Generating Chinese Classical Poems with RNN Encoder-Decoder

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun

State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing,China
yi-xy16@mails.tsinghua.edu.cn, liruoyu@6estates.com, sms@tsinghua.edu.cn

Abstract. We take the generation of Chinese classical poetry as a sequence-to-sequence learning problem, and investigate the suitability of recurrent neural network (RNN) for poetry generation task by various qualitative analyses. Then we build a novel system based on the RNN Encoder-Decoder structure to generate quatrains (*Jueju* in Chinese), with a keyword as input. Our system can learn semantic meaning within a single sentence, semantic relevance among sentences in a poem, and the use of structural, rhythmical and tonal patterns jointly, without utilizing any constraint templates. Experimental results show that our system outperforms other competitive systems.

Keywords: Chinese poetry generation · Neural network · Machine learning

1 Introduction

Chinese classical poetry is undoubtedly the largest and brightest pearl, if Chinese classical literature is compared to a crown. As a kind of literary form starting from the Pre-Qin Period, classical poetry stretches more than two thousand years, having a far-reaching influence on the development of Chinese history. Poets write poems to record important events, express feelings and make comments. There are different kinds of Chinese classical poetry, in which the quatrain with huge quantity and high quality must be considered as a quite important one. In the most famous anthology of classical poems, *Three Hundred of Tang Poems* [8], quatrains cover more than 25%, whose amount is the largest.

The quatrain is a kind of classical poetry with rules and forms which means that besides the necessary requirements on grammars and semantics for general poetry, quatrains must obey the rules of structure and tone. Figure 1 shows a quatrain generated by our system. A quatrain contains four sentences, each consists of seven or five characters. In Ancient Chinese, characters are divided into two tone categories, namely Ping (level tone) and Ze (oblique tone). Characters of particular tone must be in particular positions, which makes the poetry cadenced and full of rhythmic beauty. Meanwhile, in term of the vowels, characters

一声秋雁连天远，(*P*ZPPZ)

The twitter of a wild goose comes from the distant horizon.

万里归帆隔水<u>遥</u>。(*ZPPZZ**<u>P</u>**)

The homebound ships are still ten thousand miles away from the destination.

惆怅旧游零落处，(*Z*PPZZ)

I am so sad to be the place where I said goodbye to my travelling companions.

白头萧瑟满江<u>桥</u>。(*P*ZZP**<u>P</u>**)

There is nothing here, but a gloomy spectacle and the old me in the bridge.

Fig. 1. A 7-char quatrain generated by our system with the keyword "秋雁" (autumn wild goose) as input. The tone of each character is shown in parentheses. P, Z and * represent Ping tone, Ze tone and either respectively. Rhyming characters are underlined.

are divided into different rhyme categories. The last character of the first (optional), second and last sentence in a quatrain must belong to the same rhyme category, which enhances the coherence of poetry.

In this paper, we mainly focus on the automatic generation of quatrains. Nowadays, Deep Learning opens a new door to it, making computer no longer rely on prepared templates, and try to learn the composition method automatically from a large number of excellent poems. Poetry composition by machine is not only a beautiful wish. Based on the poem generation system, interesting applications can be developed, which can be used for education of Chinese classical poetry and the literary researches.

Different from the semantically similar pairs in machine translation tasks, the pair of two adjacent sentences in a quatrain is semantically relevant. We conduct various qualitative experiments to show RNN Encoder-Decoder structure can capture and learn the semantic relevance in Chinese classical poetry well. Based on these observations, consequently we take poem generation as a sequence-to-sequence learning problem, and use RNN Encoder-Decoder to learn the semantic meaning within a single sentence, the semantic relevance among sentences and the use of tonal patterns jointly. Furthermore, we use attention mechanism to capture character associations to improve the relevance between input and output sentences. Consisting of three independent sentences generation modules (word-to-sentence, sentence-to-sentence and context-to-sentence), our system can generate a quatrain with a user keyword. Both automatic and manual evaluations show our system outperforms other generation systems.

The rest of this paper is organized as follows. Section 2 introduces the related methods and systems. Section 3 gives the models in our system and analyses the suitability of them for poetry generation task by various qualitative experiments. Section 4 introduces our poetry generation system. Then Section 5 gives the evaluation experiments design and results. In section 6 we draw a conclusion and point out future work.

