CSLT TECHNICAL REPORT-20190001 [Friday 8th November, 2019]

A Rhythm Model for Chinese Poetry Generation

Yibo Liu*
and Dong Wang

*Correspondence: liuyibo@cslt.org
Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

Poetry is a treasure in human culture for its beauty and creativity. Automatic Chinese poetry is a challenging task in natural language processing which represents computer intelligence and has attracted researchers' attention for many years. Attributed to the development in neural network, great improvements have been achieved in recent years. Previous methods mainly focused on the semantic aspects of the content while few paid attention to the modeling of rhythm. The common way to maintain rhythm is to apply rules on rhyme and tone to rule out unsatisfying words during decoding. However, rule-based decoding not only sacrifices the semantic performances for poeticness, but also requires human design of the rhymic and tonal rules which is lack of flexibility. In this paper, we proposed a model for Chinese poetry rhythm, which can learn the tonal patterns and generate poems with poeticness. This model can balance between semantic and poetic aspects. In addition, the poeticness of the generated poems can be controlled manually.

Keywords: poetry generation; rhythm model

1 Introduction

Poetry is one of the most fascinating form of human language and illustration of human intelligence, and it is a popular form of literature in all cultures and languages. Chinese ancient poetry is important for its aesthetic value and its social functions.

Pragmatically, quatrain is one of the most popular class of Chinese poetry. According to the length in each line, quatrain can be classifies as *Qiyan* (7 words per line) and *Wuyan* (5 words per line). A quatrain is fascinating for its fluctuation in rhythm. Concretely, we describe the rhythm in two aspects: rhyme and tone. The rhymic pattern requires that the last character of some lines should be in the same rhyme. The tone rule describes the tone category that the characters should possess at specific positions. More specifically, a character may belong to either Ping tone (level tone) or Ze tone (downwards tone). The tone of a character at some particular positions depend on each other, leading to tonal patterns. Essentially, tonal patterns reflect the musical esthetics of poems.

We focus on automatic Chinese poetry generation in this paper. In recent years, Chinese poetry generation has received much attention from researches. Traditionally, rule and template based methods [1], statistical methods [2] were widely used. More recently, deep neural methods [3] are proposed and succeed in generating fluent and coherent poems, and many proposals [4] [5] [6] [7] in model structure have been made to promote the novelty and creativity of the poem.

In spite of the promising results made by aforementioned models, there are still space for promotion. Most of existing approaches focus on modeling the semantic content of characters, and rhythm regularizations are imposed simply by rule-based character selection during decoding, leading to sacrifice in semantic coherence and fluency. To be more specific, as it did in [8] and [6], several tone templates are set manually according to knowledge in Chinese ancient poetry. For example, a tone template for a *Qiyan* quatrain looks like *zzppzzp ppzzzpp ppzzppz zzppzzp*, in which *p* and *z* denote *Ping* tone and *Ze* tone respectively. When generating a character at a particular position, the tone of each candidate is compared with the template one by one according to their probability. If the character does not meet the requirement of the template, the next candidate with the highest probability will be considered, until rhyme is processed similarly. We call this method hard constraint.

Hard constraint may lead to suboptimal generation. First, it lacks flexibility to use tonal templates. Although some ancient poems strictly follow the templates proposed by the ancient poetry guideline *Ping Shui Yun*, it is too doctrinaire, and our human poem dataset shows that no more than 20% poems follow the templates exactly. Second, hard constraints give tone and rhyme the first priority, but in fact, many high-level human poets sacrifice tone rule for better semantic expression. This inspired us to find an equilibrium between semantic meaning and rhythm rules, allowing a few exceptions in rhythm rules when a candidate is highly advantageous in semantic expression, hence improving the overall performance of the generation.

