Yu Sheng: Human-in-Loop Classical Chinese Poetry Generation System

Jingkun Ma Runzhe Zhan Derek F. Wong*
NLP²CT Lab, Department of Computer and Information Science, University of Macau
nlp2ct.{jingkun, runzhe}@gmail.com, derekfw@um.edu.mo

Abstract

The development of poetry generation system mainly focuses on enhancing the capacity of generation model. However, the demands of customization and polishing are generally ignored, which highly reduces the scope of application. In this work, we present Yu Sheng, a web-based poetry generation system that is featured a human-in-loop generation framework, providing various customization options for users with different backgrounds to engage in the process of poetry composition. To this end, we propose two methods and train the models that can perform constrained generation and fine-grained polishing. The automatic and human evaluation results show that our system has a strong ability to generate and polish poetry compared to other vanilla models. Our system is publicly accessible at: https://yusheng.cis.um.edu.mo.[1]

1 Introduction

Classical Chinese poetry is written with specific rules such as historical period, phonology, etc., attracting researchers from different fields to study its writing mechanism. Apart from the in-depth analysis of poetry writing process, the automatic generation of classical Chinese poetry is an emerging research task of open-ended text generation.

Several research lines of poetry generation has been investigated in the past few years such as combinatory process (Queneau, 1961), template-based method (Gervás, 2001), machine learning (Levy, 2001), and deep learning (Yi et al., 2018). Recently, the pre-trained language model is utilized to capture poetry prior knowledge (Tian et al., 2021) and build the downstream generation models. Previous research focuses on building the generation model with an end-to-end pipeline, polishing (Yan, 2016) is an essential part of poetry generation that can be helpful to reduce linguistic errors and enhance the aesthetic. Although there are numerous online systems that can generate classical Chinese poetry based on the keywords,[2][3][4] few works integrate the polishing function into the generation system. Jiuge[5] allows the user to make the adjustment by providing candidate words but is limited to word-level replacements, which is not flexible in adjusting the poetry-level polishing. Moreover, the design of polishing services should be user-oriented. The current system (Zhipeng et al., 2019) allows users to select the candidate words, but it is not friendly for non-professional users due to their insufficient background in poetry composition, which is also limited for professional users to polish the results.

To alleviate the aforementioned problems, we design and implement a human-in-loop classical Chinese poetry generation system, Yu Sheng. Yu Sheng not only supports the poetry generation with diverse genres and constraints, but also provides fine-grained polishing functions where the unsatisfactory parts can be refined with automatic adjustment. In order to build models that are capable to handle the above functions, we propose two methods for constraint integration and poetry polishing. Specifically, the global attention mechanism is proposed to integrate different kinds of constraints. For building the polishing model, we utilize the multi-task learning approach to train the model with mask prediction and sentence reconstruction tasks. Moreover, the data augmentation techniques are also used to alleviate the scarcity of task-specific data. Both automatic and human evaluation results have shown the effectiveness of our proposed polishing-based generation model. By deploying the model trained with the above methods, Yu Sheng is characterized by customizable

*Corresponding author
[1]Our demonstration video is available at: https://vimeo.com/776525586.
[2]https://www.aichpoem.net/#/shisanbai/ctcouplet
[3]http://moonbrewer.com/poem/
[4]https://tssc.sinaapp.com/
[5]http://jiuge.thunlp.org

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics System Demonstrations, pages 57–66
May 2-4, 2023

