

Jiuge: A Human-Machine Collaborative Chinese Classical Poetry Generation System

Zhipeng Guo^{1*}, Xiaoyuan Yi^{1*}, Maosong Sun^{1†},

Wenhao Li¹, Cheng Yang¹, Jiannan Liang¹, Huimin Chen¹, Yuhui Zhang¹, Ruoyu Li²

¹Department of Computer Science and Technology, Tsinghua University, Beijing, China

Institute for Artificial Intelligence, Tsinghua University, Beijing, China

State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China

²6ESTATES PTE LTD, Singapore

Abstract

Research on the automatic generation of poetry, the treasure of human culture, has lasted for decades. Most existing systems, however, are merely model-oriented, which input some user-specified keywords and directly complete the generation process in one pass, with little user participation. We believe that the machine, being a collaborator or an assistant, should not replace human beings in poetic creation. Therefore, we proposed *Jiuge*, a human-machine collaborative Chinese classical poetry generation system. Unlike previous systems, *Jiuge* allows users to revise the unsatisfied parts of a generated poem draft repeatedly. According to the revision, the poem will be dynamically updated and regenerated. After the revision and modification procedure, the user can write a satisfying poem together with *Jiuge* system collaboratively. Besides, *Jiuge* can accept multi-modal inputs, such as keywords, plain text or images. By exposing the options of poetry genres, styles and revision modes, *Jiuge*, acting as a professional assistant, allows constant and active participation of users in poetic creation.

1 Introduction

Language is one of the most important forms of human intelligence, among different genres, poetry is a beautiful, poetic and artistic genre which expresses one's emotions and ideas with relatively fewer words. Across various countries, nationalities, and cultures, poetry is always fascinating, impacting profoundly on the development of human civilization.

Recently, researchers have worked on automatic poetry generation. Meanwhile, neural networks have proven to be powerful on this task (Zhang and Lapata, 2014; Wang et al., 2016; Yan, 2016;

Zhang et al., 2017; Yi et al., 2017). Besides the research value of exploring human writing mechanism and computer creativity, these models and systems could also benefit electronic entertainment, advertisement, and poetry education.

However, the recently released Chinese poetry generation systems are mainly model-oriented, which take some user inputs and directly complete the generation in one pass, resulting in poor user participation. Moreover, these systems generate poetry in fewer styles and genres, and provide limited options for users. For example, the *Daoxiangju* system¹ requires the user to determine the rhyme, which creates a barrier for beginners. The *Oude* system² simplifies the user's choices and only allows the input of a few options and genres. The *Microsoft Quatrain*³ provides limited candidates of a theme and each line, but it only supports the generation of quatrains.

Due to the lack of user participation, the above systems are mainly designed for entertainment. We argue that the leading role in literary creation should not be a machine, or at least not only a machine, because it is difficult for machines to handle the complex expressions of one's emotion and the use of images in poetic creation.

Rather than completely replace humans, a better way is to utilize the system to assist human creation. The human-machine collaboration mechanism in *Jiuge* system can not only improve the emotions and semantics of generated poems but also guide and teach beginners to understand the poetic creation process.

In summary, the contributions of our *Jiuge* system are as follows:

- **Multi-modal input.** *Jiuge* can accept multi-

* indicates equal contribution

† Corresponding author: M.Sun(sms@tsinghua.edu.cn)