In this paper, we propose a soft constraint rhythm model to describe the rhythm requirements in Chinese ancient poems. The model gives a prediction on tone at each decoding step according to a learned tonal pattern. The predictions are added as a score to the decoder output, hence the candidate word probabilities are regularized by the prediction scores. Different from hard constraint, this method allows the inconsistence on rhythm if the semantic probability is high, hence it is called soft constraint. The essence is that we design a tonal coherence matrix to describe the tonal dependencies between characters. This matrix is learned from data, hence describes the *probabilistic patterns* used by human poets, rather than pre-defined templates. Our model has two advantages: (1) the tonal pattern is learned from data, rather than manually designed; (2) the suggestion on the tone is given in the form of rhythm scores that can be combined to the semantic outputs, allowing the model to consider both semantic coherence and rhythm accordance. Because of (1), our model is universal to many other literature genres, like Chinese Song Iambics, which has too many templates to design manually. Due to (2), our model can assign different weights to semantic and rhythm aspects according to the preference of users, and the hard constraint is actually a special case of our model. What's more, our model can also be used in any state-of-the-art architectures to further promote their overall performances.

Our contributions can be summarized as the following:

- First proposed a rhythm model that offers soft constrains on rhythm patterns.
- The rhythm model reveals inner rhythm patterns in poems (to be described in Section IV).
- Our model achieves success on promoting the overall performance of state-of-the-art models.

2 Methods

2.1 Basic Sequence-to-sequence model

The most popular architecture for poem generation is the sequence-to-sequence framework equipped with the attention mechanism, originally proposed for neural machine translation. This model can be described as the following. We use $X = (x_1, x_2, ..., x_{T_x})$ as the input sequence for the encoder, $Y = (y_1, y_2, ..., y_{T_y})$ as the output sequence of the decoder. The encoder is based on a bidirectional RNN with GRU nodes. The input of the encoder is the sequence of the embeddings of the characters of the topic words. h_t represents the hidden state of the encoder at step t, and it is updated by $h_t = GRU_{enc}(x_t, h_{t-1})$. The outputs of the RNNs of the forward and backward directions are concatenated, i.e., $h_t = [\overleftarrow{h_t}; \overrightarrow{h_t}]$.

The decoder is a single directional GRU. At each decoding step, the hidden state s_t is updated by $s_t = GRU_{dec}(y_{t-1}, s_{t-1})$. The decoder accepts this hidden state and a context vector c_i computed by the output of encoder. The context vector can be described as the following:

$$e_{ij} = score(s_i, h_j),$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})},$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

The output of the decoder at the t-th step can be calculated by:

$$o_t = W_s tanh(W_c[c_t; s_t]),$$

$$p(y_t | y_{1:t-1}, x) = softmax(o_t).$$

In the implementation, we use key words as the input sequence and the whole poem as the output. The vocabulary involves high-frequency characters. That is, key words are firstly represented as a character sequence which are further converted into a sequence of embeddings. The decoder generates one character at each time step. Note that this basic sequence-to-sequence model focuses on semantic coherence and does not involve any rhythm knowledge.

2.2 Rhythm model

We embed a tone by a 1-dimensional vector, with $[-1]$ indicating *Ze* and $[1]$ indicating *Ping*. A rhyme is embedded by a 16-dimensional one-hot vector, since all the rhymes can be classified into 16 classes. We suppose that the tone of a position is determined by the tone of all the previously decoded characters, and the rhyme of the last character in a line is determined by the rhyme of all the last characters in previously decoded lines. So we define $S = (s_1, s_2, ..., s_{t-1})$ as tone embedding of the decoded sequences.