2 Related Work

The research about poetry generation started in 1960s, and has been a focus in recent decades. Manurung [7] proposed three criteria for automatically generated poetry: grammaticality (the generated sentences must obey grammar rules and be readable), meaningfulness (the sentences should express something related to the theme) and poeticness (generated poems must have poetic features, such as the rhythm, cadence and the special use of words).

The early methods are based on rules and templates. For example, ASPERA [3] uses the changes of accent as the templates to fill with words. Haiku generation system [11] expands the input queries to haiku sentences in term of rules extracted from the corpus. Such methods are mechanical, which match the requirements of grammaticality, but perform poorly on meaningfulness and poeticness.

One important approach is generating poems with evolutionary algorithms. The process of poetry generation is described as natural selection. Then through genetic heredity and variation, good results are selected by the evaluation functions [6, 7]. However, the methods depend on the quality of the evaluation functions which are hard to be designed well. Generally, the sentences perform better on meaningfulness but can hardly satisfy the poeticness.

Another approach is based on the methods for the generation of other kinds of texts. Yan et al. [12] generate poems with the method of automatic summarization. While SMT is first applied on the task of couplets generation by Jiang and Zhou [5]. They treat the generation of the couplets as a kind of machine translation tasks. He et al. [4] apply the method on quatrain generation, translating the input sentence into the next sentence. Sentences generated with such method is good at the relevance, but cannot obey the rules and forms.

With the cross field of Deep Learning and Natural Language Process becoming focused, neural network has been applied on poetry generation. Zhang and Lapata [13] compress all the previous information into a vector with RNN to produce the probability distribution of the next character to be generated.

Our work differs from the previous work mainly as follows. Firstly, we use RNN Encoder-Decoder as the basic structure of our system, compared with the method in [4]. Moreover, in He's system the rhythm is controlled externally and the results perform poorly on tonal patterns, while our system can learn all these things jointly. Secondly, compared with [13], our model is based on bi-directional RNN with gated units instead of the simple RNN. Besides, Zhang [13] compress all context information into a small vector, losing useful information in some degree. Thus they need two more translation models and another rhythm template to control semantic relevance and tones. The outputs of our system can obey these constraints naturally. Finally, we use attention mechanism to capture character associations and we also find reversing target sentences in training can improve performance.

3 Models and Qualitative Analyses

In machine translation tasks, the sentence pairs are semantically similar, from which the model learns the corresponding relations. While in Chinese classical quatrains, there is a close semantic relevance between two adjacent sentences. We use RNN Encoder-Decoder to learn the relevance which is then used to generate the next sentence given the preceding one. In this section, we introduce the models in our system and show why they are suitable for this semantic-relevant learning by several qualitative experiments.

3.1 Sentence Poetry Module (SPM)

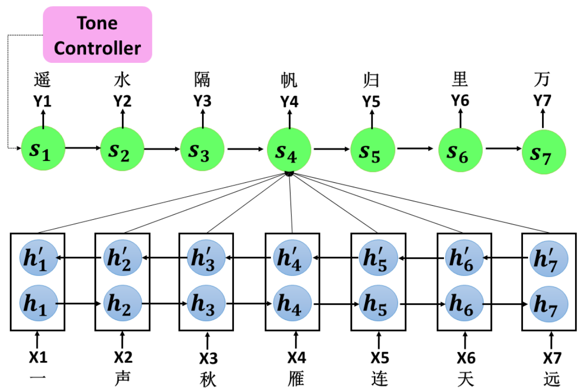

Fig. 2. An illustration of SPM. The tone controller is used to keep the last character of the second sentence level-toned.

We call the first model SPM (Sentence Poetry Module), which is used for sentence-to-sentence generation. Taking the first sentence as input, we use SPM to generate the second sentence in poem. As shown in Figure 2, we use bi-directional RNN with attention mechanism proposed in [1] to build SPM. Denote the four sentences in a quatrain are L_1, L_2, L_3, L_4, then three pairs are extracted, $< L_1, L_2 >$, $< L_2, L_3 >$ and $< L_3, L_4 >$, to train SPM.