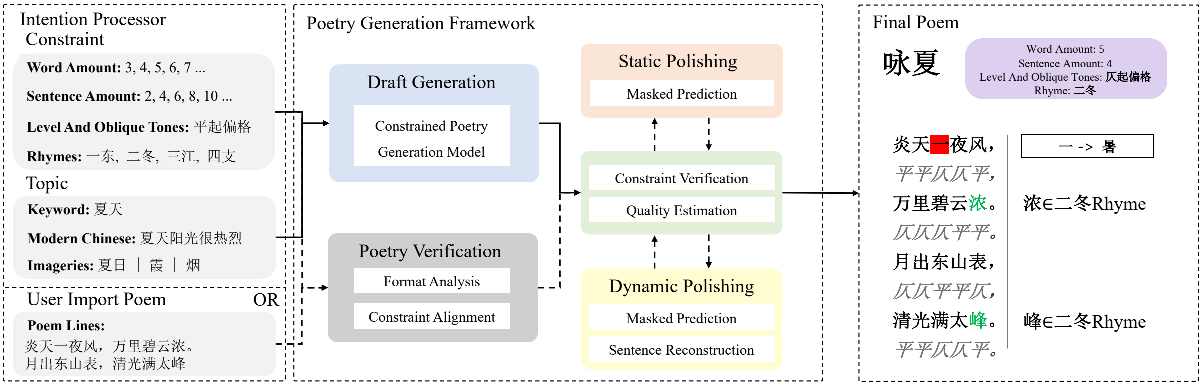


Figure 1: The overall architecture of system Yu Sheng. User input will be handled with an intention processor before sending into the essential poetry generation framework. Note that not all the modules in the generation framework are necessarily used before completing a composition.

generation and human-in-loop polishing, aiming to further improve the generation quality and flexibility of classical Chinese poetry.

2 Methodology

The generation pipeline of Yu Sheng includes two stages: poetry generation and poetry polishing. It not only supports automatic poetry generation from given information, but also accepts any existing poems as input for polishing. In this section, we first describe the proposed methods for building the generation and polishing models and introduce the data augmentation methods for compensating the data used in supervised training.

2.1 Constrained Poetry Generation

Global Attention As a typical literacy genre with predetermined features, poetry is designed, drafted, and polished under the subtle influence of different rigid features. These indispensable features not only affect the structures and multi-level morphology, but also the sentence semantics, and phonology beauty. Hence, the algorithm of extracting and integrating these features in different dimensions is crucial to the whole process of poetry generation. Global features of poetry include “Word Amount”, “Sentence Amount”, “Level and Oblique Tones”, and “Rhyme”. Apart from some features that can be directly measured, we specifically build a rule-based tool to extract other abstract features, such as “Level and Oblique Tones” and “Rhyme”, by using an ancient rhyming dictionary called PingShuiYun (Branner, 2006). Specifically, we leverage the dictionary to annotate the tone of each word in poetry dataset and then automatically predict the rhyming condition determined by tone of end word in each poem sentence. In this section, we use Transformer-based architecture (Vaswani et al., 2017) to illustrate the proposed method for generation and polishing task. To introduce all the global features into the generation process as the constraints, they are formulated as text input $\mathbf{D}$ and encoded into the vectorized representation $\mathbf{S}$ with the model embedding. Finally, the self-attention mechanism is used to integrate these constraints, which can be formally expressed as:

$$\mathbf{S} = \mathbf{D}_y \oplus \mathbf{D}_j \oplus \mathbf{D}_p \oplus \mathbf{D}_r \oplus \mathbf{D}_i \tag{1}$$

$$\mathbf{G} = \textsc{Embed}\,(\mathbf{S}) \tag{2}$$

$$\mathbf{C} = \textsc{Self-Att}\left(\mathbf{G}\mathbf{W}^Q, \mathbf{G}\mathbf{W}^K, \mathbf{G}\mathbf{W}^V\right) \tag{3}$$

where $\mathbf{D}_y, \mathbf{D}_j, \mathbf{D}_p, \mathbf{D}_r$, and $\mathbf{D}_i$ denote “Word Amount”, “Sentence Amount”, “Level and Oblique Tones”, “Rhyme”, and user input, respectively. The constraints are interacted with the inputs by self-attention calculation $\mathbf{C} = \textsc{Self-Att}(\cdot)$, acting as the global information to steer the generation of each time step t. The generation model is optimized by cross-entropy loss with constraints $\mathbf{C}$ as follows:

$$\mathcal{L}_{\text{gen}} = -\sum_{t=1}^{|\hat{\mathbf{y}}|} \log p\left(\hat{\mathbf{y}}_t \mid \mathbf{C}; \theta_{\text{gen}}\right) \tag{4}$$

where $\hat{\mathbf{y}}$ is the draft produced by the generation model θ_{gen}.

