An Iterative Polishing Framework Based on Quality Aware Masked Language Model for Chinese Poetry Generation

Liming Deng,[1] Jie Wang,[1] Hangming Liang,[1] Hui Chen,[1]
Zhiqiang Xie,[3]* Bojin Zhuang,[1] Shaojun Wang,[1] Jing Xiao[2]
[1]Ping An Technology
[2]Ping An Insurance (Group) Company of China
[3]University of Science and Technology of China
dengliming777@pingan.com.cn, photonicsjay@163.com

Abstract

Owing to its unique literal and aesthetical characteristics, automatic generation of Chinese poetry is still challenging in Artificial Intelligence, which can hardly be straightforwardly realized by end-to-end methods. In this paper, we propose a novel iterative polishing framework for highly qualified Chinese poetry generation. In the first stage, an encoder-decoder structure is utilized to generate a poem draft. Afterwards, our proposed Quality-Aware Masked Language Model (**QA-MLM**) is employed to polish the draft towards higher quality in terms of linguistics and literalness. Based on a multi-task learning scheme, **QA-MLM** is able to determine whether polishing is needed based on the poem draft. Furthermore, **QA-MLM** is able to localize improper characters of the poem draft and substitute with newly predicted ones accordingly. Benefited from the masked language model structure, **QA-MLM** incorporates global context information into the polishing process, which can obtain more appropriate polishing results than the unidirectional sequential decoding. Moreover, the iterative polishing process will be terminated automatically when **QA-MLM** regards the processed poem as a qualified one. Both human and automatic evaluation have been conducted, and the results demonstrate that our approach is effective to improve the performance of encoder-decoder structure.

Introduction

Chinese Poetry, originated from people's production and life, has a long history. The poetry is developed from few characters, vague rules to some fixed characters and lines with stable rules and forms. The rules like tonal pattern, rhyme scheme lead to poems easy to be read and remembered. The great poems, which touch millions of people at heart across the space and time, should unify the concise form, refined language and rich content together to guarantee the long-term prosperity. Writing great poems are not easy, which require strong desire for poets to express their feelings, views or thoughts and then to choose characters and build sentence carefully.

Poets are always regarded as genius with great talents and well trained in writing poems. It is hard to write a poem for ordinary people, let alone to computers. Although many works (Gervás 2001; Ghazvininejad et al. 2016; Yi et al. 2018; Li et al. 2018) have been conducted for automatic poetry generation and poetic rules and forms can be learned partially, the large gaps remain in the meaningfulness and coherence of generated poems.

In this paper, we focus on the automatic Chinese poetry generation and aim to fill these gaps. We notice that poets would first write a poem draft and then polish the draft many times to a perfect one. There is a popular story about polishing poem by Dao Jia, a famous poet in Tang Dynasty, who influences many later poets in polishing their poems intensively. Motivated by the writing poem process of poets, we aim to imitate this process and improve the coherence and meaningfulness of primitive poems. However, it is challenging for computer algorithms to automatically polish the poem draft to an excellent one. The computer algorithms are unable to choose the characters and sentences like poets with intuition and comprehensive understanding of the characters, which are only good at calculating the probability of characters and picking up ones with maximum probability from vocabulary. There are three key issues to be addressed for the polishing framework.

- Whether the text need to be polished, and when should we stop the iterative polishing process?
- Which characters in the text are improper and need to be replaced with better ones?
- How to obtain the better ones?

To address these key issues and further improve the quality of generated poem, we propose a Quality-Aware Masked Language Model (**QA-MLM**) to implement an iterative polishing process. To the best of our knowledge, this is the first work to solve the three key issues in polishing framework with one elegant model.

Our idea originates from the BERT (Devlin et al. 2018) with two-task learning schema, and we modify the tasks to aware of text quality and further obtain appropriate characters to replace the low quality characters in the text. With these two tasks, we can polish the generated poem draft it-

*This work was done when Zhiqiang Xie was at Ping An Technology

eratively, and the polishing process will be terminated automatically. The main contributions of this paper are summarized as follows:

- Our proposed model **QA-MLM**, a novel application of BERT for poem refinement, which can judge the quality of poem and polish the bad characters in the poem iteratively. The polish process will be terminated automatically until the polished poem is regarded as a qualified one by our model.
- The **QA-MLM** model can obtain high quality characters by incorporating both left and right context information, and overcomes the weakness of the unidirectional decoding that only consider one side context information for the next character prediction.
- A two-stage poem generation has been proposed to strengthen the meaningfulness and coherence of generated poems. On the first stage, the encoder-decoder structure has been utilized to generate the poem draft. Specifically, the pre-trained BERT and the transformer decoder has been utilized for poem draft generation. The poem draft is further polished by our proposed **QA-MLM** model on the second stage.

