 300 lyrics based on *Thematic*, *Fluency*, *Logic*, and *Overall*. The score ranges from 1-5, with 5 being the best and 1 being the worst. The results are shown in Table 1, where all scores are averages for one song. We observe that the thematic and comprehensive scores of Model$_{L2L}$ exceeded those of Model$_{K2L}$. Additionally, we also verify the effect of the model, which uses only lyrics as input without keywords and writing style, and find that the addition of keywords improves the fluency of the generated lyrics. When the model is used to generate lyrics for the same lyrics using all three end-to-end methods, we observe that the method based on generated lyrics outperforms the keywords in 67.25% of cases. It indicates that generated lyrics for training can improve the performance of a lyric imitation model.

4 Demonstration

This section demonstrates how the *Sudowoodo* system works.

The user interface for this demo is shown in Figure 2. As an imitation demo, it offers limited interaction with the user. The *Sudowoodo* system operates in two modes: ***Lyrics*** and ***Songs***. In ***Lyrics*** mode, the user is required to select the source

[7] Scene jumps occur when consecutive sentences describe different things or switch abruptly between different sensory perspectives, resulting in an unnatural or jarring transition.

Figure 4: An example of Songs mode, with a player that plays the rendered song with imitation lyrics above the generated lyrics.

lyrics and the desired sentiment for the generated lyrics. Additionally, the user may provide keywords, which are typically space-separated phrases such as "自由 爱情 (*freedom love*)" or, alternatively, left blank. When generating lyrics, the *Sudowoodo* system takes into account the writing style of the selected source lyrics, including its theme, rhymes, and provided keywords, as well as the desired sentiment. The provided keywords are highlighted for easy identification. Note that not all provided keywords are necessarily used in the generated lyrics. In ***Songs*** mode, the user can select the name of the source lyrics to hear the generated lyrics as a song. Due to technical limitations, the lyrics are rendered offline. In this paper, we apply three different AI singers to provide the sounds. Finally, the user can click "Generate!" to produce the output.

Next, we show some generated examples in Figure 3.

Lyrics: The leftmost column of the display lyrics represents the source lyrics selected by the user, while the three columns on the right show the generated imitation lyrics. If the user has entered keywords, these will be highlighted in red within the generated lyrics. This demo can generate smooth, high-quality lyrics in a format and writing style similar to the source lyrics for each generation.

Songs: Figure 4 shows the results in Songs mode. As the real-time rendering of songs is a challenging task, we have performed offline rendering for this demo. A player is provided above the generated lyrics, which can be clicked on to hear the resulting song after rendering with the imitation lyrics. In the future, we aim to integrate real-time rendering of songs to create a true lyric imitation system that can take source lyrics and generate corresponding songs. More experiences are available in Sudowoodo.

5 Conclusion

In this paper, we describe *Sudowoodo*, a Chinese lyric imitation system that supports two modes: Lyrics and Songs. In Lyrics mode, users can input keywords to generate imitated lyrics based on existing lyrics. In Songs mode, *Sudowoodo* uses an unspecified technology to generate music that accompanies the imitated lyrics to create a complete song. To address the lack of a lyric-to-lyric alignment corpus, we propose a novel training framework structure to construct a parallel corpus for lyric imitation. Additionally, we apply Chinese pre-trained GPT-2 for initialization. To improve the quality of the generated lyrics, we employ a post-processing module to sort the generated results and select the highest quality ones. Finally, We audio-aligned some of the imitation lyrics to form songs!

Acknowledgements

This work is supported by the Key Research and Development Program of Zhejiang Province (No.

2022C01011). We would like to thank the anonymous reviewers for their excellent feedback. We are very grateful for the professional markers provided by NetEase Crowdsourcing[8].

References

Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. 2021. Automatic story generation: Challenges and attempts. *ArXiv*, abs/2102.12634.

Pablo Samuel Castro and Maria Attarian. 2018. Combining learned lyrical structures and vocabulary for improved lyric generation. *ArXiv*, abs/1811.04651.

Haoshen Fan, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A hierarchical attention based seq2seq model for chinese lyrics generation. *ArXiv*, abs/1906.06481.

Zhipeng Guo, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jian na Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. Jiuge: A human-machine collaborative chinese classical poetry generation system. In *Annual Meeting of the Association for Computational Linguistics*.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. *ArXiv*, abs/2004.08022.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. In *AAAI Conference on Artificial Intelligence*.

Yusen Liu, Dayiheng Liu, and Jiancheng Lv. 2019b. Deep poetry: A chinese classical poetry generation system. In *AAAI*.

Enrique Manjavacas, Mike Kestemont, and Folgert Karsdorp. 2019. Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In *INLG*.

Kavita. T. Patil, R. P. Bhavsar, and B. V. Pawar. 2021. Spelling checking and error corrector system for marathi language text using minimum edit distance algorithm.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. Ghostwriter: Using an lstm for automatic rap lyric generation. In *Conference on Empirical Methods in Natural Language Processing*.

Liangbin 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 *Conference on Empirical Methods in Natural Language Processing*.

Xudong Song, Wanchun Dou, and Wei Song. 2009. A workflow framework for intelligent service composition. *2009 Workshops at the Grid and Pervasive Computing Conference*, pages 11–18.

Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. 2018. A melody-conditioned lyrics language model. In *North American Chapter of the Association for Computational Linguistics*.

Xing Yadong, Xiao-Xi Mao, Li Le, Lin Lei, Chen Yanjiang, Shuhan Yang, Chen Xuhan, Kailun Tao, Li Zhi, Gongzheng Li, Jiang Lin, Liu Siyan, Zhao Zeng, Minlie Huang, Changjie Fan, and Hu Zhipeng. 2021. Kuileixi: a chinese open-ended text adventure game. In *Annual Meeting of the Association for Computational Linguistics*.

Zhichao Yang, Pengshan Cai, Yansong Feng, Fei Li, Weijiang Feng, Elena Suet-Ying Chiu, and Hong Yu. 2019. Generating classical chinese poems from vernacular chinese. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, 2019:6155–6164.

Le Zhang, Rongsheng Zhang, Xiao-Xi Mao, and Yongzhu Chang. 2022. Qiuniu: A chinese lyrics generation system with passage-level input. In *ACL*.

Rongsheng Zhang, Xiao-Xi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2020. Youling: an ai-assisted lyrics creation system. In *Conference on Empirical Methods in Natural Language Processing*.

Yutao Zhu, Ruihua Song, Zhicheng Dou, Jianyun Nie, and Jin Zhou. 2020. Scriptwriter: Narrative-guided script generation. In *Annual Meeting of the Association for Computational Linguistics*.

[8]https://fuxi.163.com/productDetail/17