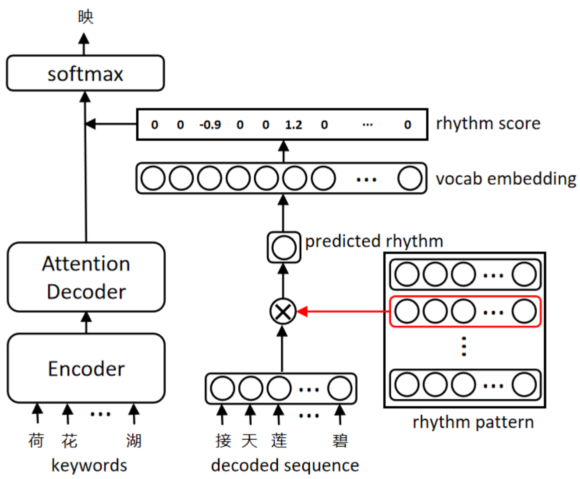

Figure 1 Rhythm Model

Then we define a tonal pattern matrix M_{tone}. The dimension of M_{tone} is 28*28 (taking *Qiyan* as an example, which contains 28 characters in a poem with 7 characters in each line), the element m_{ij} is the contribution of the i-th decoded character when generating the j-th character in the poem. If m_{ij} is a positive/negative value, it means the j-th character is likely to have the same/opposite tone as the i-th character.

At each time step, it gives a tone prediction. The tone prediction is calculated by the linear combination of tone embedding of all decoded characters. The weights of linear combination comes from a row in tonal pattern matrix.

$$score_{tone}(s_t|s_{1:t-1}) = \sum_{j=0}^{j=t-1} m_{tj}s_j, m_{tj} \in M_{tone}$$

The positive/negative value of $score_{tone}$ indicates the likelihood of *Ping*/*Ze* tone.

The scores are computed for all characters in the vocabulary, and the results are used as regularization terms added to the basic sequence-to-sequence model output that focuses on the semantic coherence. The final output can be written by:

$$o'(y_t|y_{1:t-1}, x) = w_1 o(y_t|y_{1:t-1}, x) + \\ w_2 score_{tone}(s_t|s_{1:t-1}) V_{tone}$$

$$p(y_t|y_{1:t-1}, x) = softmax(o'),$$

where V_{tone} is the embedding matrix of tone, and w_1 and w_2 are coefficients that balance the semantic and rhythmic parts.

The structure of the whole model is shown as Fig. 1.

3 Experiment

In this section we introduce our dataset used for experiments, experiment settings, baseline models as well as the evaluation metrics.

3.1 Dataset and configuration

The dataset used in this experiment contains 1031,000 different Chinese poems. Among which 370,000 are ancient poems while the rest are poems written by modern poets. All of the poems are quatrains, either *Qiyan* (7 characters per line) or *Wuyan* (5 characters per line). The ratio of the size of the training set, validation set and test set is 8:1:1.

During training, the keywords are extracted from the training poems, with one keyword extracted from each line in a random way. Using this method, a single poem can generate several keywords-poem pairs. The experimental results demonstrated this random sampling approach can greatly improved the overall performance in poem generation.

The character embeddings are pre-trained on a large out-of-domain data, where the dimension is set to be 200. The number of hidden units of the GRU layer is set to be 500, for both the encoder and decoder, and the batch size is set to 80. AdaDelta is used as the optimizer, with the same settings as in [9].

3.2 Evaluation metrics

Rhythmic performance. We propose a compliance score to evaluate rhythmic performance. Compliance score is defined as the tone accuracy and rhyme accuracy according to the rhythm rule. We use 2 templates as the gold standard according to the ancient Chinese poem guideline *Ping Shui Yun*: *0p0z0p0 0z0p0z0 0z0p0z0 0p0z0p0* and *0z0p0z0 0p0z0p0 0p0z0p0 0z0p0z0*, in which *p/z* denotes *Ping/Ze* tone, *0* denotes either *Ping* tone or *Ze* tone. Tone accuracy is defined as the percentage of generated poems that meet the gold standard. The poeticness is defined formally as follows:

$$Compliance = Accuracy_{tone}$$

Semantic performance. Previous research [5] [7] [10] used BLEU score as the metric for objective evaluation. However, this metric, as [11] pointed out, is deviated from human evaluation. According to [3], we adopt the following subjective metrics: fluency(if the sentences make sense in syntex), coherence(if the theme remain the same among different sentences), meaningfulness(if the sentences convey much information and contain few function words) and poeticness(if the poem has artistic beauties). Each score ranges from 1 to 5. We invited experts to rate the 100 poems each model for comparision.