Related Work

Automatic poetry generation has been investigated for decades. The early work focus on improving the grammatical rules and complying with poetic forms with template-based methods (Gervás 2001; Wu, Tosa, and Nakatsu 2009) and evolution algorithms (Manurung 2004; Zhou, You, and Ding 2010). The statistical machine translation methods (He, Zhou, and Jiang 2012) and text summarization methods (Yan et al. 2013) have also been utilized for generating more natural and flexible poems.

As neural network demonstrates powerful ability for natural language representation (Bengio et al. 2003; Goldberg 2017), different neural network structures have been utilized for poem generation and shown great advances. The main structures are developed from vanilla recurrent neural network (Zhang and Lapata 2014) to bi-directional long short term memory network and bi-directional gated recurrent unit network (Wang et al. 2016; Yi, Li, and Sun 2017). The poem generation is widely interpreted as a sequence-to-sequence problem, which utilize the encoder-decoder framework to encode the previous sequence and generate the later sequence with the decoder (Wang, Luo, and Wang 2016; Yi, Li, and Sun 2017). To strengthen the relation between the encoder and decoder, the attention mechanism has been incorporated for poem generation (Wang, Luo, and Wang 2016; Zhang et al. 2017; Yi, Li, and Sun 2017). Besides, some tricks like working memory mechanism and salient-clue mechanism (Yi et al. 2018; Yi, Li, and Sun 2018) have been proposed to improve the coherence in meanings and topics for poem generation. Recently, the conditional variational autoencoder (C-VAE) has been utilized to generate the poem (Yang et al. 2017; Li et al. 2018). The C-VAE can obtain high quality poem with topic coherence to some extent. Some bad cases are also reported by (Li et al. 2018).

All the previous methods are generating the poem directly without any refinement. The most similar work to our paper is *i,poet* (Yan 2016), which has implemented an polishing framework via encoding the writing intents repetitively to rewrite the poem. This work empirically polishes all the characters that generated at previous iterative step, and assumes that the further encoding of the writing intents would enhance the theme consistency. Another similar work is the *Deliberation Network* (Xia et al. 2017), which has utilized a second decoder to generate the final sequence with the additional input of sequence that generated by the first decoder. Followed by the *Deliberation Network*, a more recent work (Zhang et al. 2019) has employed transformer decoder to implement the two-decoder polishing process for text summarization. All these methods fail to sense the text quality and regard rewriting the whole sequence as polishing process, which are inefficiency and may replace qualified characters with low-quality ones. Besides, the polishing process in these methods are heavily coupled with the initial draft generation process, which refers to not only the generated drafts but also the additional information that has been utilized in the draft generation. Therefore, these polishing process cannot be utilized to polish the text drafts that generated separately by other models.

By contrast, our proposed **QA-MLM** model is different to the previous works significantly and implements the iteratively polishing process like humans. Our model can first aware of the text quality and decide whether the text need to be polished. Furthermore, our model can aware of the low-quality characters and pick them up to be polished with both left and right context information. Since our model can sense the text quality, the iterative polish step will be terminated automatically once the polished text has been regarded as qualified. Our model only polishes a small part of low-quality characters instead of rewriting the whole text. The polishing process implemented by our model is independent to the draft generation process, which can polish the draft generated by other separate model. In this paper, we apply our proposed **QA-MLM** model for Chinese poetry polishing, which can significantly improve the quality of poem particular in terms of meaningfulness and coherence. We will introduce our approach in the following sections.

Model Design

Overview

We focus on the generation and polishing of quatrain, which is a famous kind of Chinese classic poetry with strict constraints of poetic rules and forms. In general, the quatrain is with four poem lines, the number of characters for each line is equal, which is either five or seven. The tone level and rhythm are constrained with specific patterns (Wang 2002). We follow the text-to-text generation paradigm (Wang et al. 2016; Li et al. 2018) and generate the poem line by line. The first poem line is generated by keywords or topic words, and the following lines are generated by the preceding lines or their combinations. The key task turns into designing a proper model to generate the following lines with given key-