Let us denote an input poetry sentence by $X = (x_1, x_2, ..., x_{T_x})$, and an output one by $Y = (y_1, y_2, ..., y_{T_y})$. $e(x_t)$ is the word-embedding of the t-th character x_t. h_t and $h_t^{'}$ represent the forward and backward hidden states in Encoder respectively.

In Encoder:

$$d_t = tanh(U[h_{t-1} \odot r_t] + We(x_t)) \tag{1}$$

$$u_t = \sigma(U_u h_{t-1} + W_u e(x_t)) \tag{2}$$

$$r_t = \sigma(U_r h_{t-1} + W_r e(x_t)) \tag{3}$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot d_t \tag{4}$$

$$g_t = [h_t; h_t^{'}] \tag{5}$$

(1)-(5) are formulas for the computation of forward hidden states. The computation of backward hidden states is similar. $\odot$ is element-wise multiplication. g_t is the final hidden state of t-th character in Encoder. r_t and u_t are the reset gate and update gate respectively introduced in [2]. The formulas in decoder are similar. The difference is that a context vector c_t of attention mechanism is used to calculate the hidden state s_t of t-th character in Decoder. c_t is computed as:

$$c_t = \sum_{i=1}^{T_x} \alpha_{t,i} g_i \tag{6}$$

$$\alpha_{t,i} = \frac{exp(v_{t,i})}{\sum_{j=1}^{T_x} exp(v_{t,j})} \tag{7}$$

$$v_{t,i} = v_a^T tanh(W_a s_{t-1} + U_a g_i) \tag{8}$$

3.2 Target Sentences Reversing

As shown in Figure 2, when training SPM we reverse the target sentences for two reasons. First, the final character of second sentence must be level-tone. However, if the final character of input sentence is level-tone, SPM can't determine the tone of the final character of output sentence because of some special tonal patterns in training pairs. Thus we add a tone controller into SPM. Obviously, this control will do harm to the semantic meaning of the outputs. Therefore we reverse the target sentences in training so that SPM can generate the tail character first, which will decrease the damage on meaningfulness as possible.

Furthermore, we find reversing target sentences can improve the performance. In [9], Sutskever et al. find reversing source sentences can improve the LSTM’s performance on machine translation. But in section 5, by quantitative evaluation, we show that reversing source sentences makes little improvement in our task. We think this is because we use a bi-directional Encoder, which handles the problem Sutskever points out in some degree.

Besides, we think the improvement of reversing target sentences is because of the attributive structure in Chinese classical poetry. There are many words with the structure *attributive + central word* in Chinese poetry. For example, “青草” (green grass), the character “青” (green) is attributive and the character “草” (grass) is central word. In normal generation order, “青” will be generated earlier than “草”. But there are many other characters can be central word of “青”,

such as “青山” (green mountain), ” 青云”(green cloud), “青烟” (green smoke), which increases the uncertainty in the generation of “草”. Whereas reversing target sentence can reduce this uncertainty since the attributive of “草” is often “青” in Chinese classical poetry.

3.3 Qualitative Analyses of SPM

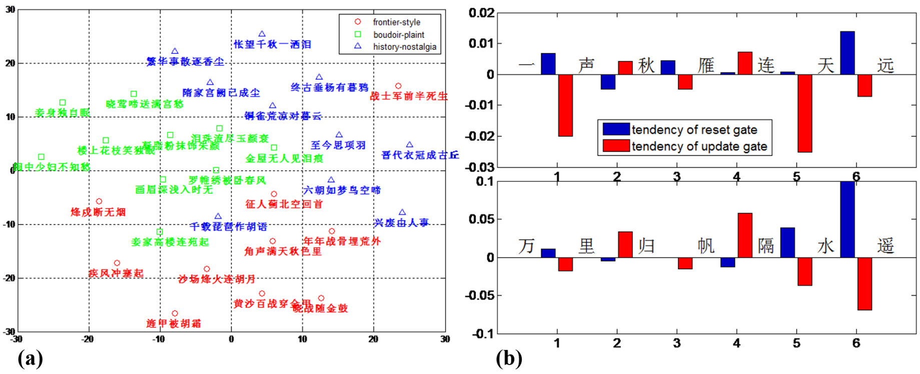

Fig. 3. (a) 2-D visualization of the learned poem sentences representations. The circle, square and triangle represent frontier-style, boudoir-plaint and history-nostalgia poetry sentences respectively. (b) An example of word boundaries recognition by gated units. Between every two adjacent characters, the first and the second bars show the reset and update tendencies respectively. The upper plot is the input sentence and the lower one is the output one.