¹<http://www.poeming.com/web/>

²<https://crl.ptopenlab.com:8800/poem/index>

³<http://duilian.msra.cn/jueju/>

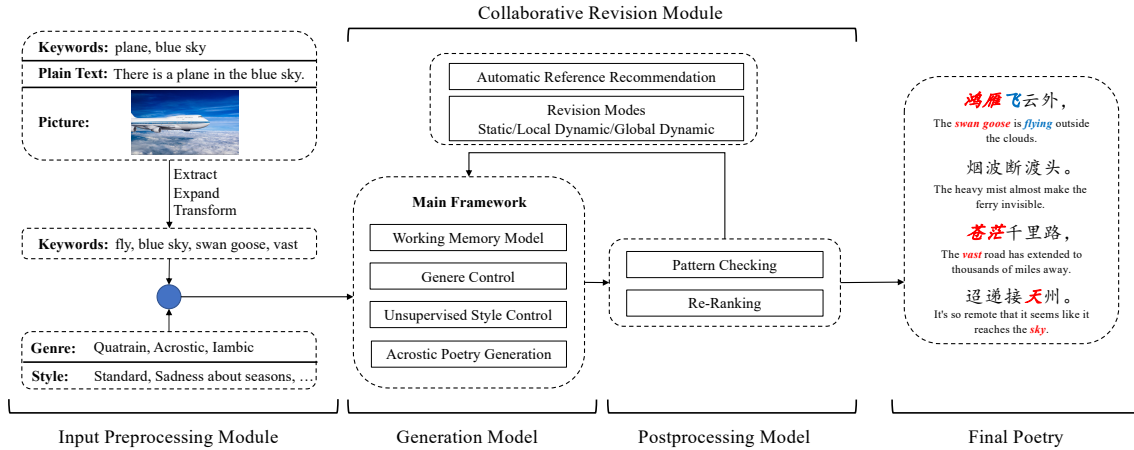


Figure 1: The architecture of Jiuge system.

modal input such as keywords, plain text, and even images. For modern concepts in the input, Jiuge utilizes a knowledge graph to map them into relevant keywords in classical Chinese poetry.

- **Various styles and genres.** Unlike previous systems, Jiuge provides more than twenty options of genre and ten options of style, and can generate more diverse poems.
- **Human-machine collaboration.** Jiuge supports human-machine collaborative and interactive generation. The user can revise the unsatisfied parts of a generated poem. In terms of the revision, Jiuge will dynamically update and re-generate the poem. During this process, Jiuge also offers candidate words and human-authored poetry as references for beginners.

2 Architecture

2.1 Overview

We show the overall architecture of Jiuge system in Fig. 1, which mainly consists of four modules: 1) *input preprocessing*, 2) *generation*, 3) *postprocessing* and 4) *collaborative revision*. Given the user-specified genre, style, and inputs (keywords, plain text or images), the preprocessing module extracts several keywords from the inputs and then conducts keyword expansion to introduce richer information. Jiuge also transforms the words in modern concepts, which are incompatible with classical Chinese poetry (written in ancient Chinese language), such as *refrigerator* and *airplane*, to appropriate relevant ones, e.g., *airplane* \rightarrow *fly*. With these preprocessed keywords, the generation module generates a poem draft. The postprocessing module re-ranks the candidates of each line

and removes the ones that do not conform to structural and phonological requirements. At last, the collaborative revision module interacts with the user and dynamically updates the draft for several times according to the user’s revision, to collaboratively create a satisfying poem.

We detail each module in the following parts.

2.2 Input Preprocessing Module

Keyword Extraction. Jiuge allows multi-modal input to meet the needs of generating poetry according to keywords, tweets or photos.

For plain text, we first use THULAC⁴ (Li and Sun, 2009) to conduct Chinese word segmentation and compute the importance $r(w)$ of each word w :

$$r(w) = [\alpha * ti(w) + (1 - \alpha) * tr(w)], \quad (1)$$

where $ti(w)$ and $tr(w)$ are the TF-IDF (Term Frequency-Inverse Document Frequency) value and TextRank (Mihalcea and Tarau, 2004) score calculated with the whole poetry corpus respectively. α is a hyper-parameter to balance the weights of $ti(w)$ and $tr(w)$. Afterwards, we select top K words with the highest scores.

For each image, we use the Aliyun image recognition tool⁵, which gives the names of five recognized objects with corresponding probability $s(w)$. Then we select top K words with the highest $s(w) \cdot r(w)$.

Keyword Mapping. The extracted or recognized keywords could be some modern concepts, such as *airplane* and *refrigerator*. Since these words never occur in the classical poetry corpus, the generation module will take them as a *UNK* symbol and generate totally irrelevant poems.

To address this problem, we build a Poetry Knowledge Graph (PKG) from Wikipedia data,

⁴<http://thulac.thunlp.org/>

⁵<https://ai.aliyun.com/image>

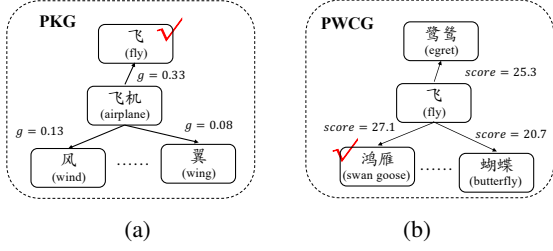


Figure 2: (a) A sampled subgraph of PKG. (b) A sampled subgraph of PWCG.

which contains 616,360 entities and 5,102,192 relations. 40,276 of these entities occur in our poetry corpus. Before keywords extension and selection, we first use PKG to map the modern concepts to its most relevant entities in poetry, to guarantee both quality and relevance of generated poems.