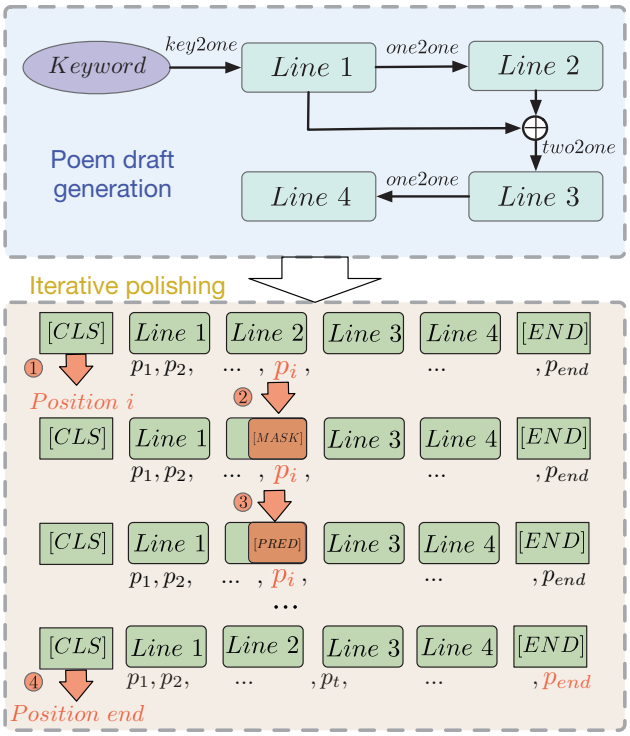

Figure 1: An overview of our poem generation approach, including poem draft generation and iterative polishing process. ① Predict the character position with worst quality, ② Mask the worst quality character, ③ Predict the masked character and update it into the text accordingly, ④ Repeat the previous steps until no character needs to be polished.

words or preceding lines.

Inspired by the real procedure of writing poems for poets, we generate the poem lines with two stages. The poem draft lines are first generated with encoder-decoder framework and then polished with our proposed **QA-MLM**. The overall structure of our approach can be shown in Figure 1.

Input Representation

Our input representation is similar to the embedding methods in BERT (Devlin et al. 2018). In addition to sum the token, segment and position embeddings as the representation, we also add the tonal and rhythm embeddings into the input representation. The tone of each character is either Ping (the level tone) or Ze (the downward tone) (Wang 2002). We encode the tone with three categories due to some tokens like comma without any tone. The rhythm of last character for each poem line will be encoded and we utilize the Thirteen Rhyme. [1] With the tone and rhythm are embedded into the representation, we can improve the poeticness of generated poem significantly without sacrifice much of the poem quality. The visualization of our input representation is given in Figure 2.

[1]Classify the final vowels into thirteen categories according to the rhyme table.

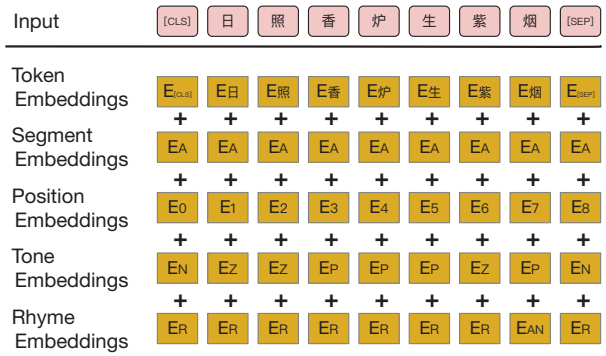

Figure 2: The input representation. The input embeddings is the sum of the token embeddings, segmentation embeddings, position embeddings and tone embeddings as well as rhyme embeddings. The E_N and E_R represent the token without tone or no need to embed the rhyme respectively.

Poem Draft Generation

The poem draft can be generated via encoder-decoder structure. This structure learns the relation between the target sequence $\mathbf{t} = \{t_1, t_2, ..., t_n\}$ and the source sequence $\mathbf{s} = \{s_1, s_2, ..., s_m\}$. The generation probability $P(\mathbf{t}|\mathbf{s}; \Theta)$ can be obtained by Equation (1), where the Θ is model parameters and learned from the sequence pairs $(\mathbf{s}, \mathbf{t}) \in (\mathbb{S}, \mathbb{T})$. We maximize the objective function in Equation (2).

$$P(\hat{\mathbf{t}}|\mathbf{s}; \Theta) = \prod_{j=1}^{n} P(\hat{t}_j|t_{<j}, \mathbf{s}; \Theta) \quad (1)$$

$$L_{character} = \sum_{(\mathbf{s},\mathbf{t})\in(\mathbb{S},\mathbb{T})} \log P(\hat{\mathbf{t}}|\mathbf{s}; \Theta) \quad (2)$$

As shown in Figure 3, the BERT is utilized as the encoder to represent the source sequence $\mathbf{s}$ with vector $\mathbf{h}$, and the representation vector $\mathbf{h}$ is then fed to a two-layer transformer decoder to generate the target sequence $\hat{\mathbf{t}}$ (Devlin et al. 2018; Zhang et al. 2019). The source sequence can be a keyword or poem lines, and the target sequence is the poem line that we want to generate. All or part of previous sequence have been utilized as source sequence by previous works (Wang et al. 2016; Yi, Li, and Sun 2017). After carefully considering the relevance among poem lines and without making the generation system complicated, we build three different models with the same structure for each poem line generation, namely: **key2one**, **one2one**, and **two2one**.