2.2 Poem Polishing

The polishing model θ_{pol} iteratively optimizes the generation draft $\mathbf{y}$ following the constraints $\mathbf{C}$.

Mask Prediction Strategy Previous works use the independent polishing model under practical scenarios. Deng et al. (2020) propose a BERT-based polishing scheme that highly relies on the mask-prediction pre-training task. However, this method generates the candidate words without considering original input and constraints. Li et al. (2020) successfully introduce the input into the polishing process through additional embedding modules. But it is not efficient to design an additional embedding module for newly introduced constraints due to computational cost, nor to ignore user input from the prior generation and only conduct masked prediction tasks. To this end, we design a polishing model that is capable to update the specific word and reconstruct the entire poetry. Within decoding process of polishing task, the model should provide word candidates predictions $\mathbf{w}$ for mask position $\mathbf{m}$ and generate polished sentence according to previous obtained poetry draft $\hat{\mathbf{y}}$ and feature constraints $\mathbf{C}$. For training the polishing model, a multi-task learning approach is applied to jointly optimize the loss of mask prediction and sentence reconstruction, which can be formulated as:

$$\mathcal{L}_{\mathtt{mask}} = -\sum_{t=1}^{|\mathbf{m}|} \log p\left(\mathbf{w}_t \mid \mathbf{m}_t, \mathbf{C}; \theta_{\mathtt{pol}}\right) \quad (5)$$

$$\mathcal{L}_{\mathtt{rec}} = -\sum_{t=1}^{|\mathbf{y}|} \log p\left(\mathbf{y}_t \mid \hat{\mathbf{y}}_t, \mathbf{C}; \theta_{\mathtt{pol}}\right) \quad (6)$$

where $\mathbf{y}$ is the ground truth. $\mathcal{L}_{\mathtt{mask}}$ and $\mathcal{L}_{\mathtt{rec}}$ are the cross-entropy loss of mask-prediction task and sentence reconstruction task, respectively. Finally, the model is optimized by two kinds of loss:

$$\mathcal{L} = \mathcal{L}_{\mathtt{mask}} + \mathcal{L}_{\mathtt{rec}} \quad (7)$$

2.3 Data Augmentation

Generation Augmentation For constrained generation task, combining poetry data with all the pre-determined constraints could limit the scope of application. In this case, hard constraints would restrict the diversity of model input, leading to limited customization choices in the deployment stage, let alone the generation case without constraint. Hence we propose a constraint-level masked-style data augmentation method. It masks the constraints with equal probabilities so that it can increase the diversity of data with different sets of pre-determined constraints. The model is forced to learn to generate diverse candidates based on augmented types of constraints, meanwhile, it also supports more flexible customization of poetry generation.

Polishing Augmentation Since there is no labeled data for training the polishing model, a pseudo dataset is built by masking the random tokens of ground truth text. To construct the masked sentences, we set up the mask ratio as 0.5, resulting in an equal probability for masked and unmasked tokens. Due to the randomness of the masking operation, the masked sentences will not cover all polishing scenarios. Hence, we further quadruple the polishing data for each poetry by masking the tokens that are different from the original pseudo data, covering a wider range of polishing requests.

2.4 Setup

We exploit GPT-2 (Radford et al., 2019) as our fundamental model and follow the pretrain-finetuning paradigm to build the downstream generation and polishing models. Our training corpus consists of 1,004,039 poems which are built based on open-source data[6][7]. The amount of training data reaches 4,016,159 instances after data augmentation techniques. For decoding settings, we employ Top-p sampling method with $p = 1$, and the temperature parameter $t = 1.0$ is applied to the softmax layer.

