Automatic Poetry Generation with Mutual Reinforcement Learning

Xiaoyuan Yi[1,2,3], **Maosong Sun**[1,2,4]*, **Ruoyu Li**[5], **Wenhao Li**[1,2,3]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Institute for Artificial Intelligence, Tsinghua University
[3]State Key Lab on Intelligent Technology and Systems, Tsinghua University
[4]Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University
[5]6ESTATES PTE LTD, Singapore
{yi-xy16, liwh16}@mails.tsinghua.edu.cn,
sms@mail.tsinghua.edu.cn, liruoyu@6estates.com

Abstract

Poetry is one of the most beautiful forms of human language art. As a crucial step towards computer creativity, automatic poetry generation has drawn researchers' attention for decades. In recent years, some neural models have made remarkable progress in this task. However, they are all based on maximum likelihood estimation, which only learns common patterns of the corpus and results in loss-evaluation mismatch. Human experts evaluate poetry in terms of some specific criteria, instead of word-level likelihood. To handle this problem, we directly model the criteria and use them as explicit rewards to guide gradient update by reinforcement learning, so as to motivate the model to pursue higher scores. Besides, inspired by writing theories, we propose a novel mutual reinforcement learning schema. We simultaneously train two learners (generators) which learn not only from the teacher (rewarder) but also from each other to further improve performance. We experiment on Chinese poetry. Based on a strong basic model, our method achieves better results and outperforms the current state-of-the-art method.

1 Introduction

Language is one of the most important forms of human intelligence and poetry is a concise and graceful art of human language. Across different countries, nationalities and cultures, poetry is always popular, having far-reaching influence on the development of human society.

In this work, we concentrate on automatic poetry generation. Besides the long-term goal of building artificial intelligence, research on this task could become the auxiliary tool to better analyse poetry and understand the internal mechanism of human writing. In addition, these generation systems are also helpful for electronic entertainments and literary education.

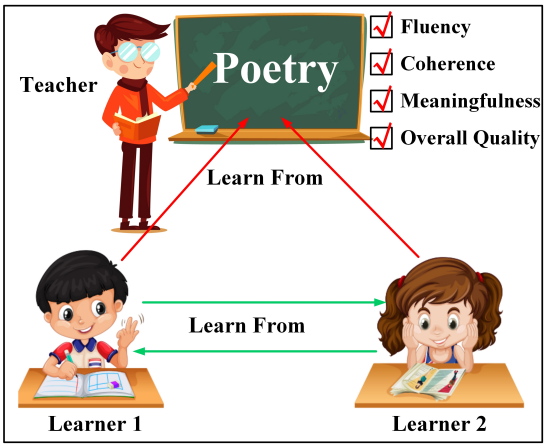

Figure 1: An artistic illustration of our mutual reinforcement learning method.

In recent years, neural networks have proven to be powerful on poetry generation. Some neural models are proposed and achieve significant improvement. However, existing models are all based on maximum likelihood estimation (MLE), which brings two substantial problems. First, MLE-based models tend to remember common patterns of the poetry corpus (Zhang et al., 2017), such as high-frequency bigrams and stop words, losing some diversity and innovation for generated poetry. Moreover, based on word-level likelihood, two kinds of loss-evaluation mismatch (Wiseman and Rush, 2016) arise. One is *evaluation granularity mismatch*. When evaluating, human experts usually focus on sequence level (a poem line) or discourse level (a whole poem), while MLE optimizes word-level loss, which fails to hold a wider view of generated poems. The other is *criteria mismatch*. Instead of the likelihood, humans usually evaluate poetry in terms of some criteria. In this work we focus on the main four criteria

*Corresponding author: sms@mail.tsinghua.edu.cn.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3143–3153
Brussels, Belgium, October 31 - November 4, 2018.

(Manurung, 2003; Zhang and Lapata, 2014; Yan, 2016; Yi et al., 2017): **fluency** (are the lines fluent and well-formed?), **coherence** (is the poem as a whole coherent in meaning and theme?), **meaningfulness** (does the poem convey some certain messages?), **overall quality** (the reader's general impression on the poem). This mismatch may make the model lean towards optimizing easier criteria, e.g., fluency, and ignore other ones.