The **key2one** model is utilized to generate the first poem line with the input of keyword. The **one2one** model is employed to generate the second poem line and the fourth poem line due to the similar relevance with their preceding poem lines. As for generating the third poem line, we consider both the first and second poem lines via the **two2one** model. The whole poem draft generation process can be visualized in the upper part of Figure 1.

Iterative Polishing

There is an obvious deficiency for the aforementioned encoder-decoder method (Xia et al. 2017). During the de-

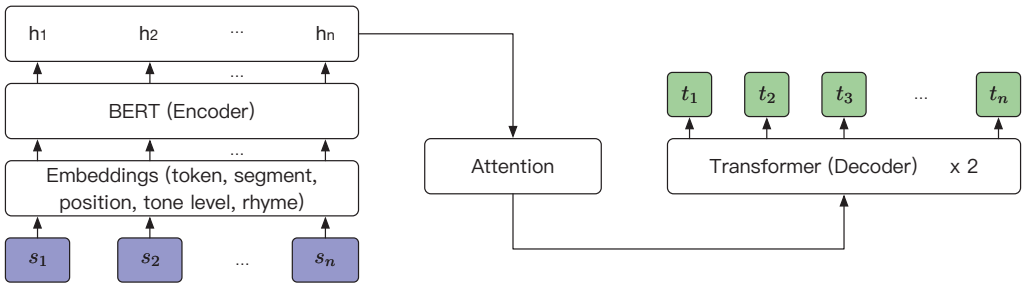

Figure 3: The sequence-to-sequence generation with BERT and Transformer decoder.

coding, the character generated sequentially is affected by the previous characters and ignores the influence of subsequent characters, as demonstrated in Equation (1). Therefore, an iterative polishing framework which can utilize both previous and subsequent context information to polish center character is critical to obtain a semantic consistency and coherence poem.

We propose a quality aware masked language model (**QA-MLM**) to implement the iterative polishing process. The quality aware reflects the model capability of distinguishing the character quality and deciding which character need to be polished. Besides, our model can decide whether the text need to be polished and when should we stop the iterative polishing process. The masked language model is to mask the low quality character first and then predict another character by referring the two-side context information. The predicted character is assumed to be with better semantic consistency and coherence due to the consideration of both the previous and subsequent information. Inspired by (Devlin et al. 2018), we train the **QA-MLM** with two prediction tasks and apply it to polish the generated poem draft, as described in the following subsections.

Prediction Tasks In order to provide reasonable solutions to the aforementioned three key issues in polishing framework, we design a quality prediction task and a masked language model task based on BERT (Devlin et al. 2018). Unlike the BERT, which learns from multi-task for context representation, our proposed **QA-MLM** predicts the positions of low quality characters and then replaces the low quality characters with newly predicted ones for text refinement. The structure of **QA-MLM** can be visualized in Figure 4. The quality prediction task is to predict the character positions that the characters are with low quality. We regard the original poem lines from poetry corpus are the gold standards, and any changing of the original poem lines would hurt the quality of the poem. Therefore, we randomly replace the characters in original poem line with random tokens. We denote s_g as the original poem line, and the s_c as the changed poem line. The positions that have been replaced can be denoted as $p = [p_{i1}, p_{i2}, ..., p_{ir}]$, where $ir < n$, and the real characters that have been replaced are $s_i = [s_{i1}, s_{i2}, ..., s_{ir}]$. The number of replaced positions r reflects the learning capacity of the **QA-MLM** that how many mismatched characters can be learned. A larger r allows a more powerful model for polishing the bad poem

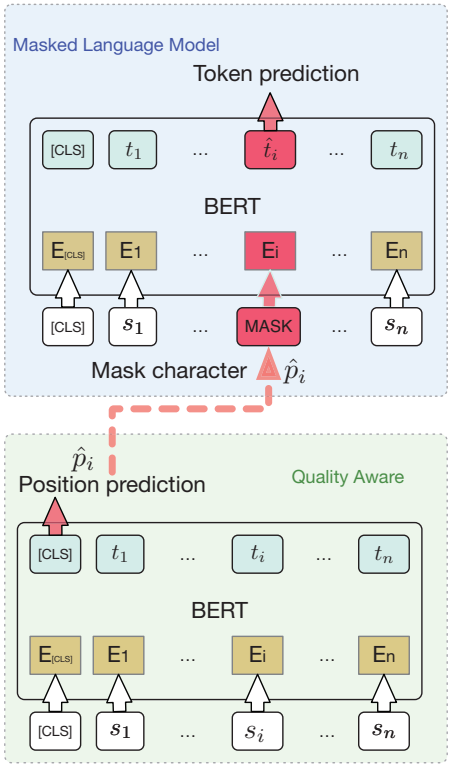

Figure 4: The structure of quality aware masked language model.

lines intensively. However, the large r would lead to the changed poem lines very bad and increases the training difficult. It should be careful to choose an appropriate r considering both the model capacity and learning quality.