For a modern concept word w_i , we score its each neighbor word w_j by:

$$g(w_j) = tf_{wiki}(w_j|w_i) \cdot \log\left(\frac{N}{1+df(w_j)}\right) \cdot \arctan\left(\frac{p(w_j)}{\tau}\right), \quad (2)$$

where $tf_{wiki}(w_j|w_i)$ is the term frequency of w_j in the Wikipedia article of w_i , $df(w_j)$ is the number of Wikipedia articles containing w_j , N is the number of Wikipedia articles, and $p(w_j)$ is the word frequency counted in all articles. We give an example of mapping the modern word ‘‘airplane’’ in Fig. 2(a).

Keyword Extension. The generation module can handle multi-keywords input. More keywords could lead to richer contents and emotions in generated poems. Therefore, if the number of extracted keywords is less than K , we further conduct keywords extension. To this end, we build a Poetry Word Co-occurrence Graph (PWCG) as shown in Fig. 2 (b). This graph indicates the co-occurrence of two words in the same poem. The weight of the edge between two words is calculated according to the Pointwise Mutual Information (PMI) as follows:

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i) * p(w_j)}, \quad (3)$$

where $p(w_i)$ and $p(w_i, w_j)$ are the word frequency and co-occurrence frequency in poetry corpus. For a given word w , we get all its adjacent words w_k in PWCG and select those with higher values of $\log p(w_k) * PMI(w, w_k) + \beta * r(w_k)$ where β is a hyperparameter.

2.3 Generation Module

As shown in Fig. 3, the core component of the generation module is our proposed working mem-

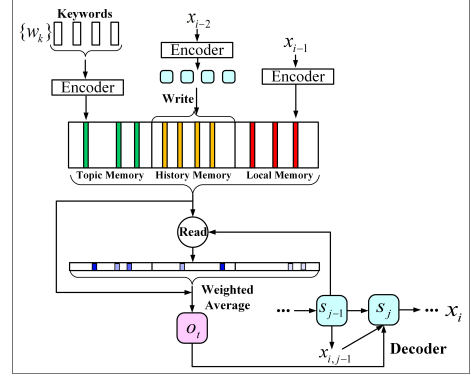


Figure 3: The simplified structure of the working memory model, which mainly comprise an encoder, a decoder and there memory components. x_i is the i -th line and $x_{i,j}$ is the j -word in the i -th line. Please refer to (Yi et al., 2018b) for more details.

ory model (Yi et al., 2018b), which takes at most K preprocessed keywords as input. The encoder maps each word or line into vector representations, and the decoder generates each line word-by-word. The topic memory stores keywords explicitly and independently, which can learn a flexible order and form of keywords expression. The history memory and local memory are dynamically read and written to improve the context coherence of generated poems.

Genre Control. Chinese classical poetry involves various genres, and each genre strictly defines the structural and phonological pattern of a poem, such as the length of each line, the tone of each word, and the number of lines. We use our designed genre embedding (Yi et al., 2018b) to disentangle the semantic content and the genre pattern. The genre embedding indicates the line length, word tone, and rhyme, which is fed to the decoder. By this way, we can train one model with all genres of poems and control the genre of generated poems by specifying a pattern.

Training patterns are automatically extracted from the corpus. For generating, we make the genre as a user option. However, the selection of rhyme may be difficult for users without relevant literature knowledge. Therefore, we train a classifier (implemented with a feedforward neural network) to predict an appropriate rhyme in terms of the keywords.

Unsupervised Style Control. Besides genres, there are also diverse styles in Chinese poetry such as *battlefield*, *romantic*, *pastoral*, etc. For certain contents or topics, creating different styles of poetry is one main user requirement. Since the labelled data is quite rare and expensive, we use

our proposed style disentanglement model (Yang et al., 2018) to achieve unsupervised style control. This method disentangles the style space into M different sub-spaces by maximizing the mutual information between the style distribution and the generated poetry distribution.