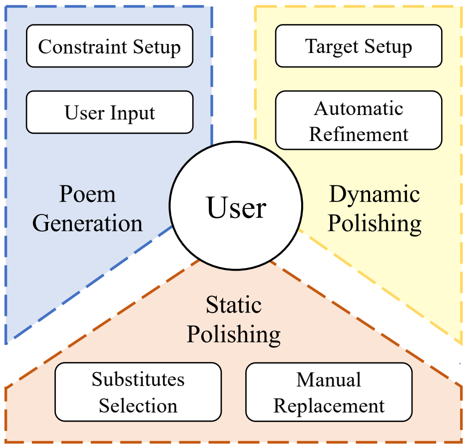


Figure 2: Overview of system functionality.

3 Functionality Design

As shown in Figure 2, we specially design three main functions for Yu Sheng system, meeting the

[6] https://github.com/Werneror/Poetry
[7] https://github.com/chinese-poetry/chinese-poetry

Model	Diversity (Distinct) ↑		F-Score ↑			
	Distinct-1	**Distinct-2**	**Word Amount**	**Sentence Amount**	**Tones**	**Rhyme**
Transformer	1.10	14.39	13.84	2.10	3.38	33.25
+ *Polish*	1.39	23.10	45.20	30.60	**99.96**	25.67
GPT-2	2.42	44.62	76.18	86.96	99.76	68.53
+ *Polish*	**2.45**	44.67	75.98	86.00	99.56	67.51
+ *Gen. Aug.*	2.42	44.97	**80.56**	87.24	99.72	**70.02**
+ *Polish Aug.*	2.44	44.61	74.51	**87.67**	99.72	69.11
+ *Both*	2.44	**45.02**	79.76	86.52	99.64	69.42

Table 1: Automatic evaluation results of generated poems with same input and constraints. Overall, the polishing model enhances the generation quality. "Gen." and "Aug." denote "generation" and "augmentation", respectively.

user's demand at different levels. As the fundamental function, the user can easily obtain poetry by indicating expected features. As for people who have advanced requirements of aesthetics, they can pay more effort into poetry generation by using two polishing modes for optimizing the generation draft. The dynamic polishing provides the functionality of poetry-level refinement. The user can simply select the unsatisfactory words or sentences as the target, and then the system will perform automatic polishing for updating the chosen parts. If the user wants to dive deeper and make subtle changes, static polishing can recommend substitutes for the unsatisfactory text, where the user can choose the proper candidate and manually replace the corresponding parts. Moreover, professional user can import their own poetry work and obtain inspiration from polishing candidates provided by Yu Sheng. To facilitate polishing process, the system will provide explanatory information of each feature to user for understanding the candidates. For improving the inference speed in a practical environment, we enable a inherit-based decoding trick for polishing task, which significantly reduces response time by preserving the unchanged tokens at each decoding step. The whole generation pipeline makes up of the aforementioned functions, and the specific use cases of each function are introduced in Appendix.

4 Evaluation

4.1 Automatic Evaluation

4.1.1 Data and Metrics

Data 2,522 constraint-poem pairs are filtered out from the original data as the test data, which is independent of the training corpus. This test set is passed to original vanilla Transformer and GPT-2 generation model to obtain initial poetry drafts and then used as the input of the polishing model. To simulate the users' operation, we randomly choose the polishing part of each poem.

Distinct is used to evaluate the diversity of generated poetry. It is calculated by dividing the number of words by the number of unique words in a sentence, which can be formulated as:

$$\mathrm{Distinct}(n) = \frac{\text{Count(unique } N\text{-gram)}}{\text{Count(word)}} \tag{8}$$

F-score Since our system focuses on generating the poetry according to poetry-related constraints, F-score is utilized to evaluate the model accuracy and recall of constraint integration. We calculate micro-F1 score to evaluate the integration of different constraints due to large inter-class gap.

4.1.2 Results

The evaluation results are as shown in Table 1. In general, GPT-2 is a stronger generation model compared to vanilla Transformer. Hence, we choose GPT-2 as a testbed to further verify the effectiveness of different polishing strategies. Although all metrics show that basic polishing model improves the generation performance of vanilla Transformer, original GPT-2 model cannot benefit from the basic polishing model. The reason may be that the difficulty of polishing increases with the improvement of generation quality. Encouragingly, the polishing model trained with augmentation data improves the performance of most metrics, demonstrating the effectiveness of proposed augmentation methods.