In this work, each poem line is randomly replaced [2] according to the following rules:

- 60% of the time: replace one character with random token, eg., the original poem line $\mathbf{s_g} = [s_1, s_2, s_3, s_4, s_5, s_6, s_7]$ ($n = 7$, for example) is changed to $\mathbf{s_c} = [s_1, s_2, \mathbf{s_{i1}}, s_4, s_5, s_6, s_7]$ and the position label is $p = 3$, then the masked poem line is $\mathbf{s_m} = [s_1, s_2, [\mathbf{MASK}], s_4, s_5, s_6, s_7]$.

[2] Refers to both the positions and the tokens are randomly selected

- 20% of the time: replace two characters with random tokens, eg., the original poem line $\mathbf{s_g} = [s_1, s_2, s_3, s_4, s_5, s_6, s_7]$ is changed to $\mathbf{s_c} = [s_1, \mathbf{s_{i1}}, s_3, s_4, s_5, \mathbf{s_{i2}}, s_7]$ and the position label is $p = [2, 6]$, then the masked poem line is $\mathbf{s_m} = [s_1, [\mathbf{MASK}], s_3, s_4, s_5, [\mathbf{MASK}], s_7]$.
- 20% of the time: keep the poem line unchanged, in this situation we set the position label as 0, namely $\mathbf{s_g} = \mathbf{s_c}$ and $p = 0$, there is no need to mask the poem line $\mathbf{s_c}$ and no need to polish poem line $\mathbf{s_c}$, which infers $p_{end} = 0$.

Learning and Inference We can jointly learning the aforementioned two tasks to train **QA-MLM** by minimizing the following loss functions:

$$Loss_q = -\sum_{\mathbf{s_c} \in \mathbb{S}_c} \sum_{j=i1}^{ir} \log P(\hat{p}_j = p_j | \mathbf{s_c}; \Theta) \quad (3)$$

$$Loss_m = -\sum_{(\mathbf{s_m}, \mathbf{s_g}) \in (\mathbb{S}_m, \mathbb{S}_g)} \sum_{j=i1}^{ir} \log P(\hat{s}_{m,j} = s_{g,j} | s_{m,\neq j}; \Theta) \quad (4)$$

$$Loss_{total} = Loss_m + \lambda Loss_q \quad (5)$$

After learning our proposed **QA-MLM** over the constructed poem corpus $(\mathbb{S}_g, \mathbb{S}_c, \mathbb{S}_m)$, we can utilize the **QA-MLM** to polish the poem draft. At the beginning of polishing process, the **QA-MLM** predicts the character position that with worst character quality. If the predicted position p_i is equal to p_{end} (in our setting $p_{end} = 0$), which means that all characters in the draft are qualified enough and no more any polishing, otherwise the character identified with the worst quality will be masked in the draft, and the masked draft will be further utilized for a better character prediction via **QA-MLM** model. The predicted character is regarded as more appropriate than the masked character due to the incorporation of two-side context information during the prediction. Thus, we replace the masked character in sequence draft with the predicted character, and one polishing step is finished. By repeating the above polishing step, the sequence draft can be iteratively polished many times until the end indicator p_{end} is predicted. At this time, the iterative polishing process will be terminated automatically, and the sequence draft has been polished completely. The iterative polishing process can be shown in Algorithm 1.

Algorithm 1 : Iterative Polishing with **QA-MLM**

```
Perform iterative polishing on sequence draft s_d = [s_1,s_2,...,s_i,...,s_n]
Predict bad character position p_i = QA(s_d)
while p_i ≠ p_end do
    s_i ← [MASK] , s_d = [s_1,s_2,...,[MASK],...,s_n]
    Predict the new character ŝ_i = MLM(s_d)
    s_d ← [s_1,s_2,...,ŝ_i,...,s_n]
    p_i ← QA(s_d)
return polished s_d
```

The sequence draft can be a poem line or several poem lines and even a whole poem, our approach is capable of polishing all of them. In this work, we polish the whole poem together, which incorporates the whole context information for inappropriate character prediction, and the inappropriate characters will be replaced with highly qualified characters to obtain semantic consistency and coherence.