Besides, language model (LM) score serves as an supplement of the human evaluation. This is inspired by [11], in which language model is used as a reinforcement learning reward to evaluate fluency. Formally, given a line of a poem L_i, the n-gram language model probability $P_{lm}(L_i)$ indicates the likeliness of the presence of the line in corpus that is used to train the language model. The higher the probability, the more well-formed the line is. The LM score of a poem is defined as the average over all lines:

$$LM = \frac{1}{n}\sum_{i=1}^{n} P_{lm}(L_i).$$

Table 1 Poetry generation performance

	Rhythmic	Semantic					
	Compliance	*Fluency*	*Coherence*	*Meaningfulness*	*Poeticness*	*Overall*	$LM(\times 10^{-10})$
Basic	0.306	2.2	2.5	1.6	2.1	2.1	3.81
Basic+rule	1	3.6	3.1	2.5	2.9	3.023	1.55
Soft	**0.408**	3.5	3.6	2.9	3.0	**3.25**	3.54
Soft+rule	1	3.0	3.5	2.6	3.0	3.025	3.00
Jiuge	0.549	-	-	-	-	-	-

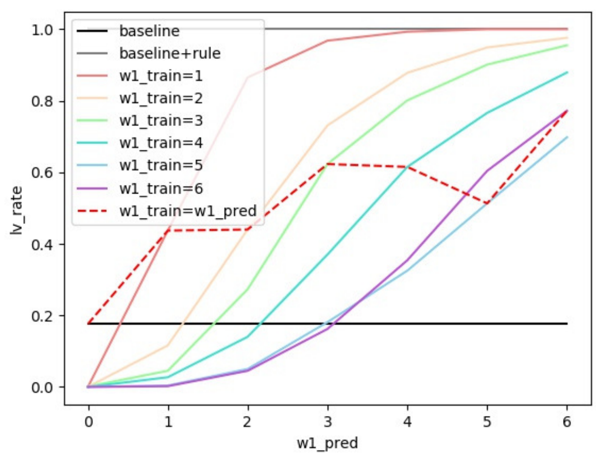

Figure 2 Tone Accuracy

3.3 Baseline systems

We compare the proposed rhythm-augmented model (**Soft**) with a number of baselines: **GT**, the ground truth, i.e., the human written poems; **Basic**, the basic sequence-to-sequence model with attention described in section II.A; **Basic + rule**, adding rhythm rules to the basic model; **Soft + rule**, rhythm-augmented model applied with hard constrain. **Jiuge**, a state-of-the-art model in Chinese poetry generation.

4 Results and analysis

As we can see, Soft performes better than all other models, including the state-of-the-art model Jiuge.

4.1 The balance between rhythmic and semantic performance

As demonstrated in Tab. 4, the tone accuracy of Soft are both higher than Basic. The result proves that the rhythm model is effective to improve tonal compliance comparing to basic model when no explicit rhythm rules are applied and has successfully learned tonal patterns from the data. Meanwhile, the semantic performances remain good.

We did further experiments to explore how the rhythm restriction can be controlled manually. We changed the coefficients $w_2 : w_1$ (the sum of w_1 and w_2 remains 1) to see the changes in tone rhythmic and semantic performances. To implement

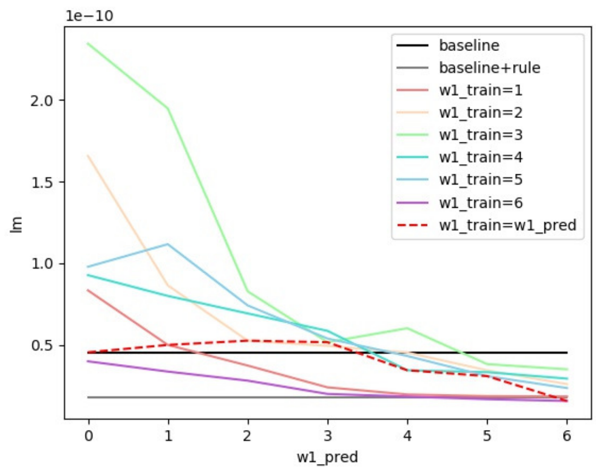

Figure 3 LM score

large scale experiments, here we use a small dataset that is extracted from the original dataset, consisting 10,000 poems. To simplify the evaluation process, we only use LM score as the evaluation of semantic performance.