Although the polishing model learns the diversified polishing cases from the augmentation data, both sentence reconstruction and mask prediction tasks cannot improve the generation diversity due to their objective functions. Hence, the scores of Distinct-1 and Distinct-2 are fluctuating for augmentation models. But the proposed methods gain stable improvement from the perspective of morphology, phonology, and lexical-related factors.

Overall, our proposed polishing model can further improve the quality of GPT-2 generation results by mending constraint errors with the aforementioned augmentation strategies, which demonstrates the effectiveness of proposed methodology. Therefore, we can realize that the system is capable to generate high-quality poetry together with flexible customization. To prove the competitiveness of our system, we also compare it with the existing system in Section 4.2.

4.2 Human Evaluation

4.2.1 Data and Metrics

The goal of poetry generation and polishing is conforming to human preference. To better understand the model's behavior from human points of view and evaluate the effectiveness of polishing, we conduct a human evaluation of the polished poetry. Besides model evaluation, we provide system comparison regarding generated poems based on poetry customization options that all systems support.

For each model and system, we randomly sample 60 poems from the generation results as evaluation data. Then we invite native Chinese speakers with poetry knowledge as evaluators to conduct evaluation using four metrics: Poeticness, Fluency, Meaning, and Coherency. Three evaluators are assigned to evaluate poems generated under different models and another four evaluators are assigned to conduct system comparison.

The rating criteria for each metric are listed below:

- **Poeticness** (Poe.): Score the current poem based on the sense of beauty: 0 (tedious), 1 (sense of beauty exists in partial sentence), 2 (all poem sentences contain a sense of beauty).
- **Fluency** (Flu.): Score the current poem based on the phonology and fluency: 0 (phonology crash), 1 (partial sentence is unreadable), 2 (all the sentences can be read fluently with smooth phonology).
- **Meaning** (Mea.): Score the current poem based on its relevance to the user intention: 0 (digress from the main subject), 1 (partial sentences are irrelevant), 2 (relevant to the subject).
- **Coherency** (Coh.): Score the current poem based on its coherency: 0 (all sentences are independent), 1 (partial sentence does not follow surrounding context), 2 (all the sentences are coherent with the other context).

Model	**Poe.**	**Flu.**	**Mea.**	**Coh.**
Transformer	0.88	0.87	0.43	0.63
+ *Polish*	0.62	0.52	0.55	0.50
GPT-2	1.27	0.82	0.55	0.50
+ *Polish*	1.12	0.82	1.13	0.77
+ *Gen. Aug.*	1.18	0.93	1.10	0.88
+ *Polish Aug.*	1.10	0.85	0.98	0.82
+ *Both*	**1.30**	**1.02**	**1.20**	**1.08**

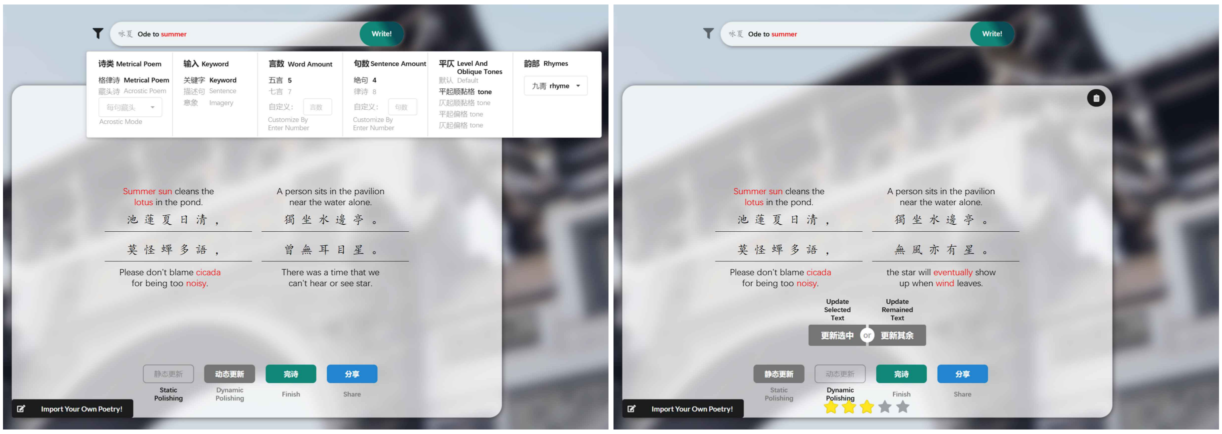
Table 2: Human evaluation results of the poems generated by different models. By training the models with augmentation data, the poems generated/polished by our model are better than GPT-2 baseline. "Gen." and "Aug." denote "generation" and "augmentation", respectively.