To tackle these problems, we directly model the four aforementioned human evaluation criteria and use them as explicit rewards to guide gradient update by reinforcement learning. This is a criterion-driven training process, which motivates the model to generate poems with higher scores on these criteria. Besides, in writing theories, writing requires observing other learners (Bandura, 2001). It is also shown that writing is supported as an activity in which writers will learn from more experienced writers, such as other students, teachers, or authors (Prior, 2006). Therefore it is necessary to equip generators with the ability of mutual learning and communication. Inspired by this, we propose a novel mutual reinforcement learning schema (Figure 1), where we simultaneously train two *learners* (generators). During the training process, one learner will learn not only from the *teacher* (rewarder) but also from the other. We will show this mutual learning-teaching process leads to better results.

In summary, our contributions are as follows:

- To the best of our knowledge, for the sake of tackling the loss-evaluation mismatch problem in poetry generation, we first utilize reinforcement learning to model and optimize human evaluation criteria.
- We propose a novel mutual reinforcement learning schema to further improve performance, which is **transparent** to model architectures. One can apply it to any poetry generation model.
- We experiment on Chinese quatrains. Both automatic and human evaluation results show that our method outperforms a strong basic method and the state-of-the-art model.

2 Related Work

As a desirable entry point of automatic analysing, understanding and generating literary text, the research on poetry generation has lasted for decades. In recent twenty years, the models can be categorized into two main paradigms.

The first one is based on statistical machine learning methods. Genetic algorithms (Manurung, 2003; Levy, 2001), Statistical Machine Translation (SMT) approaches (He et al., 2012; Jiang and Zhou, 2008) and Automatic Summarization approaches (Yan et al., 2013) are all adopted to generate poetry.

More recently, the second paradigm, neural network, has shown great advantages in this task, compared to statistical models. Recurrent Neural Network (RNN) is first used to generate Chinese quatrains by (Zhang and Lapata, 2014). To improve fluency and coherence, Zhang's model needs to be interpolated with extra SMT features as shown in their paper. Focusing on coherence, some works (Yi et al., 2017; Wang et al., 2016a) use sequence-to-sequence model with attention mechanism (Bahdanau et al., 2015) to generate poetry. Wang et al. (2016b) design a special Planning schema, which plans some sub-keywords in advance by a language model and then generates each line with the planned sub-keyword to improve coherence. Pursuing better overall quality, Yan (2016) proposes an iterative polishing schema to generate Chinese poetry, which refines the poem generated in one pass for several times. Aiming at enhancing meaningfulness, Ghazvininejad et al. (2016) extend user keywords to incorporate richer semantic information. Zhang et al. (2017) combine a neural memory, which saves hundreds of human-authored poems, with a sequence-to-sequence model to improve innovation of generated poems and achieve style transfer.

These neural structures have made some progress and improved different aspects of generated poetry. Nevertheless, as discussed in Section 1, the two essential problems, lack of diversity and loss-evaluation mismatch, are still challenging resulting from MLE. Compared to further adjusting model structures, we believe a better solution is to design more reasonable optimization objectives.

Deep Reinforcement Learning (DRL) first shows its magic power in automatic game playing, such as Atari electronic games (Mnih et al., 2013) and the game of Go (Silver et al., 2016). Soon, DRL is used to playing text games (Narasimhan et al., 2015; He et al., 2016) and then applied to dialogue generation (Li et al., 2016b).

From the perspective of poetry education, the

teacher will judge student-created poems in terms of some specific criteria and guide the student to cover the shortage, which naturally accords with DRL process. Therefore we take advantage of DRL. We design four automatic rewarders for the criteria, which act as the teacher. Furthermore, we train two generators and make them learn from each other, which imitates the mutual learning of students, as a step towards multi-agent DRL in literary text generation.

3 Methods

3.1 Basic Generation Model

We apply our method to a basic poetry generation model, which is pre-trained with MLE. Therefore, we first formalize our task and introduce this model.

The inputs are user topics specified by K keywords, $\mathcal{W} = \{w_k\}_{k=1}^{K}$. The output is a poem consisting of n lines, $\mathcal{P} = L_1, L_2, \cdots, L_n$. Since we take the line-by-line generation process, the task can be converted to the generation of an i-th line given previous i-1 lines $L_{1:i-1}$ and $\mathcal{W}$.

We use GRU-based (Cho et al., 2014) sequence-to-sequence model. $\overrightarrow{h}_t$, $\overleftarrow{h}_t$ and s_t represent the forward encoder, backward encoder and decoder hidden states respectively. For each topic word $w_k = c_1, c_2, \cdots, c_{T_k}$, we feed characters into the encoder and get the keyword representation $v_k = [\overrightarrow{h}_{T_k}; \overleftarrow{h}_{T_k}]$, where [;] means concatenation. Then we get the topic representation by[1]:

$$o = f(\frac{1}{K}\sum_{t=1}^{K} v_k), \tag{1}$$

where f defines a non-linear layer.