First, we compare Soft and Basic to show how tone is controllable by the coefficient w_2. Using different w_2 in training and generation can change the tone accuracy. As shown in Fig. 2 and Fig. 3, when w_2 for training matches w_2 for generation, when increasing w_2, tone accuracy tends to increase and LM score tends to decrease, as shown by the red dot line. For all w_2, tone accuracy is higher than Basic model. This shows the effect of rhythmic model. While w_2 for training and for generation does not match, tone accuracy increases from 0 to 1, and LM score decreases when gradually increasing w_2 in generation.

It is interesting that for all w_2 in training, when $w_2 = 0$ in generation, tone accuracy is 0 and LM score is relatively high. From this result, it is reasonable to claim that the sequence-to-sequence model and the rhythmic model are able to work on different tasks seperately, and this seperation helps to promote on both tasks. This reveals the essence of our rhythm model, multi-task learning. Therfore, this phenomenon can be explained that when $w_2 = 0$ in generation, only the basic sequence-to-sequence model is working, and it has good performance on learning semantic pattern, ignoring rhythmic pattern. So LM score is higher than (or comparable to) Basic model, while tone accuracy is almost 0. Attributed to the multi-task learning, it is able to find some specific configurations that has better performances than Basic both semantically and rhythmically.

In addition, we compare Soft and Soft + rule to demonstrate how Soft prevents rule harm the performance. A generation example is shown as Fig. 4. In the second line, the word *Bi (close)* is changed to *Chan (dhyana)* when applying Basic + RM. Although it breaks the tonal rule at this position, it has some connotation of Budhism, which lively described the scene. While the word *Bi* does not have the same effect aesthetically. It is reasonable to select the word *Chan* in Soft. The probability $o(y_t|y_{1:t-1})$ of *Chan* calculated from sequence-to-sequence model is relatively high, showing that *Chan* is an appropriate word at this position. The Soft + rule model rules out this candidate because of its non-compliance on tone. However, Soft only gives it a punishment. Since the probability is high, the punishment is balanced out. This example shows how rhythm model keeps a balance between semantic and rhythmic performances.

Keywords: 枯藤，僧寺，霜，雨后 Dry vine, temple, frost, after rain	
Basic + RM:	**Basic + rule:**
松山霜月挂岩苔，*0 p 0 z 0 p 0*	松山霜月挂岩苔，*0 p 0 z 0 p 0*
Moonlight is above the pine tree mountain and mossy rock,	Moonlight is above the pine tree mountain and mossy rock,
闲寺僧门住禅来。*0 z 0 p 0 p 0*	闲寺僧门住闭来。*0 z 0 p 0 z 0*
Dhyana lives in the leisure temple.	The temple is closed at the leisure time.
雪打寒风凝露色，*0 z 0 p 0 z 0*	寒雨清风吹雪色，*0 z 0 p 0 z 0*
Snow knocks, wind blows, dew is frozen,	Cool rain, cold wind, snow is blowing,
雨声落叶几时开。*0 p 0 z 0 p 0*	山林落叶几时开。*0 p 0 z 0 p 0*
When will the leaves fall in the rain?	When will the leaves fall in the forest?

Figure 4 Poem example

4.2 Tonal patterns for poetry

The tonal pattern is visualized as Fig. 5. The color indicates the dependencies between different position when predicting tone. In Fig. 5, there are 28 small pictures, each for a single position. The i-th picture shows the tone dependency when predicting the i-th character. This figure shows some interesting patterns. For example, the fourth character in each line largely depends on the second character in the same line, and the sixth character mainly depends on the second character and the fourth character in the same line and all the previous lines. The last character in the fourth line depends on the last character of the second line. Besides, the first, third, fifth characters show more ambiguous dependency than the second, fourth, sixth character in each line. These features are consistent with the rhythm rule that only the second, fourth and sixth characters need to be regularized in terms of tones.