System	**Poe.**	**Flu.**	**Mea.**	**Coh.**
Jiuge	1.40	1.25	1.15	1.17
Yu Sheng	**1.62**	**1.53**	**1.55**	**1.62**

Table 3: Human evaluation results of the poems generated by our system Yu Sheng and another system Jiuge. The poems were generated by each system based on poetry constraints that both systems are capable to customize. Overall quality of the poems generated by Yu Sheng is competitive with the other systems in terms of each metric.

4.2.2 Results

The human evaluation results are as shown in Table 2. It shows that polished poems achieve the highest score on poeticness, which supports the rationality of high F-score in the automatic evaluation. In our system, the model can generate poetry with accurate morphology by following the constraints given by the user, such as "Word Amount" and "Sentence Amount", and other high-level rhythmic options like "Rhyme" and "Level and Oblique Tones" could enhance the sense of beauty. Regarding fluency and coherency, the augmentation data highly improve their scores since the polishing model is trained by diversified constraints. System comparison in Table 3 shows high competitiveness of our



(a) Constrained generation. (b) Poetry polishing.

Figure 3: Use cases of constrained poetry generation and polishing. Constraints customized in filter bar and topic words entered in input bar steer the constrained poetry generation. Refinement results are reflected on the demonstration panel, replacing the previous draft.

system regarding the same constraints supported by both systems.

5 Discussion

Figure 3 presents a case of using Yu Sheng to customize poetry generation with constraints.

Constrained Poem Generation As shown in Figure 3a, the user may first use the feature selector to customize all generation constraints. Then the user enters the topic words according to the type of keywords and click generate button, Yu Sheng will return the results in the demonstration panel.

Poem Polishing We also showcase the poetry polishing procedure on the automatic generation results or user-imported poem as illustrated in Figure 3b. The users can import their own poem into workspace by entering keywords and poem sentences into the import module. When the user switches to Dynamic Polishing mode and chooses "自", "有" as the unsatisfactory tokens, the system would perform polishing action and return the result with poetry-level adjustment. The user also can switch to static polishing mode, and then select specific word or sentence. The system would recommend different substitutes for words and sentences for replacement. Preliminary analysis of the poem would be automatically generated in the workspace for users to understand current features and structural information before performing next operation. At the end of poetry generation, Yu Sheng also allows users to evaluate the current composition and propose a beautiful share poster for sharing the poem work. More details can be founded in the Appendix due to page limitations.

More Backbone Models In this work, we focus on exploring the generation and polishing capability of general casual language model. Since other pre-trained models also inherit the Transformer architecture, we provide the results of vanilla Transformer to prove the generality of proposed method. The polishing model can greatly enhance the quality of model outputs in the first stage. However, the polishing model becomes less effective for high-quality poems. We are also passionate to explore in the future work.

6 Conclusion and Future Work

We propose Yu Sheng, a comprehensive, human-in-loop classical Chinese poetry generation system. Yu Sheng establishes a generation pipeline that covers multi-dimensional demands of generation and polishing, enabling amateurs to engage in the process of poetry generation, and provide inspirations for composition work of human poets. It provides the functionalities for users with different backgrounds to conduct poetry composition conveniently and flexibly. With the global attention mechanism and the human-in-loop poetry generation paradigm, Yu Sheng could be easily updated with diverse constraints and further extended to different languages. In the future, we will also collect the data and construct an annotated polishing corpus for training a more robust polishing model.