Experiments and Evaluations

Experimental Setup

In this paper, we concentrate on the generation of Chinese quatrain with seven fixed characters for each line. Our poetry corpus is consist of poems from *Tang Dynasty*, *Song Dynasty*, *Yuan Dynasty*, *Ming Dynasty* and *Qing Dynasty*. About 130,525 poems with total 905,790 number of poem lines are filtered from the poetry corpus. Each filtered poem contain four or multiple of four poem lines, and each poem line with seven characters. These poems are randomly split into three part for model training (90%), validation (5%) and testing (5%). Three models (**key2one**, **one2one**, and **two2one**) trained by different sequence-to-sequence pairs are utilized to generate the poem draft lines.

The BERT is selected as the encoder with 12 layers and initialized with the parameters pre-trained by (Devlin et al. 2018). The 2-layer transformer decoder is selected for the poem generation. After the poem draft has been generated, the **QA-MLM** is proposed for the polishing. The aware of poem quality is implemented by the quality task to predict the position character that with worst semantic quality. In addition to the current total 28 positions for the seven-character quatrain, an end position $p_{end} = 0$ is added to indicate the end of iterative polishing. The character located by the quality prediction task will be masked and then replaced with newly predicted one by masked language model task. The quality prediction task and the masked language model task are based on the 12-layer BERT and learned jointly.

The conventional RNN encoder-decoder structure with attention mechanism (**AS2S**) (Wang et al. 2016) and a more recent work **CVAE-D** (Li et al. 2018) are selected as the baselines for poem draft generation. Besides, we also implement a more powerful encoder-decoder structure with pre-trained BERT and transformer decoder (**B&T**) for poem draft generation. The poem drafts are generated with the input of keywords or writing intents, and we follow the keywords extraction method adopted by **AS2S** (Wang et al. 2016), which cuts the poem lines into several word segmentations by *Jieba* (Sun 2012) and then utilizes the *TextRank* (Mihalcea and Tarau 2004) method to select keyword with the highest score. The poems generated by the aforementioned three models are further polished with the proposed **QA-MLM**. Both the automatical evaluation criteria and human evaluations have been conducted. The following subsection will introduce the detail about the evaluations.

Evaluation Metrics

It is difficult for computer to estimate the quality of poem. Therefore, we utilize both automatic evaluation metrics and human judgements for model comparisons. The automatic evaluation metrics including BLEU (Papineni et al. 2002),

Table 1: The BLEU score results on different generated poem line with same extracted keyword or previous poem lines. **BL-1** and **BL-2** are the BLEU scores on unigrams and bigrams.

Model	key → 1		1 → 2		1&2 → 3		3 → 4		Average	
	BL-1	BL-2	BL-1	BL-2	BL-1	BL-2	BL-1	BL-2	BL-1	BL-2
AS2S	0.072	0.047	0.016	0.005	0.019	0.006	0.021	0.007	0.032	0.016
AS2S-P	0.074	0.047	0.026	0.009	0.030	0.010	0.036	0.012	**0.042**	**0.020**
CVAE-D	0.109	0.059	0.013	0.005	0.015	0.005	0.019	0.006	0.039	0.019
CVAE-D-P	0.105	0.057	0.015	0.005	0.016	0.006	0.021	0.007	0.039	0.019
B&T	0.102	0.050	0.028	0.010	0.036	0.014	0.035	0.013	**0.050**	**0.022**
B&T-P	0.100	0.050	0.028	0.009	0.034	0.013	0.033	0.012	0.049	0.021

Table 2: The evaluation results. **Sim12** refers to the similarity between first poem line and second poem line; **Sim34** refers to the similarity between the third poem line and the fourth poem line; **Sim2L** refers to the similarity between first two poem lines and the last two poem lines; **TA.** and **RA.** are the tone level predicted accuracy and the rhythm predicted accuracy respectively; **Con.**, **Flu.**, **Mea.**, and **Poe.** represent the *Consistency*, *Fluency*, *Meaningfulness* and *Poeticness* respectively.

Model	Automatic Evaluation					Human Evaluation			
	Sim12	Sim34	Sim2L	TA.	RA.	Con.	Flu.	Mea.	Poe.
AS2S	0.479	0.487	0.648	**0.539**	0.121	2.46	2.37	2.35	2.28
AS2S-P	**0.484**	**0.495**	**0.650**	0.517	**0.124**	**2.64**	**2.63**	**2.59**	**2.59**
CVAE-D	0.494	0.500	0.651	0.521	0.086	2.62	2.50	2.55	2.42
CVAE-D-P	**0.499**	**0.507**	**0.653**	**0.524**	**0.091**	**2.75**	**2.72**	**2.74**	**2.64**
B&T	0.500	0.516	0.640	**0.976**	**0.956**	3.01	2.99	3.06	2.88
B&T-P	**0.502**	**0.519**	**0.642**	0.962	0.841	**3.14**	**3.19**	**3.24**	**3.09**

Similarity (Wieting et al. 2015), tone accuracy and rhythm accuracy are adopted in this paper.