In summary, the result demonstrated that the pattern matrix is a good representation of rhythm patterns in Chinese poetry generation.

5 Conclusion

In this paper, we proposed a rhythm model for Chinese poetry generation. The model can learn rhymic an tonal patterns from data. These learned patterns can provide a soft constraint on rhythm, thus offers a good trade-off between semantic coherence and rhythm aesthetics. Experiments showed the rhythm model outperforms the baselines not only on rhythmic and tonal compliance. Moreover, since the semantic and rhythmic performances are well balanced, the overall quality of generated poems is improved. Analysis also reveals that pattern matrix is a good structure to represent the rhythm patterns of Chinese poems.

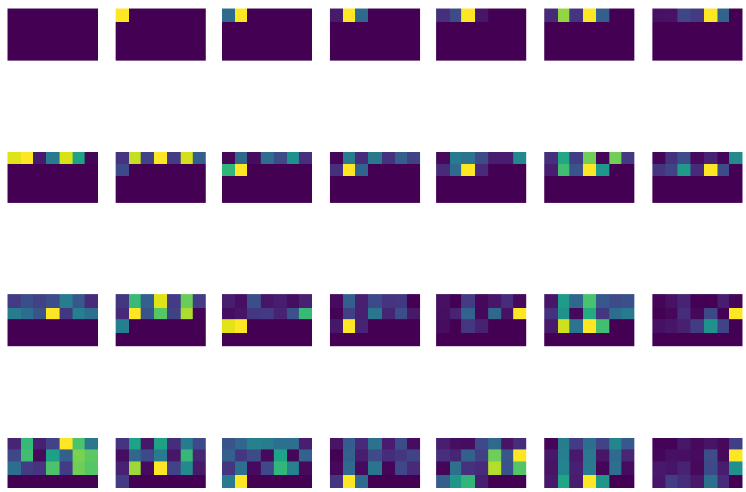

Figure 5 Tonal pattern

Note how the caption is centered in the column.

Acknowledgement

References

1. Yan Rui, Jiang Han, 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 *International Joint Conference on Artificial Intelligence*, 2013.
2. He Jing, Zhou Ming, and Jiang Long, "Generating chinese classical poems with statistical machine translation models," in *Twenty-sixth Aaai Conference on Artificial Intelligence*, 2012.
3. Xingxing Zhang and Mirella Lapata, "Chinese poetry generation with recurrent neural networks," in *Conference on empirical methods in natural language processing, vol. 1: Conference on empirical methods in natural language processing (EMNLP 2014), 25-29 October 2014, Doha, Qatar*, Doha, 1 2014, pp. 670–680, Association for Computational Linguistics.
4. Qixin Wang, Tianyi Luo, Wang Dong, and Xing Chao, "Chinese song iambics generation with neural attention-based model," in *International Joint Conference on Artificial Intelligence*, 2016.
5. Rui Yan, "i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema," pp. 2238–2244, 2016.
6. Jiyuan Zhang, Feng Yang, Wang Dong, Wang Yang, Andrew Abel, Shiyue Zhang, and Andi Zhang, "Flexible and creative chinese poetry generation using neural memory," 2017.
7. Xiaoyuan Yi, Ruoyu Li, and Maosong Sun, "Chinese poetry generation with a salient-clue mechanism," 2018.
8. Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li, "Stylistic chinese poetry generation via unsupervised style disentanglement," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 3960–3969.
9. Matthew D. Zeiler, "Adadelta: An adaptive learning rate method," *Computer Science*, 2012.
10. Juntao Li, Yan Song, Dongmin Chen Haisong Zhang, Shuming Shi, Dongyan Zhao, and Rui Yan, "Generating classical chinese poems via conditional variational autoencoder and adversarial training," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 3960–3969.
11. Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li, "Automatic poetry generation with mutual reinforcement learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 3143–3153.