Limitations

The limitations of our system are three-folded. First, there is no public polishing dataset in the open-source community. To address this problem, we mask the tokens of human-written poetry to build the pseudo data. Since the polishing model aims to refine the machine-generated poetry, the pseudo data used to train our model is still far away from the realistic scenario compared to the annotated data. Secondly, polishing quality is hard to control due to knowledge background of users. Although poetry features can be easily checked and evaluated, aesthetics highly relies on the users' preferences. Non-professional users may hardly notice the subtle problem of the generated poetry and make the right decision on whether to polish it or not. Thirdly, polyphonic disambiguation is hard to solve due to the lack of phonology data, resulting mild corruption of sentence semantic.

Acknowledgements

This work was supported in part by the Science and Technology Development Fund, Macau SAR (Grant Nos. FDCT/0070/2022/AMJ, FDCT/060/2022/AFJ) and the Multi-year Research Grant from the University of Macau (Grant No. MYRG2020-00054-FST). This work was performed in part at SICC which is supported by SKL-IOTSC, and HPCC supported by ICTO of the University of Macau.

References

David Prager Branner. 2006. Introduction: What are rime tables and what do ti-ey mean? *The Chinese Rime Tables: Linguistic philosophy and historical-comparative phonology*, 271:1.

Liming Deng, Jie Wang, Hang-Ming Liang, Hui Chen, Zhiqiang Xie, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2020. An iterative polishing framework based on quality aware masked language model for chinese poetry generation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7643–7650. AAAI Press.

Pablo Gervás. 2001. An expert system for the composition of formal spanish poetry. In *Applications and innovations in intelligent systems VIII*, pages 19–32. Springer.

Robert P Levy. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*. Citeseer.

Piji Li, Haisong Zhang, Xiaojiang Liu, and Shuming Shi. 2020. Rigid formats controlled text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 742–751, Online. Association for Computational Linguistics.

Raymond Queneau. 1961. 100.000. 000.000. 000 de poemes. *Gallimard Series. Schoenhof's Foreign Books, Incorporated.*

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Huishuang Tian, Kexin Yang, Dayiheng Liu, and Jiancheng Lv. 2021. Anchibert: a pre-trained model for ancient chinese language understanding and generation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Rui Yan. 2016. i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2238–2244. IJCAI/AAAI Press.

Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3143–3153, Brussels, Belgium. Association for Computational Linguistics.

Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. Jiuge: A human-machine collaborative Chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Detailed Use Case

Constrained Generation Figure 1 presents a case study of using Yu Sheng to customize poetry generation task with constraints. As shown in Figure 1a, the user first uses the feature selector to customize all correlated features ("Poem Type" as "Metrical Poem", "Input Type" as "Keyword", "Word Amount" as 5, "Sentence Amount" as 4, "Level and Oblique Tones" as "平起黏格" and "Rhymes" as "九青"). Then the user enters the keyword "夏天"(Summer) as the topic word according to the chosen type of keywords and starts the generation by clicking the "Write" button on the right side of input bar.

Within a short time after receiving the generation request, Yu Sheng returns and presents the expected generated poem in the demonstration panel, which is shown in Figure 1b. The user can easily obtain diversified poems under the same constraints and topic setup by repeatedly clicking the "Write" button without limitation.

Poetry Polishing We also showcase the polishing functionalities as illustrated in Figure 2. With the upholding poem draft, a user stays in the static polishing mode and clicks on unsatisfactory word "無" and sentence "曾無耳目星". The system will provide different substitutes as shown in Figure 2a. The user can replace sentence "無風自有星" in the original draft with substitute "曾無耳目星" by clicking the preferred substitute block in the sentence recommendation panel. The updated poem is shown in 2b.

