A Sentiment and Style Controllable Approach for Chinese Poetry Generation

Yizhan Shao
syzmaxwell@seu.edu.cn
Southeast University
Nanjing, Jiangsu, China

Tong Shao
213181203@seu.edu.cn
Southeast University
Nanjing, Jiangsu, China

Minghao Wang
wmh@seu.edu.cn
Southeast University
Nanjing, Jiangsu, China

Peng Wang*
pwang@seu.edu.cn
Southeast University
Nanjing, Jiangsu, China

Jie Gao
213181437@seu.edu.cn
Southeast University
Nanjing, Jiangsu, China

ABSTRACT

Sentiment and style control are two vital aspects in automatic poetry generation. Excellent Chinese classical poetry should express a certain emotion and embody a specific style at the same time. Existing work still has deficiencies in controlling sentiment and style simultaneously. To address above issues, in this paper, we propose a novel approach for Chinese classical poetry generation, which can generate sentiment-controllable and style-controllable poems. First, it classifies hundreds of thousands of poems by style, sentiment, format, and primary keyword. Then, it utilizes masking self-attention mechanism to associate multiple tags and verses. Besides, it can generate metrical rhyming verses with distinctive sentiment and style characteristics according to the tag-set and secondary keywords. Finally, this approach is applied in *Chang Qing Yin*, which can collaborate with users to polish generated poems, providing alternatives automatically. Experimental results show that our approach performs well in sentiment and style control, and quality of generated poems outperforms several strong baselines.

CCS CONCEPTS

• **Computing methodologies → Natural language generation**.

KEYWORDS

Chinese poetry generation; sentiment control; style control

ACM Reference Format:
Yizhan Shao, Tong Shao, Minghao Wang, Peng Wang, and Jie Gao. 2021. A Sentiment and Style Controllable Approach for Chinese Poetry Generation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3481964

*Corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11...$15.00
https://doi.org/10.1145/3459637.3481964

1 INTRODUCTION

Chinese classical poetry assembles the characteristics of the brilliant culture of China. Diverse styles and rich sentiments make Chinese classical poetry attractive. Style, e.g., *Bold* and *Graceful*, is the essence of poetry, bringing different reading experiences to the readers. Sentiment is the soul of poetry, which is the most moving part of a poem. Thus, the same imagery may have various meanings in different poems because of style and sentiment. Style and sentiment complement each other. Therefore, to generate a high quality Chinese classical poem, both of them are indispensable.

Automatic Chinese classical poetry generation is a hot topic in text generation, and many models have been proposed. After RNN is early applied in poetry generation [17], more and more models are proposed and proved effective, e.g., Polish [14], planning [13], Seq2Seq [1], Working Memory [16]. Moreover, Li et al. propose SongNet based on predefined rigid formats [9]. Hu and Sun apply GPT-2 to learning formats [8]. To improve the diversity, MixPoet [15], QA-MLM [4] and Deep Poetry [11] are proposed. These models improve the quality of poetry but neglect sentiment and style control. Thus, as to style control, Cheng et al. propose a style-controllable model through unsupervised style disentanglement [3]. As to sentiment control, Chen et al. propose a sentiment-controllable model via temporal sequence module [2]. However, those studies always focus on one aspect, rather than exploring the linkage between sentiment and style. As for poetry generation systems, *Jiu Ge* [7] and *Yue Fu* [10] can meet the users' need, but also lacks consideration of the interaction between attributes.

To address above issues, we propose SSPG, which can control sentiment and style simultaneously. Firstly, corpus is classified by multiple attributes and trained via generation model. Then, tags of attributes are packaged into a unique tag-set, which is fully utilized when generating verses. Generation model leverages masking self-attention mechanism to guarantee strong attention to tag-set. Finally, SSPG is embedded into *Chang Qing Yin*, a poetry generation system which supports extracting keywords and polishing verses interactively. Results show that SSPG can simultaneously control the attributes of poetry well, and the generated verses are of high quality. Our contributions are as follows.