Denote the generated i-th line in decoder, $Y = (y_1 y_2 \dots y_{T_i})$. $e(y_t)$ is the word embedding of y_t. The probability distribution of each y_t to be generated in L_i is calculated by:

$$s_t = GRU(s_{t-1}, [e(y_{t-1}); o; g_{i-1}]), \tag{2}$$
$$P(y_t|y_{1:t-1}, L_{1:i-1}, \mathcal{W}) = softmax(W s_t), \tag{3}$$

where W is the projection parameter. g_{i-1} is a global history vector, which records what has been generated so far and provides global-level information for the model. Once L_i is generated, it is updated by a convolutional layer:

$$a_t = f([s_t; \cdots; s_{t+d-1}]), \tag{4}$$
$$g_i = f(g_{i-1}, \sum_t a_t), g_0 = \mathbf{0}, \tag{5}$$

where $\mathbf{0}$ is a vector with all 0-s and d is convolution window size. Then the basic model is pre-trained by minimizing standard MLE loss:

$$\mathcal{L}_{MLE}(\theta) = -\sum_{m=1}^{M} logP(\mathcal{P}^m|\mathcal{W}^m; \theta), \tag{6}$$

where M is data size and θ is the parameter set to be trained.

This basic model is a modified version of (Yan, 2016). The main differences are that we replace vanilla RNN with GRU unit, use convolution to calculate the line representation rather than directly use the last decoder hidden state, and we remove the polishing schema to better obverse the influence of DRL itself. We select this model as our basic framework since it achieves satisfactory performance and the author has done thorough comparisons with other models, such as (Yan et al., 2013) and (Zhang and Lapata, 2014).

3.2 Single-Learner Reinforcement Learning

Before presenting the single-learner version of our method (abbreviated as **SRL**), we first design corresponding automatic rewarders for the four human evaluation criteria.

Fluency Rewarder. We use a neural language model to measure fluency. Given a poem line L_i, higher probability $P_{lm}(L_i)$ indicates the line is more likely to exist in the corpus and thus may be more fluent and well-formed. However, it's inadvisable to directly use $P_{lm}(L_i)$ as the reward, since over high probability may damage diversity and innovation. We expect moderate probabilities which fall into a reasonable range, neither too high nor too low. Therefore, we define the fluency reward of a poem $\mathcal{P}$ as:

$$r(L_i) = max(|P_{lm}(L_i) - \mu| - \delta_1 * \sigma, 0), \tag{7}$$
$$R_1(\mathcal{P}) = \frac{1}{n}\sum_{i=1}^{n} exp(-r(L_i)), \tag{8}$$

where μ and σ are the mean value and standard deviation of P_{lm} calculated over all training sets. δ_1 is a hyper-parameter to control the range.

Coherence Rewarder. For poetry, good coherence means each line L_i should be coherent with

[1]For brevity, we omit biases in all equations.

previous lines in a poem. We use Mutual Information (MI) to measure the coherence of L_i and $L_{1:i-1}$. As shown in (Li et al., 2016a), MI of two sentences, S_1 and S_2, can be calculated by:

$$MI(S_1, S_2) = logP(S_2|S_1) - \lambda logP(S_2), \quad (9)$$

where λ is used to regulate the weight of generic sentences. Based on this, we calculate the coherence reward as:

$$\begin{aligned} MI(L_{1:i-1}, L_i) = & logP_{seq2seq}(L_i|L_{1:i-1}) \\ & - \lambda logP_{lm}(L_i), \end{aligned} \quad (10)$$

$$R_2(\mathcal{P}) = \frac{1}{n-1}\sum_{i=2}^{n} MI(L_{1:i-1}, L_i), \quad (11)$$

where $P_{seq2seq}$ is a GRU-based sequence-to-sequence model, which takes the concatenation of previous i-1 lines as input, and predicts L_i. A better choice is to use a dynamic λ instead of a static one. Here we directly set $\lambda = exp(-r(L_i)) + 1$, which gives smaller weights to lines with extreme language model probabilities.