The **BLEU** is designed for machine translation and also widely adopted by many previous works (Zhang and Lapata 2014; Li et al. 2018) in poem generation. The BLEU is to measure the overlapping of characters between the generated sentence and the referred sentence. Unlike the machine translation, the generated sentence can be significantly different from the referred sentence but also regarded as high quality by human judgements. The poem generation is more related to creativity and the generated poem may far away from the referred poem, which may lead to the comparison of BLEU score is trivial. Therefore, we compare the BLEU score on one sentence instead of the whole poem. Each sentence is generated by different approaches with the same keyword or sentence input.

The **Similarity** is aimed to automatically measure the coherency or consistency among poem lines. The embedding of characters can partially reflect the similarities and we accumulate the embeddings of all the characters for each poem line for sentence-level embeddings (Wieting et al. 2015). Then, the similarity between two poem lines can be calculated by the $cosine$ function on the sentence-level embeddings.

The **Tone Accuracy** and **Rhythm Accuracy** are also employed for the evaluation. The tone accuracy is the percentage that the tone level (Ping or Ze) is predicted correct to all the generated samples, and the rhythm accuracy is similar about the last character of each poem line that the rhyme is predicted correct.

The **Human Evaluation** is inevitable for poem evaluation, which is more reliable and credible than the automatic evaluation metrics. Twenty well educated annotators are invited to evaluate the generated poems in four dimensions, namely *Consistency*, *Fluency*, *Meaningfulness* and *Poeticness* (Zhang and Lapata 2014; Li et al. 2018). Each dimension is rated using the 1 to 5 scale to represent from bad quality to excellent. Each model generates one thousand poems and the poems are divided equally into twenty pieces. To reduce the individual scoring bias, the poems rated by each participant are from all models, but the participant has no information about the model that each poem belongs to. Therefore, we can obtain 6000 ($20 \times 6 \times 50$) poem ratings.

Results and Discussions

The BLEU scores are compared in Table 1. The compared models are shown in the first column of the table, where the suffix **-P** indicates that the poems generated by previous models have been polished by **QA-MLM**. The keywords and poem lines are extracted from test dataset with one thousand poems. In general, the BLEU scores of **CVAE-D** are higher than **AS2S** but lower than **B&T**, which partially reflects the generation performance of these models. The polishing process improves the BLEU scores for **AS2S** model while has no obvious improvement or even a bit hurt of BLEU scores for **CVAE-D** and **B&T**. This is probably due to the creativity and diversity properties in poem generation. The quality of generated poem is not proportional to the BLEU score when the BLEU score is comparative high. The BLEU score should be referred conservative during the quality evaluation for poetry generation.

The other automatic evaluation results and human score results can be found in Table 2. These evaluation results are also based on one thousand poems from test dataset. The

Figure 5: The example of poem draft and the polished poem. The left poem is the poem draft generated by *B&T* and the right poem is the poem iteratively polished by **QA-MLM**. The iterative polishing is automatically terminated after 3 polish steps.

keywords are extracted from these poems and utilized to generate poem drafts by **AS2S**, **CVAE-D** and **B&T**. All the poem drafts generated by this three encoder-decoder structure models have been further polished by our proposed **QA-MLM** model. We can find that the polish procedure improves the scores of **Similarity** criteria on all the compared encoder-decoder structure models, which demonstrates the effectiveness of our proposed **QA-MLM** for the improvement of sematic coherence and theme consistency. The tone level and rhythm for **AS2S** and **CVAE-D** models seem randomly. By contrast, the accuracy of tone level and rhythm for **B&T** are significantly higher than the other two baselines, which demonstrates the embedding-based method is effective to control the tone level and rhythm. The polishing framework on the **B&T** hurts a bit accuracy of tone level and rhythm, which sacrifices the tone level and rhythm constraints to obtain better semantic meaning and coherence.

The human evaluations are consistent with the automatic evaluation metrics. As for the poem draft generators, the **CVAE-D** outperforms the **AS2S** on all the quality aspects evaluated by our annotators, which is consistent with the results from (Li et al. 2018). However, the poem generation performance of **B&T** is better than **CVAE-D**, which is probability due to the powerful text representation ability of BERT or transformer.

Above all, our proposed **QA-MLM** can further improve the qualities (*consistency*, *fluency*, *meaningfulness* and *poeticness*) of poems generated by all the three aforementioned encoder-decoder structure models, which demonstrates that the quality prediction task is effective to locate the bad characters and the masked language model task is also effective to obtain better predictions when referring to the global context information. Therefore, the unidirectional sequential decoding deficiency of the encoder-decoder structure can be largely saved by our proposed **QA-MLM** for poetry refinement.