- It is the first attempt to control sentiment and style simultaneously when generating Chinese classical poems, significantly increasing the quality of poetry.

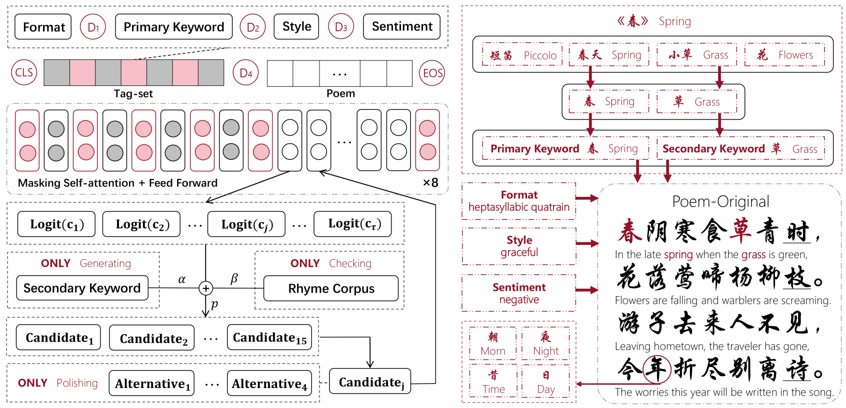

Figure 1: The overview of SSPG. Rhyming characters are underlined. Keywords are marked in red.

- In order to control attributes simultaneously, SSPG uses self-attention mechanism to achieve association between verses and multiple tags.
- We constructed a corpus delicately classified by style, sentiment, and primary keyword via comparing cosine distance of poem vector.
- Experiments show that poems generated by SSPG are easy to identify in sentiment and style, and the quality achieves new state-of-the-art performance.

2 METHODOLOGY

To generate poems with controllable attributes, we formulate the problem: we try to extract keywords $\mathbf{K} = (w_1, ..., w_n)$ from texts to guide the whole poem $\mathbf{P} = (v_1, ..., v_l)$, where v_i represents the i-th verse. n and l are the number of keywords and verses, respectively. The generated poem is supposed to be relevant to the secondary keywords $Skey$ and tag-set $\mathbf{T}$ which is composed of format $Form$, primary keyword $Pkey$, style Sty, and sentiment $Senti$.

We show the overview of SSPG in Figure 1. First, corpus is classified delicately to prepare for training. Next, SSPG forms a unique tag-set, autoregressively generating attribute-controllable verses. Finally, generated poems can be polished furthermore.

2.1 Poetry Classification

Firstly, we classify the corpus by sentiment and style. Cosine distance of the poem vector is used to measure similarity between labeled and unlabeled poems, since the poem vector integrates all tokens and ignores the specific meaning of word, better reflecting style and sentiment of poetry. Besides, we use BERT [5] to encode the original corpus. In addition to the cosine distance, we also adopt supervised learning paradigm to compare with, using BERT for hidden feature extraction and SVM for classification.

At the same time, corpus is classified by primary keyword and format. We artificially determine 30 standard primary keywords $SPKs$, e.g., *Spring*, *Wine*, etc. For each poem, SSPG determines the unique primary keyword by retrieving common synonyms of $SPKs$, ensuring that each poem has a unique $Pkey$.

Subsequently, SSPG performs unified standardized preprocessing on corpus and packages tags of each poem into a tag-set:

$$Form[D_1]Pkey[D_2]Sty[D_3]Senti$$

where $[D_1]$, $[D_2]$, $[D_3]$ are delimiters to distinguish tags, set as &, % and #. Similarly, preprocess each poem as follows:

$$[CLS]\mathbf{T}[D_4]\mathbf{P}[EOS]$$

where $[CLS]$ and $[EOS]$ delimiters represent the beginning and end. $[D_4]$ delimiter separates tag-set and verses. After standardizing, poems are uniformly pre-trained via generation model.