Meaningfulness Rewarder. In dialogue generation task, neural models are prone to generate generic sentences such as "I don't know" (Li et al., 2016a; Serban et al., 2016). We observed similar issues in poetry generation. The basic model tends to generate some common and meaningless words, such as *bu zhi* (don't know), *he chu* (where), and *wu ren* (no one). It's quite intractable to quantify the meaningfulness of a whole poem, but we find that TF-IDF values of human-authored poems are significantly higher than values of generated ones (Figure 2). Consequently, we utilize TF-IDF to motivate the model to generate more meaningful words. This is a simple and rough attempt, but it makes generated poems more "meaningful" from the readers perspective.

Direct use of TF-IDF leads to serious out-of-vocabulary (OOV) problem and high variance, because we need to sample poems during the training process of DRL, which causes many OOV words. Therefore we use another neural network to smooth TF-IDF values. In detail, we have:

$$R_3(\mathcal{P}) = \frac{1}{n}\sum_{i=1}^{n} F(L_i), \quad (12)$$

where $F(L_i)$ is a neural network which takes a line as input and predicts its estimated TF-IDF value. For each line in training sets, we calculate standard TF-IDF values of all words and use the average as the line TF-IDF value. Then we use them to train $F(L_i)$ with Huber loss.

Overall Quality Rewarder. The three kinds of rewards above are all based on line-level. In fact, human experts will also focus on discourse-level to judge the overall quality of a poem, ignoring some minor defects. We train a neural classifier to classify a given poem (in terms of the concatenation of all lines) into three classes: computer-generated poetry (class 1), ordinary human-authored poetry (class 2) and masterpiece (class 3). Then we get the reward by:

$$R_4(\mathcal{P}) = \sum_{k=1}^{3} P_{cl}(k|\mathcal{P}) * k. \quad (13)$$

This classifier should be as reliable as possible. Due to the limited amount of masterpieces, normal classifiers don't work well. Therefore we use an adversarial training based classifier (Miyato et al., 2017), which achieves F-1 0.96, 0.73, 0.76 for the three classes respectively on the validation set.

Based on these rewarders, the total reward is:

$$R(\mathcal{P}) = \sum_{j=1}^{4} \alpha_j * \tilde{R}_j(\mathcal{P}), \quad (14)$$

where α_j is the weight and the symbol $\tilde{}$ means the four rewards are re-scaled to the same magnitude. As (Gulcehre et al., 2018), we reduce the variance by:

$$R^{'}(\mathcal{P}) = \frac{R(\mathcal{P}) - b_u}{\sqrt{\sigma_u^2 + \epsilon}} - B(\mathcal{P}), \quad (15)$$

where b_u and σ_u are running average and standard deviation of R respectively. $B(\mathcal{P})$ is a neural network trained with Huber loss, which takes a poem as input and predicts its estimated reward.

DRL Process. For brevity, we use $P_g(\cdot|\mathcal{W}; \theta)$ to represent a basic generator and use REINFORCE algorithm (Williams, 1992) to optimize the model, which minimizes:

$$\mathcal{L}_{DRL}(\theta) = -\sum_{m=1}^{M} \mathbb{E}_{\mathcal{P}\sim P_g(\cdot|\mathcal{W}^m;\theta)}(R^{'}(\mathcal{P})). \quad (16)$$

Training with solely Eq.(16) is unstable. Lacking of original MLE supervisory signals, the

Algorithm 1 Global Mutual Learning

Set history reward lists V_1 and V_2 empty;
for number of iterations **do**
  Sample batch $(\mathcal{W}^m, \mathcal{P}_g^m)$ from training data set;
  for each $\mathcal{W}^m$ **do**
    Sample $\mathcal{P}_1^m \sim P_g(\cdot|\mathcal{W}^m; \theta_1)$;
    Sample $\mathcal{P}_2^m \sim P_g(\cdot|\mathcal{W}^m; \theta_2)$;
    Add $R(\mathcal{P}_1^m)$ to V_1, $R(\mathcal{P}_2^m)$ to V_2
  end for
  Set $\mathcal{L}_M(\theta_1)$=$\mathcal{L}(\theta_1)$, $\mathcal{L}_M(\theta_2)$=$\mathcal{L}(\theta_2)$;
  if mean value $\overline{V_2} > \overline{V_1} * (1 + \delta_3)$ **then**
    $\mathcal{L}_M(\theta_1)$=$\mathcal{L}(\theta_1) + KL(P_g(\theta_2)||P_g(\theta_1))$;
  else if $\overline{V_1} > \overline{V_2} * (1 + \delta_3)$ **then**
    $\mathcal{L}_M(\theta_2)$=$\mathcal{L}(\theta_2) + KL(P_g(\theta_1)||P_g(\theta_2))$;
  end if
  Update θ_1 with $\mathcal{L}_M(\theta_1)$, θ_2 with $\mathcal{L}_M(\theta_2)$;
end for