Poem Sentence Representations We used the average of g_t in formula (5) as sentence representation. We selected three types of classical poetry: frontier-style poetry (poetry about the wars), boudoir-plaint poetry (poetry about women's sadness) and history-nostalgia poetry (poetry about history). For each type, we obtained ten sentences and used Barnes-Hut-SNE [10] to map their representations into two-dimensional space. As shown in Figure 3 (a), sentences with the same type gather together, which shows these representations can capture the semantic meanings of poetry sentences well.

Gated Units in Word Boundary Recognition The training and generation are based on characters (because there is no effective word segmentation algorithm for Chinese classical poetry) without any explicit word boundaries in the sequences, but we find gated units can recognize the word boundaries roughly. As in formula (2) and formula (3), when r_t tends to be zero and u_t tends to be one, the gated units tend to use current input to update the hidden state, whereas gated units tend to keep previous hidden states. We used the average value of every elements in r_t as the reset value of t-th character. Along the direction of hidden states propagation, we calculated the difference between reset values of two adjacent characters to get the reset tendency. Similarly, we got the update tendency.

As shown in Figure 3 (b), higher reset tendency and lower update tendency mean that the two characters tend to be an entirety (a whole word). Whereas they tend to be separated. In sentence “一声秋雁连天远”, “一声”and “秋雁” are both words. We can see the reset tendency and update tendency reflect the word boundaries roughly. Furthermore, the tendency of gated units in Decoder is similar with that in Encoder. This nature makes the vector representations contain information of whole words and makes output sentences keep the similar structures as input ones.

Attention Mechanism in Capturing Associations Different from word alignments in translation task, attention mechanism can capture the implicit associations between two characters. Figure 4 shows the visualizations of $\alpha_{t,i}$ in formula (8).

The top two plots show the associations between input and output sentences generated by a SPM trained with reversed target sentences. The bottom two plots are the results of SPM trained with normal target sentences.

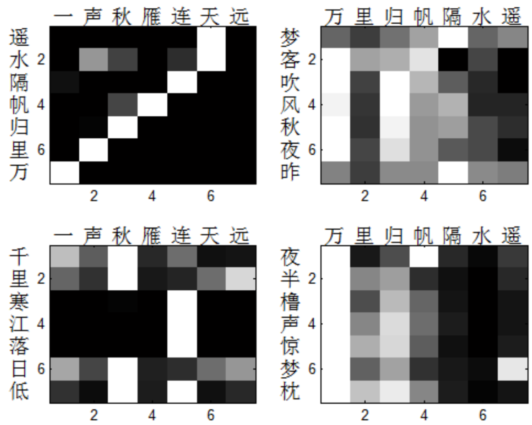

Fig. 4. Examples of attention visualizations. Each pixel shows the association(0: black, 1: white) between characters in input sentences (horizontal) and in output sentences (vertical). Outputs on the top two plots are reversed for the reasons described in section 3.2.

In the top left plot, each character in output sentence focuses on the semantic relevant character in input sentence, such as “水”(sea) and “天”(sky), “万”(ten thousand) and “一”(one), “帆”(sail) and “雁”(wild goose).

Also, in the top right plot, besides the first character, output sentence mainly focuses on the second character “归” (return), since the input is about home-bound ships and the output is about the travellers.

Besides, we get a great improvement by reversing the target sentences in training. There are no obvious associations in the bottom two plots compared with results in the top two plots, which shows reversing target sentences improves attention performance and thus leads to better generation performance. Detailed quantitative evaluation results please refer to section 5.

3.4 Context Poetry Module (CPM)

To utilize more context information, we build another Encoder-Decoder called CPM (Context Poetry Module). The structure of CPM is similar with that of SPM. The difference is that we concatenate two adjacent sentences in a quatrain as a long input sequence, and use the next sentence as the target sequence in training. We extract two pairs from each quatrain, $< L1L2, L3 >, < L2L3, L4 >$ to train CPM. By this means, the model can utilize information of previous two sentences when generating current sentence.