2.2 Poetry Generation

To start with, *Chang Qing Yin* obtains primary and secondary keywords from plain texts. Next, SSPG obtains tags of other attributes, uniformly forms tag-sets and generates metrical rhyming verses utilizing masking attention mechanism.

Keyword Preprocessing. Firstly, we compute the $Weight$ of each word w_i:

$$Weight(w_i) = a * TF(w_i) + (1 - a) * TR(w_i) \tag{1}$$

where $TF(w)$ is TF-IDF value and $TR(w)$ is TextRank [12] score. a is a hyper-parameter to balance the weights.

Secondly, considering many modern concepts cannot appear directly in poems, keywords need correcting. We collect two corpora in advance, namely character corpus and phrase corpus. Afterwards, we fix selected keywords cw_i:

$$cw_i = \arg\max_k s(w_i, k), k \in C \tag{2}$$

where C represents the corpora. $s(k, w_i)$ represents the resemblance, employing synonyms for calculating similarity.

Then, we distribute keywords into $Pkey$ and $Skey$:

$$pw_i = \arg\max_k s(cw_i, k), k \in SPKs \tag{3}$$

where pw_i is the assigned primary keyword. The purpose of prioritizing keywords is to make poems more similar to human writing.

Poetry Generating. First, SSPG packages $Pkey$, $Senti$, Sty and $Form$ into tag-set $\mathbf{T}$. Then, it takes advantage of GPT-2 to capture more information and achieve higher parallel efficiency. Generation model consists of the following steps: Input, composed of embedding vectors and positional encoding; Masking Self-attention, shield subsequent tokens and calculate attention by $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$; Feed Forward Neural Network, transfer parameters; Residual Network, fix

Table 1: Statistics of our corpus.

Attribute	Type	# of Poems	Type	# of Poems
Sentiment	Positive	58,396	Negative	59,554
Style	Bold	27,744	Graceful	13,392
	Idyllic	15,694	Farewell	16,329
	Homesick	16,153	Landscape	28,638

degradation of neural network. The masking self-attention mechanism facilitates strengthening attention to tag-set and increasing ongoing attention to distinctive labels, guaranteeing the implementability of our approach. Besides, top-k stochastic sampling strategy [6] is adopted. Generation model first opts candidates with higher weight $FinalW$, and then randomly samples one as output:

$$S(c_i) = \max s(c_i, k), k \in Skey \tag{4}$$

$$TempW(c_i) = Logit(c_i) + \alpha * (n-1) * S(c_i) \tag{5}$$

$$FinalW(c_i) = p * TempW(c_i) \tag{6}$$

where p is the penalty-factor to avoid repetition. α is a hyper-parameter, working with p to ensure full use of keywords.

Poetry Checking. We prepare a rhyme corpus R to select the rhyming tokens. Then, $FinalW$ will be modified as follows:

$$FinalW(c_i) = p * (TempW(c_i) + \beta * I_{R[j]}(c_i)) \tag{7}$$

where I is an indicator function and β is a hyper-parameter. $R[j]$ is the j-th slot of R, representing the index of the rhyme is j.

Poetry Polishing. After users specify the location, SSPG reacquires tag-set, secondary keywords, and the previous verses, generating several alternatives. Users can choose one of them or customize tokens, yielding interaction of human-computer.

3 EXPERIMENTS

3.1 Datasets and Settings

Our dataset is the collection of poems with the total number of 143,280. We exclude 25,330 poems with complex or inconspicuous characteristics. The details are shown in Table 1. We set the dimension of embedding vector, position vector and internal feature vector to 512; set the number of head in multi-headed self-attention to 8; set 8-layer transformer structure and the leaning rate to 1e-5; set the number of the standard primary keywords to 30; set the number of candidates to 15 when generating poems.