model is easy to get lost and totally ignore the corresponding topics specified by $\mathcal{W}$, leading to explosive increase of MLE loss. We use two steps to alleviate this issue. The first one is the Teacher Forcing (Li et al., 2017). For each $\mathcal{W}$, we estimate $\mathbb{E}(R^{'}(\mathcal{P}))$ by n_s sampled poems, as well as the ground-truth $\mathcal{P}_g$ whose reward is set to $max(R^{'}(\mathcal{P}_g), 0)$. The second step is to combine MLE loss and DRL loss as:

$$\mathcal{L}(\theta) = (1 - \beta) * \mathcal{L}_{MLE}(\theta) + \beta * \tilde{\mathcal{L}}_{DRL}(\theta), \quad (17)$$

where $\tilde{}$ means the DRL loss is re-scaled to the same magnitude with MLE loss. Ultimately, we use Eq.(17) to fine-tune the basic model.

3.3 Mutual Reinforcement Learning

As discussed in Section 1 & 2, to further improve the performance, we mimic the mutual writing learning activity by simultaneously training two generators defined as $P_g(\theta_1)$ and $P_g(\theta_2)$. The two learners (generators) learns not only from the teacher (rewarders) but also from each other.

From the perspective of machine learning, one generator may not explore the policy space sufficiently and thus is easy to get stuck in the local minima. Two generators can explore along different directions. Once one generator finds a better path (higher reward), it can communicate with the other and lead it towards this path. This process could also be considered as the ensemble of different generators during the training phase.

Models	$\tilde{R}_1$	$\tilde{R}_2$	$\tilde{R}_3$	$\tilde{R}_4$	R
Base	0.156	0.214	0.509	0.351	0.282
Mem	0.192	0.257	0.467	0.383	0.308
MRL	**0.207**	**0.268**	**0.613**	**0.494**	**0.369**
GT	0.582	0.609	0.625	0.759	0.649
SRL	0.169	0.228	0.563	0.432	0.321
LMRL	0.187	0.246	0.602	0.467	0.348
GMRL	0.199	0.262	0.606	0.480	0.360
MRL	**0.207**	**0.268**	**0.613**	**0.494**	**0.369**

Table 1: Automatic rewards of different models and strategies. $\tilde{R}_1$: fluency, $\tilde{R}_2$: coherence, $\tilde{R}_3$: meaningfulness, $\tilde{R}_4$: overall quality, R: weighted-average reward. LMRL: local MRL, GMRL: global MRL.

We implement the Mutual Reinforcement Learning (abbreviated as **MRL**) by two methods.

Local MRL. The first one is a simple instance-based method. For the same input, suppose $\mathcal{P}_1$, $\mathcal{P}_2$ are generated by $P_g(\theta_1)$ and $P_g(\theta_2)$ respectively. If $R(\mathcal{P}_1) > R(\mathcal{P}_2) * (1 + \delta_2)$ and $\tilde{R}_j(\mathcal{P}_1) > \tilde{R}_j(\mathcal{P}_2)$ for all j, then $P_g(\theta_2)$ uses $\mathcal{P}_1$ instead of $\mathcal{P}_2$ to update itself in Eq.(16) and vice versa. That is, if a learner creates a significantly better poem, then the other learner will learn it. This process gives a generator more high-reward instances and allows it to explore larger space along a more proper direction so as to escape from the local minima.

Global MRL. During the training process, we need to sample poems from the generator, and hence local MRL may cause high variance. Instead of an instance, mutual learning can also be applied on the distribution level. We can pull the distribution of a generator towards that of the other by minimizing KL divergence of them. We detail this method in algorithm 1. The inner thought is that if learner 1 is generally better than learner 2, that is, during the creating history, learner 1 achieves higher average rewards, then learner 2 should directly learn from learner 1, rather than learn the poem itself. This process allows the generator to learn from long-period history and focus on a higher level.

In practice, we combine these two methods by simultaneously communicating high-reward samples and using KL loss, which leads to the best testing rewards (Table 1).