The final characters of the second and the fourth sentence must rhyme and the final character of the third sentence must not. When generating the fourth sentence, SPM can't determine the rhyme category. By taking the second sentence into consideration, CPM can capture the rhyme pattern. Thus we use CPM to generate the third and the fourth sentences. Zhang [13] utilizes context by compressing all previous sentences into a 200-dimensional vector, which causes some loss to semantic information. Whereas our method can save all information. When generating current sentence, the model can learn to focus on important characters with attention, rather than use all context indiscriminately, which will improve semantic relevance between the inputs and the outputs. We don't concatenate more previous sentences for two reasons. Firstly, too long sequences result in low performance. Secondly, relevance between the fourth sentence and the first sentence is relatively weak, there is no need to make the system more complicated.

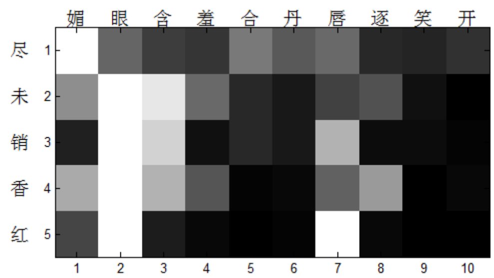

Fig. 5. An attention visualization example of CPM. The input are two concatenate sentences and each consists of five characters. The output is reversed.

Figure 5 shows an attention visualization of CPM. As we can see, because the input sentence is a description of a beautiful woman, attention mechanism focuses on two characters, “眼” (eyes) and “唇” (lips). Though there is a color word “丹” (red) in the input sentence, attention mechanism chooses to focus on “眼” and “唇” instead of “丹” for generating the character “红” (also means red color) since in Chinese classical poetry, “红” is often used to describe the beauty of women. Compared with the simple alignments of words with same semantic meanings in translation task, attention mechanism can learn the associations and helps the model to focus on the most relevant information instead of all context, which results in a stronger relevance between input and output sentences.

3.5 Word Poetry Module (WPM)

A big shortcoming of SMT-based methods is that they need another model to generate the first sentence. For example, He [4] expands user keywords, then uses constraint templates and a language model to search for a sentence.

For RNN Encoder-Decoder, words and sentences will be mapped into the same vector space. Since our system is based on characters, words can be considered as short sequences. Ideally, SPM will generate a relevant sentence taking a word as input. But the training pairs are all long sequences, it won't work well when the input is a short word.

Therefore, we train the third Encoder-Decoder, called Word Poetry Module (WPM). In detail, we pre-trained a SPM, then extracted some <word, sentence> pairs (e.g. <' 明月',' 床前明月光'>) and trained the SPM with these pairs for several epoches , improving the ability of generating long sequences with short sequences.

4 Poetry Generation System

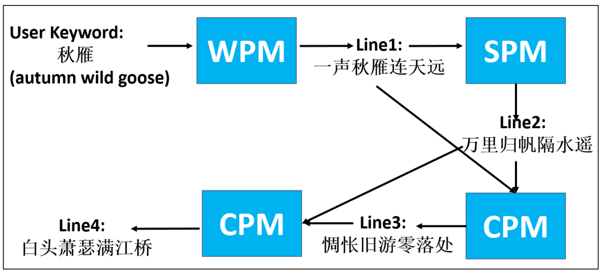

Fig. 6. An illustration of poem generation process with the keyword “秋雁” as input.

Based on observations above, we use RNN Encoder-Decoder to learn the relevance which is then used to generate the next sentence given the previous one. To utilize context information in different levels, we use three generation modules, Word Poetry Module (WPM), Sentence Poetry Module (SPM) and Context Poetry Module (CPM) to generate a whole quatrain.

As illustrated in Figure 6, the user inputs a keyword to show the main content and emotion the poem should convey. Firstly, WPM generates the first sentence relevant to the keyword. Then SPM takes the first sentence as input and generates the relevant second sentence. CPM generates the third sentence with the first two sentences as input. Finally, CPM takes the second and third sentences as input and generates the last sentence.