3.2 Evaluation of Quality

We collected 1,300 masterpieces as references and fine-tuned BLEU and Rouge to evaluate the quality. Apart from CQY (*Chang Qing Yin*), we compared WM [16], SPG [3], CVAE [2] and GPT-2 [8] as baselines. At the same time, we use CQY (w/o *Skey.*) with secondary keywords removed, CQY (w/o *Sty.*) with style tags removed and CQY (w/o *Senti.*) with sentiment tags removed for horizontal comparison. The specific score comparison is shown in Table 2.

It can be seen that *CQY* (w/o *Skey.*) outperforms others under BLEU and *CQY* (w/o *Senti.*) outperforms baselines under Rouge, which means our generated poems are more fluid and coherent due to the word-to-word generation mode. At the same time, it is clear that after removing the secondary keywords or one of the tags, the consistency and fluency are greatly improved. This is because the secondary keywords and the tags of attributes break the comfort zone in order to increase the connotation and diversity of poetry. But overall, *CQY* performs better than other models.

Table 2: Automatic evaluation results on quality of poetry.

Model	Sentiment	Style	BLEU	Rouge
WM	×	×	0.488	0.530
SPG	✓	×	0.459	0.481
CVAE	×	✓	0.594	0.576
GPT-2	×	×	0.617	0.612
CQY (w/o *Skey.*)	✓	✓	**0.672**	0.654
CQY (w/o *Senti.*)	×	✓	0.660	**0.665**
CQY (w/o *Sty.*)	✓	×	0.658	0.649
CQY	✓	✓	0.664	0.643

Table 3: Human evaluation results on quality of poetry.

Model	F	C	M	A	D
WM	3.11	3.20	2.87	2.94	2.82
SPG	2.94	3.11	3.25	3.37	3.14
CVAE	3.47	3.45	3.29	3.41	3.37
GPT-2	3.54	3.58	3.22	3.18	3.63
CQY (w/o *Skey.*)	**3.72**	**3.68**	3.21	3.23	3.21
CQY (w/o *Senti.*)	3.70	3.55	3.25	3.13	3.533
CQY (w/o *Sty.*)	3.62	3.60	3.10	3.21	3.49
CQY	3.66	3.58	**3.31**	**3.43**	**3.89**
Human	3.76	3.72	4.02^{++}	3.97^{++}	3.91

Besides, we also adopt manual evaluation and design 5 criteria: **Fluency** (F), **Coherence** (C), **Meaning** (M), **Aesthetics** (A) and **Diversity** (D). Each criterion needs to be scored from 1 to 5. For each model, we randomly select unified keywords to generate 20 poems for testing. For human, we select 20 poems containing the limited keywords. Then, 50 poetry enthusiasts are invited to evaluate them and divided into 5 groups. The specific average scores are shown in Table 3. The ICC (Intraclass Correlation Coefficient) of the 5 groups is 0.5, indicating an acceptable inter-annotator agreement.

It is evident that *CQY* achieves better results than other baseline models on manual evaluation. The scores of *CQY* (w/o *Skey.*), *CQY* (w/o *Senti.*) and *CQY* (w/o *Sty.*) for meaning, aesthetics, diversity are obviously weaker than the default, which confirms that secondary keywords can greatly improve semantic richness but will lose coherence and fluency. At the same time, our model gets close to human on fluency, coherence and diversity. However, in terms of meaning and aesthetics, there is still a big gap between generated poems and the poems created by human beings.

3.3 Evaluation of Attribute Control

We matched 12 different tag-sets according to 6 styles and 2 sentiments. For each tag-set we generate 20 poems, and the keywords and formats are randomly fixed. Subsequently, 50 poetry enthusiasts are invited to classify them and divided into 5 groups as well. We draw the corresponding matrix as shown in Figure 2, and compute the accuracy of sentiment and style classification, respectively. Each row represents the manual evaluation of corresponding model generations. Darker block indicates higher accuracy.