Then, the user switches to the dynamic polishing mode for automatic polishing. The words "自" and "有" are marked as targets as shown in Figure 2c. Then the user clicks "Update Selected Text" button to label the text with a low-quality tag. Yu Sheng will automatically polish these unsatisfactory words and returns the result with dynamic adjustment as shown in Figure 2d. Finally, the result meets the user's requirement after iterative polishing.

When the user decides to complete the current work, Yu Sheng allows the user to evaluate the presented composition by rating stars after clicking "Finish" button. The user can also click the "Share" button to generate a poster. The poster will pop up in the middle of the interface (Figure 2e, 2f) for downloading and sharing.

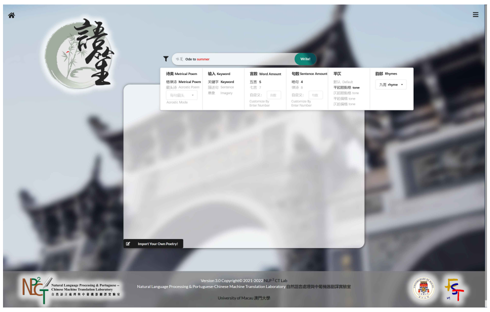
A.2 Examples

As shown in Table 1, we also present two examples obtained from our pipeline system.

B Revisions

To address the reviewers' concerns, we revised our paper as follows:

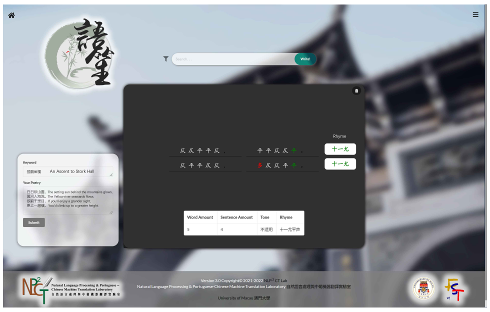
- We fixed the typos and added a link of system. Furthermore, a multilingual user interface has been added in Yu Sheng for serving the user comes from different language background.
- We offered more detailed explanations in terms of training loss and modelling approaches. Sentence reconstruction loss is formulated as the posterior probability of predicting words in the original poem sentence based on the draft and constraints.
- We described the rule-based method to extract tone and rhyme in Section 2.



(a) Constraint Setup

(b) Poem Generation Result

(c) Import Poem



(d) Poem Analysis

Figure 1: Use Case: Constrained Poetry Generation

Constrained Generation	Setup	Topic Word: 夏天; Word Amount: 5; Sentence Amount 7; Rhyme: 九青; Tone: 平平仄仄平　仄仄仄平平　仄仄平平仄　平平仄仄平
	Result	池蓮夏日清，獨坐水邊亭。莫怪蟬多語，曾無耳目星。
Static Polishing	Result	池蓮夏日清，獨坐水邊亭。莫怪蟬多語，無風自有星。
Dynamic Polishing	Result	池蓮夏日清，獨坐水邊亭。莫怪蟬多語，無風亦有星。

Constrained Generation	Setup	Topic Word: 月下獨飲; Word Amount: 5; Sentence Amount 7; Rhyme: 十一尤; Tone: 仄仄仄平平　平平仄仄平　平平平仄仄　仄仄仄平平
	Result	月下弄鳴弦，秋聲滿樹頭。今來何處去，日夜憶南州。
Dynamic Polishing	Result	月下弄鳴弦，秋聲滿樹頭。今來何處去，此夜憶南州。
Static Polishing	Result	月下弄鳴弦，秋聲滿樹頭。何因逢老病，此夜憶南州。
Dynamic Polishing	Result	月下弄鳴弦，秋聲滿樹頭。何因驚老病，此夜憶南州。

Table 1: Examples of pipeline generation.

(a) Static Polishing: Substitutes Recommendation

(b) Static Polishing: Result

(c) Dynamic Polishing: Text Indication

(d) Dynamic Polishing: Result

(e) Share Page

(f) Poster

Figure 2: Use Case: Poem Polishing