We train the model and generate poems based on Chinese characters, since there are no effective segmentation tools for Ancient Chinese. Fortunately, the length of Chinese classical poem sentences is fixed five or seven characters. And most words in Chinese classical poetry consist of one or two Chinese characters. Therefore, this method is feasible.

5 Experiments

5.1 Data and Settings

Our corpus contains 398,391 poems from Tang Dynasty to the contemporary. We extracted three pairs from each quatrain to train SPM, and extracted two pairs from each quatrain to train CPM. For training WPM, we selected 3000 words, and for each word we got 150 sentences which the word appears in. Finally we obtained 450,000 word-to-sentence pairs (half are 5-char and the other half are 7-char). We built our system based on GroundHog.[1]

5.2 Evaluation Design

Models	**sentence2**		**sentence3**		**sentence4**		**Average**	
	5-char	7-char	5-char	7-char	5-char	7-char	5-char	7-char
SMT	0.526	0.406	0.262	0.214	0.432	0.314	0.407	0.311
SPM0	0.773	0.956	0.478	0.728	0.831	1.450	0.694	1.045
SPM0 + src reverse	0.739	1.048	0.671	1.049	0.876	1.453	0.762	1.183
SPM0 + trg reverse	**1.126**	**1.900**	**1.251**	**1.441**	**1.387**	**2.306**	**1.255**	**1.882**

Table 1. BLEU-2 scores on quatrains. SPM0 is the structure without reversing source or target sentences. And src reverse means source sentence reversing; trg reverse means target sentence reversing.

BLEU Score Evaluation. Referring to He [4] and Zhang [13], we used BLEU-2 score to evaluate our model automatically. Since most words in Chinese classical poetry consist of one or two characters, BLEU-2 is effective. It's hard

[1] `https://github.com/lisa-groundhog/GroundHog.`

Models	Fluency		Coherence		Meaningfulness		Poeticness		Entirety	
	5-char	7-char	5-char	7-char	5-char	7-char	5-char	7-char	5-char	7-char
SMT	1.65	1.56	1.52	1.48	1.42	1.33	1.69	1.56	1.48	1.42
DX	2.53	2.33	2.19	1.96	2.31	2.00	2.52	2.31	2.29	2.08
PG	3.75	3.92	3.42	3.48	3.50	3.50	3.50	3.67	3.60	3.67
Human	3.92	3.96	3.81	4.10	4.08	4.13	3.75	4.02	3.96	4.21

Table 2. Human evaluation results. The Kappa coefficient of the two groups' scores is 0.62.

to obtain human-authored references for poetry so we used the method in [4] to extract references automatically. We first selected 4,400 quatrains from corpus (2,200 of them are 5-char and other 2,200 are 7-char) and extracted 20 references for each sentence in a quatrain (except the first sentence). Then the left quatrains are used as training set. We compared our SPM (with different strategies) with the system in [4].

Human Evaluation. Since poetry is a kind of creative text, human evaluation is necessary. Referring to the three criteria in [7], we designed five criteria: Fluency (are the sentences fluent and well-formed?), Coherence (does the quatrain has consistent topic across four sentences?), Meaningfulness (does the poem convey some certain messages?), Poeticness (does the poem have poetic features?), Entirety (the reader's general impression on the poem). Each criterion was scored from 0 to 5.

We evaluated four systems. **PG**, our system. **SMT**, He's system [4]. [2] **DX**, the DaoXiang Poem Creator.[3] DX system is the pioneer for Chinese classical poetry generation. It has been developed for 15 years and been used over one hundred million times. **Human**, the poems of famous ancients poets containing the given keywords.

We selected 24 typical keywords and generated two quatrains (5-char and 7-char) for each keyword using the four systems. By this means, we obtained 192 quatrain (24*4*2) in total. We invited 16 experts on Chinese classical poetry to evaluate these quatrains. Each expert evaluated 24 quatrains. The 16 experts were divided into two groups and each group completed the assessments of the 192 poems. Thus we got two scores for each quatrain and used the average score.

5.3 Evaluation Results

Table 1 shows the BLEU-2 scores. Because DX system generates poetry as a whole, we only compared our system with SMT on single sentence generation task. In Chinese classical poetry, the relevance between two sentences in a pair is related to the position. Therefore He et al. [4] use pairs in different positions to train corresponding position-sensitive models. Because of the limited training data, we used pairs in all positions to train SPM. Even so, we got much higher

[2] `http://duilian.msra.cn/jueju/`.