It can be seen that the dark areas are mainly concentrated near the diagonal, meaning that *CQY* can basically control style and sentiment of poetry. The error in sentiment classification is lower,

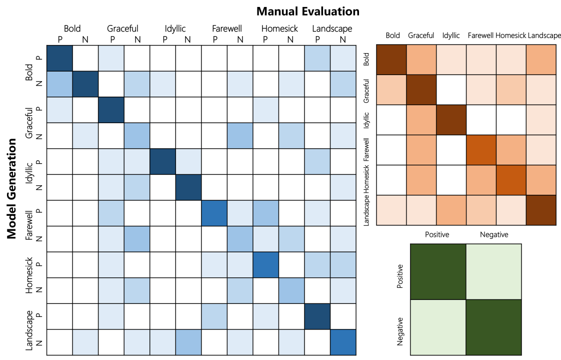

Figure 2: Human evaluation results on attribute control.

Table 4: Automatic evaluation results on attribute control.

Attribute	Model	A-CD	A-SVM	A-MLP
Style	SPG	0.64	0.60	0.58
	CQY	**0.71**	**0.68**	**0.64**
Sentiment	CVAE	0.78	0.80	0.77
	CQY	**0.85**	**0.89**	**0.83**

since the difference between *Positive* and *Negative* is relatively large. The higher error rate of style classification is mainly due to the blurring of the boundaries between styles. For example, *Farewell* poems and *Homesick* poems are very close, so precise and perfect classification cannot be achieved.

We also adapt automatic evaluation for attribute control. In each attribute, we compare with its baseline. For each model, we randomly generated 200 poems. We encode the corpus by BERT[5] and compare the cosine distance of the poem vector with the labeled poems. Besides, we utilize SVM (Support Vector Machine) and MLP (Multi-Layer Perceptron) to classify attributes as well. The accuracy is shown in Table 4. It is clear whether cosine distance similarity or supervised classification, *CQY* performs better in sentiment and style control. This is mainly due to our finely classified corpus and self-attention mechanism to ensure strong association with tags.

4 DEMONSTRATION

We built a Web interface to display *Chang Qing Yin*. Figure 3 shows a screenshot illustrating its main components.

Fixing Attributes. *Chang Qing Yin* provides 6 styles, 2 sentiments and 4 formats, and can automatically extract unique primary keyword and secondary keywords from plain texts.

Generating Poems. Next, *Chang Qing Yin* packages the tags into a tag-set. It is put into generation model to generate verses, and after automatic checking, poem is generated as shown in Figure 3.

Scoring and Polishing. If users are not satisfied with the generated poem, they are able to polish specified words. For example, system can automatically list 4 alternatives *morn*, *day*, *night* and *time* for users to replace *year*.

Comparison of Style Changing. If only modify the style from *Graceful* to *Bold*, system generates *Poem-Sty* as shown in Figure 4. It is clear that its style is totally different from *Poem-Original*. It describes a magnificent spring scenery and expresses the aroma of officialdom frustration in a forceful way. The first two verses depict a vast spring picture via *connects the sky* and *stretch thousands of miles*. *Cuckoos* often represents sorrow and tragedy in Chinese poetry, and the last verse reflects the feeling of depression due to unrealized ambition. Obviously, it has an uninhibited style, and expresses sadness when author's ambition is not repaid, meaning that system achieves good results in style control.

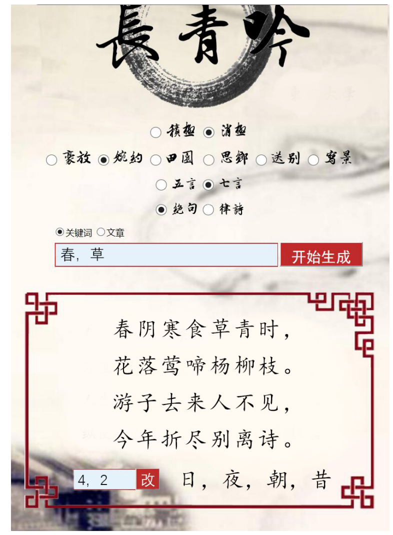

Figure 3: The Interface of CQY System.