Although the improvements brought by the **QA-MLM** in automatic evaluation metrics seem trivial, the improvements are significant in the human evaluation results. It is easy to understand that the polishing process only update a small part of characters, and the improvements will be averaged on all the characters for automatic evaluation metrics. However, human can understand and notice the significant difference of the changed characters to the whole poem context. Even one character changed would lead to a big improvement for poetry quality. We can notice the difference from the example in Figure 5.

Conclusion

In this paper, we present an iterative polishing framework for Chinese poetry generation by imitating the real poem writing process. Following the famous encoder-decoder paradigm, a pre-trained BERT encoder and a transformer decoder are combined to generate poem drafts. Then, poem polishing is accomplished by a multifunctional **QA-MLM**, which can improve poem quality in terms of semantics, syntactics and literary. Based on the multi-task learning, the trained **QA-MLM** is able to aware of the poem quality and locate improper characters. Besides, the **QA-MLM** is capable of predicting better ones to replace the improper characters by synthesizing the all-round poem context information. Moreover, the **QA-MLM** will automatically terminate the iterative polishing process when the polished draft is classified as qualified.

Both automatic evaluation and human scores demonstrate that our proposed approach is effective in Chinese poetry generation. Our model can automatically modify preliminary poems to elegant ones while keeping their original intents. Even though our proposed **QA-MLM** polishing framework is concentrated on Chinese poetry generation in this work, this new text refinement approach can be extended to other natural language generation areas.

Acknowledgments

We thank Haoshen Fan, Weijing Huang, Mingkuo Ji and Shaopeng Ma for helpful discussions.

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems* 14(3-4):181–188.

Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1183–1191.

Goldberg, Y. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies* 10(1):1–309.

He, J.; Zhou, M.; and Jiang, L. 2012. Generating chinese classical poems with statistical machine translation models. In *AAAI*.

Li, J.; Song, Y.; Zhang, H.; Chen, D.; Shi, S.; Zhao, D.; and Yan, R. 2018. Generating classical chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3890–3900.

Manurung, H. 2004. An evolutionary algorithm approach to poetry generation.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc Meeting of the Association for Computational Linguistics*.

Sun, J. 2012. 'jieba'chinese word segmentation tool.

Wang, Z.; He, W.; Wu, H.; Wu, H.; Li, W.; Wang, H.; and Chen, E. 2016. Chinese poetry generation with planning based neural network. *arXiv preprint arXiv:1610.09889*.

Wang, Q.; Luo, T.; and Wang, D. 2016. Can machine generate traditional chinese poetry? a feigenbaum test. In *International Conference on Brain Inspired Cognitive Systems*, 34–46. Springer.

Wang, L. 2002. A summary of rhyming constraints of chinese poems.

Wieting, J.; Bansal, M.; Gimpel, K.; and Livescu, K. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Wu, X.; Tosa, N.; and Nakatsu, R. 2009. New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system. In Natkin, S., and Dupire, J., eds., *Entertainment Computing – ICEC 2009*, 191–196. Berlin, Heidelberg: Springer Berlin Heidelberg.

Xia, Y.; Tian, F.; Wu, L.; Lin, J.; Qin, T.; Yu, N.; and Liu, T.-Y. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, 1784–1794.

Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. i, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *IJCAI*, 2197–2203.

Yan, R. 2016. i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *IJCAI*, 2238–2244.

Yang, X.; Lin, X.; Suo, S.; and Li, M. 2017. Generating thematic chinese poetry with conditional variational autoencoder. *CoRR*.

Yi, X.; Sun, M.; Li, R.; and Yang, Z. 2018. Chinese poetry generation with a working memory model. *arXiv preprint arXiv:1809.04306*.

Yi, X.; Li, R.; and Sun, M. 2017. Generating chinese classical poems with rnn encoder-decoder. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer. 211–223.

Yi, X.; Li, R.; and Sun, M. 2018. Chinese poetry generation with a salient-clue mechanism. *arXiv preprint arXiv:1809.04313*.

Zhang, X., and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 670–680.

Zhang, J.; Feng, Y.; Wang, D.; Wang, Y.; Abel, A.; Zhang, S.; and Zhang, A. 2017. Flexible and creative chinese poetry generation using neural memory. *arXiv preprint arXiv:1705.03773*.

Zhang, H.; Gong, Y.; Yan, Y.; Duan, N.; Xu, J.; Wang, J.; Gong, M.; and Zhou, M. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.

Zhou, C.-L.; You, W.; and Ding, X. 2010. Genetic algorithm and its implementation of automatic generation of chinese songci. *Journal of Software* 21(3):427–437.