It is noteworthy that this method is transparent to model structures which can be applied to any generation model. In this stage, we employ it for the generation of Chinese quatrain poetry (*Jueju*), which will be extended to more genres in the future. We set the number of styles $M = 10$. After training, we manually annotate each style with some descriptive phrases, such as *sorrow during drinking* and *rural scenes*, to indicate the theme of the corresponding style. The style selection is also set as a user option.

Acrostic Poetry Generation In Chinese poetry, there is another special genre called acrostic poetry. Given a sequence of words $seq = (x_{0,0}, x_{1,0}, \dots, x_{n,0})$, which could be someone’s name or a blessing sentence, the author is required to create a poem using each word $x_{i,0}$ as the first word of each line x_i and the created poem should also conform to the genre pattern and convey proper semantic meanings.

The input for this genre is the sequence seq . As our generation module takes keywords as input, we first use pre-trained word2vec embeddings (Mikolov et al., 2013) to get K keywords related to seq according to the cosine distance of each keyword and the words in seq . Then we directly feed each $x_{i,0}$ into the decoder at the first step.

To alleviate the disfluency caused by this constraint, we generate the second word with the conditional probability: $p_{gen}(x_{i,1}|x_{i,0}) = p_{dec}(x_{i,1}|x_{i,0}) + \delta * p_{lm}(x_{i,1}|x_{i,0})$, where p_{dec} and p_{lm} are probability distributions of the decoder and a neural language model respectively.

If the length of the input sequence is less than n (the number of lines in a poem), we also use the language model to extend it to n words.

2.4 Postprocessing Module

Jiuge takes a line-to-line generation schema and generates each line with beam search (beam size= B). As a result, we can get B candidates for each line. We design a postprocessing module to automatically check and re-rank these candidates, and then select the best one, which is used for the generation of subsequent lines in a poem.

Pattern Checking. The genre embedding introduced in Sec. 2.3 cannot guarantee that generated poems perfectly adhere to required patterns. Thus, we further remove the invalid candidates according to the specified length, rhythm, and rhyme.

Re-Ranking. Our preliminary experiments show that the best candidate may not be ranked as the top 1 because the training objective is Maximum Likelihood Estimation (MLE), which tends to give the generic and meaningless candidates lower costs (Yi et al., 2018a). To automatically select the best candidate, we adopt the automatic rewarders we proposed in (Yi et al., 2018a), including a fluency rewarder, a context coherence rewarder, and a meaningfulness rewarder. Then the candidate with the highest weighted-average rewards given by them will be selected.

2.5 Collaborative Revision Module

We call the poem generated by the generation module in one pass the draft, since the user may revise it for several times to collaboratively create a satisfying poem together with the machine. We implement such collaboration with a revision module.

Revision Modes. Define a n -line poem draft as $X = (x_1, x_2, \dots, x_n)$, and each line containing l_i words as $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,l_i})$. At every turn, the user can revise one word in the draft. Then the revision module returns the revision information to the generation module which updates the draft according to the revised word. We implement three revision modes in terms of the updating way: *static*, *local dynamic*, and *global dynamic*.

- **Static updating mode.** The revision is required to meet the phonological pattern of the draft and the draft will not be updated except the revised word. The rhythm and rhyme information is given to the user together with the generated draft and the invalid revision will be alerted. During the beam search process, we also store top 10 candidate words in each position as recommendations.
- **Local dynamic updating mode.** If the user revises word $x_{i,j}$, then Jiuge will re-generate the succeeding subsequence, $x_{i,j+1}, \dots, x_{i,l_i}$, in the i -th line by feeding the revised word to the decoder for the revised position.
- **Global dynamic updating mode.**

If the user revises word $x_{i,j}$, Jiuge will re-generate all succeeding words, $x_{i,j+1}, \dots, x_{i,l_i}, \dots, x_{n,l_n}$. In terms of the revision position, e.g., the rhymed positions, a new phonological pattern may be adopted.

Thanks to this collaborative revision-updating process, the user can choose a mode and gradually revise the draft until she/he feels satisfied.

Automatic Reference Recommendation. For poetry writing beginners or these lacking professional knowledge, it is hard to revise the draft appropriately. To aid in the revision process, we implement an automatic recommendation component. This component searches several human-authored poems, which are semantically similar to the generated draft, for the user as references. Then the user could decide how to make revision with these references.