Figure 4: *Poem-Sty* and *Poem-Senti*.

Comparison of Sentiment Changing. If only modify sentiment from *Negative* to *Positive*, system generates *Poem-Senti* as shown in Figure 4. It is evident that the style is the same as *Poem-Original*, which is graceful and subtle, but sentiments are quite different. It shows a bleak but cozy scene in spring, expressing author's comfort and coziness though boredom. Characters like *orioles* and *fragrant flowers* focus on the leisure spring time, reflecting a positive emotional atmosphere. Therefore, system achieves good results in sentiment control.

5 CONCLUSIONS

In summary, based on masking self-attention mechanism, we propose a style and sentiment controllable approach for Chinese poetry generation. First, SSPG packages specified attributes into a unique tag-set. Afterwards, it generates attribute-controllable poem automatically via generation model. Experimental results show that SSPG has better results compared with baselines, and also has prominent performance in controlling style and sentiment. In addition, our approach is applied to *Chang Qing Yin*, which allows users to polish verses, further increasing the diversity of poetry.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 5th International Conference on Learning Representations.*

[2] Huimin Chen, Xiaoyuan Yi, Maosong Sun, and et al. 2019. Sentiment-Controllable Chinese Poetry Generation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 4925–4931.

[3] Yang Cheng, Maosong Sun, Xiaoyuan Yi, and et al. 2018. Stylistic Chinese Poetry Generation via Unsupervised Style Disentanglement. In *Proceedings of the 56th Conference on Empirical Methods in Natural Language Processing*. 3960–3969.

[4] Liming Deng, Jie Wang, Hangming Liang, and et al. 2019. An Iterative Polishing Framework based on Quality Aware Masked Language Model for Chinese Poetry Generation. In *Proceedings of the 33rd Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*. 7643–7650.

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 16th Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[6] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 889–898.

[7] Zhipeng Guo, Xiaoyuan Yi, Maosong Sun, and et al. 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*. 25–30.

[8] Jinyi Hu and Maosong Sun. 2020. Generating Major Types of Chinese Classical Poetry in a Uniformed Framework. In *Proceedings of the 16th International Conference on Language Resources and Evaluation*. 4658-4663.

[9] Piji Li, Haisong Zhang, Xiaojiang Liu, and et al. 2020. Rigid Formats Controlled Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 742–751.

[10] Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. GPT-based Generation for Classical Chinese Poetry. *arXiv:1907.00151* (2019).

[11] Yusen Liu, Dayiheng Liu, and Jiancheng Lv. 2019. Deep Poetry: A Chinese Classical Poetry Generation System. In *Proceedings of the 33rd Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*. 13626–13627.

[12] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the 42nd Conference on Empirical Methods in Natural Language Processing*. 404–411.

[13] Zhe Wang, Wei He, Hua Wu, and et al. 2016. Chinese Poetry Generation with Planning based Neural Network. In *Proceedings of the 26th International Conference on Computational Linguistics*. 1051–1060.

[14] Rui Yan. 2016. I,poet: automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 2238–2244.

[15] Xiaoyuan Yi, Ruoyu Li, Cheng Yang, and et al. 2020. MixPoet: Diverse Poetry Generation via Learning Controllable Mixed Latent Space. In *Proceedings of the 34th Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*. 9450–9457.

[16] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and et al. 2018. Chinese poetry generation with a working memory mode. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4553–4559.

[17] Xingxing Zhang and Mirella Lapata. 2014. Chinese Poetry Generation with Recurrent Neural Networks. In *Proceedings of the 52nd Conference on Empirical Methods in Natural Language Processing*. 670–680.