[3] `http://www.poeming.com/web/index.htm`.

BLEU scores than SMT in all positions. Moreover, 95% of the sentences generated by our system obey tonal constraints, but only 31% of SMT's outputs obey the constraints.

We can also see that reversing source sentences made a little change while reversing target sentences led to great improvement.

As shown in Table 2, our system got higher scores than other systems, expect the Human. For SMT and DX, the scores of 7-char poems are lower than that of 5-char poems in all criteria (both in Human evaluation and BLEU evaluation) because the composition of 7-char quatrains is more difficult. But the poems generated by PG got higher scores on 7-char poems, benefiting from gated units and attention mechanism. Scores of PG is closed to that of Human, though there is still a gap.

We also asked the experts to select a best sentence from the evaluated quatrains. 37% of the selected 5-char sentences are our system's and 42% are poets'. And 45% of the selected 7-char sentences are our system's, 45% are poets'. This indicates that our system has little difference with poets in generating meaningful sentences at least.

6 Conclusion and Future Work

梦断中秋月，
When I woke up from the dream suddenly,
I saw the mid-autumn moon.
天寒咽暮蝉。
It was too cold for the cicadas to sing in the evening.
不堪送归客，
I couldn't bear the pain of seeing my friends off.
寂寞对床眠。
The only thing I could do was trying to fall asleep with loneliness.

谁怜两地中秋月，
Who will feel sympathy for the separated us?
Only the autumn moon will.
独照西窗一夜凉。
The moonlight through the window is so lonely on the cold night.
行到故园应怅望，
Maybe your are overlooking and trying to find where I am in the distance.
哀词遗恨满潇湘。
While I can only put my missing and sadness in my poems, and let the melancholy fill the Xiao River and the Xiang River.

Fig. 7. Another two poems generated by our system with keyword "秋月" (autumn moon).

In this paper, we take the generation of poem sentences as a sequence-to-sequence learning problem, and build a novel system to generate quatrains based on RNN Encoder-Decoder. Compared with other methods, our system can jointly learn semantic meanings, semantic relevance, and the use of rhythmical and tonal patterns. Both automatic and human evaluations show that our system outperforms other systems.

We show that RNN Encoder-Decoder is also suitable for the learning tasks on semantically relevant sequences. The attention mechanism can capture character associations, and gated units can recognize word boundaries roughly. Moreover, reversing target sentences in training will lead to better performance.

There are lots to do for our system in the future. We will improve our system to generate other types of Chinese poetry, such as Songci and Yuefu. We also hope our work could be helpful to other related work, such as the building of poetry retrieval system and literature researches.

References

1. Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations*, San Diego, CA, 2015.
2. K. Cho, B. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar, 2014.
3. Pablo Gervás. *An Expert System for the Composition of Formal Spanish Poetry*, pages 181–188. Springer London, 2001.
4. Jing He, Ming Zhou, and Long Jiang. Generating chinese classical poems with statistical machine translation models. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1650–1656, Toronto, Canada, 2012.
5. Long Jiang and Ming Zhou. Generating chinese couplets using a statistical mt approach. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 377–384, Manchester, UK, 2008.
6. Robert P. Levy. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*, 2001.
7. Hisar Maruli Manurung. *An evolutionary algorithm approach to poetry generation*. PhD thesis, University of Edinburgh, 2003.
8. Zhu Sun. *Three hundred Poems of the Tang Dynasty(《唐诗三百首》)*. 1764.
9. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 4:3104–3112, 2014.
10. Laurens van der Maaten. Barnes-hut-sne. In *Proceedings of the First International Conference on Learning Representations(ICLR 2013)*, Scottsdale, Arizona, 2013.
11. Xiaofeng Wu, Naoko Tosa, and Ryohei Nakatsu. New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system. In *Proceedings of the 8th International Conference on Entertainment Computing*, pages 191–196, Paris, France, 2009.
12. Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. I, poet:automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2197–2203, Beijing, China, 2013.
13. Xingxing Zhang and Mirella Lapata. 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, 2014.