In detail, define a n -line human-created poem as $Y = (y_1, \dots, y_n)$ and a relevance scoring function as $rel(x_i, y_i)$ to give the relevance of two lines. Then we return N poems with the highest relevance score $rs(X, Y)$, calculated as:

$$rs(X, Y) = \sum_{i=1}^n rel(x_i, y_i) + \gamma * I(Y \in D_{master}), \quad (4)$$

where D_{master} is a poetry set of masterpieces, I is the indicator function, and the hyper-parameter γ is specified to balance the quality and relevance of searched poems. For more details of the relevance scoring function, we refer the reader to our previous work (Liang et al., 2018).

3 Demonstration

We implement a webpage version of Jiuge⁶ which allows users to create diverse poems and share with the others conveniently. The initial page provides some basic options: multi-modal input and the selection of genre and style, as shown in Fig. 4.

Jiuge is easy to use. After selecting the input mode and providing corresponding content, the user can further choose a favourite genre and style⁷. As shown in Fig. 5 (a), the user inputs two keywords, *desert* and *cavalry*, and chooses to generate a quatrain with the normal style. Then the user clicks the “Generate Poetry” button. After a few seconds, the system returns the processed keywords and a poem draft.

⁶Our system is available at <https://jiuge.thunlp.cn/>.

⁷Temporarily only the Jueju genre supports multiple styles.

Figure 4: The initial page of Jiuge System.

In order to help the user collaboratively revise the generated poem, Jiuge provides some high-quality human-authored poems which are semantically similar to the generated one, as the references. The user can click the button to change the references. At this time, the user can select an unsatisfying word to be revised in the draft and then Jiuge will give some recommended revision candidates. Besides these candidates, other choices are also allowed. After selecting the revision mode introduced in Sec. 2.5, Jiuge will update or re-generate the draft according to the revised word. Through several turns of collaborative revision, the user and Jiuge work together to create a satisfying poem, as in Fig. 5 (b)⁸.

In addition, Jiuge also supports picture sharing. By clicking the “Share Poetry” button, Jiuge will print the created poem on a beautiful picture so that the user can share it with others.

4 Conclusion and Future Work

We demonstrate *Jiuge*, a human-machine collaborative Chinese classical poetry generation system. Our system accepts multi-modal input and allows deep user participation in the choice of various styles and genres, as well as in collaborative revision. With an easy-to-use web interface, the user can collaboratively create a satisfying poem together with the system. Instead of being a simple entertainment software, Jiuge takes a step towards professional AI assistant for poetic education.

We collect a large number of human usage records from the interface, laying the foundation for enhancing the collaborative creation method in the future. We will also continue to integrate more styles and extend the collaborative creation

⁸The poems in Fig.5 are manually translated for better demonstration. In practice, we utilize a neural translation model to conduct automatic translation.



Figure 5: The collaborative creation page: an example of revision. (a) The poem draft generated in one pass. (b) The poem after four turns of revision and updating.

to more genres.

5 Acknowledgements

We would like to thank Cunchao Tu, Yunqiu Shao and anonymous reviewers for their insightful comments. It is supported by Natural Science Foundation of China (NSFC) Grant 61532001 and the NExT++ project, the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative.

References

- Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for chinese word segmentation. *computational linguistics*, 35(4):505–512,.
- Jiannan Liang, Maosong Sun, Xiaoyuan Yi, Cheng Yang, Huimin Chen, and Zhenghao Liu. 2018. Neural network-based jiju poetry generation. In *Proceedings of the Seventeenth China National Conference on Computational Linguistics*, Changsha, China.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Zhe Wang, Wei He, Hua Wu nad Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016. Chinese poetry generation with planning based neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1051–1060, Osaka, Japan.
- 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*, pages 2238–2244, New York, USA.
- Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. 2018. Stylistic chinese poetry generation via unsupervised style disentanglement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3960–3969, Brussels, Belgium.
- Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating chinese classical poems with rnn encoder-decoder. In *Proceedings of the Sixteenth Chinese Computational Linguistics*, pages 211–223, Nanjing, China.
- Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018a. 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.
- Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Zonghan Yang. 2018b. Chinese poetry generation with a working memory mode. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4553–4559, Stockholm, Sweden.
- Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. 2017. Flexible and creative chinese poetry generation using neural memory. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1364–1373. Association for Computational Linguistics.
- Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Doha, Qatar.