4 Experiments

4.1 Data and Setups

Our corpus consists of three sets: 117,392 Chinese quatrains (CQ), 10,000 Chinese regulated verses (CRV) and 10,000 Chinese iambics (CI). As men-

Models	Bigram Ratio	Jaccard
Base	0.126	0.214
Mem	**0.184**	0.183
MRL	0.181	**0.066**
GT	0.218	0.006
SRL	0.133	0.146
LMRL	0.178	0.085
GMRL	**0.186**	0.075
MRL	0.181	**0.066**

Table 2: Automatic evaluation results of diversity and innovation. The Jaccard values are multiplied by 10 for clearer observation. We expect higher bigram ratio and smaller Jaccard values.

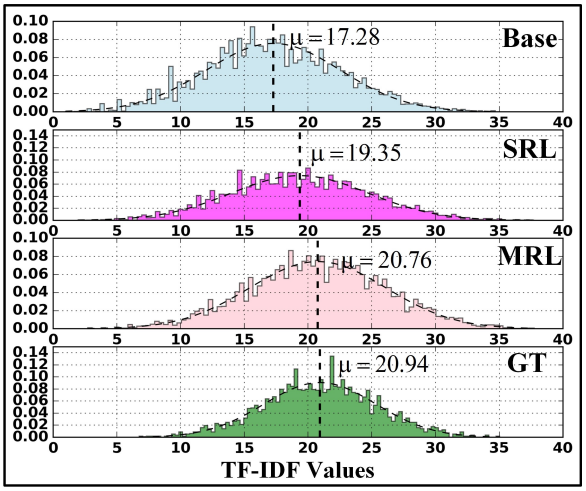

Figure 2: TF-IDF distributions of poems generated by different models. We show real TF-IDF, instead of the estimated $\tilde{R}_3$.

tioned, we experiment on the generation of quatrain which is the most popular genre of Chinese poetry and accounts for the largest part of our corpus. From the three sets, we randomly select 10% for validation. From CQ, we select another 10% for testing. The rest are used for training.

For our model and baseline models, we run TextRank (Mihalcea and Tarau, 2004) on all training sets and then extract four keywords from each quatrain. Then we build four < keyword(s), poem > pairs for each quatrain using 1 to 4 keywords respectively, so as to enable the model to cope with different numbers of keywords.

For the models and rewarders, the sizes of word embedding and hidden state are 256 and 512 respectively. History vector size is 512 and convolution window size $d = 3$. The word embedding is initialized with pre-trained word2vec vectors. We use tanh as the activation function. For other more configurations of the basic model, we directly follow (Yan, 2016).

P_{lm} and $P_{seq2seq}$ are trained with the three sets. We train $F(L_i)$ and $B(\mathcal{P})$ with the CQ, CRV and 120,000 generated poems. There are 9,465 masterpieces in CQ. We use these poems, together with 10,000 generated poems and 10,000 ordinary human-authored poems to train the classifier P_{cl}. For training rewarders, half of the generated poems are sampled and the other half are generated with beam search (beam size 20). For testing, all models generate poems with beam search.

We use Adam (Kingma and Ba, 2015) with shuffled mini-batches. The batch size is 64 for MLE and 32 for DRL. For DRL, we random select batches to fine-tune the basic model. We set $\delta_1 = 0.5$, $\delta_2 = 0.1$, $\delta_3 = 0.001$, $\alpha_1 = 0.25$, $\alpha_2 = 0.31$, $\alpha_3 = 0.14$, $\alpha_4 = 0.30$, $n_s = 4$, and $\beta = 0.7$.

A key point for MRL is to give the two pre-trained generators some diversity, which can be achieved by using different model structures or parameters. Here we simply initialize the generators differently and train one of them for more epoches.

4.2 Models for Comparisons

We compare **MRL**[2] (our model, with both local and global mutual learning), **GT** (ground-truth, namely human-authored poems), **Base** (the basic model described in Section 3.1) and **Mem** (Zhang et al., 2017). The Mem model is the current state-of-the-art model for Chinese quatrain generation, which also achieves the best innovation so far.

4.3 Automatic Evaluation

Some previous models (He et al., 2012; Zhang and Lapata, 2014; Yan, 2016) adopt BLEU and perplexity as automatic evaluation metrics. Nevertheless, as discussed in Section 1, word-level likelihood or n-gram matching will greatly diverge from human evaluation manner. Therefore we dispense with them and automatically evaluate generated poems as follows:

Rewarder Scores. The four rewarder scores are objective and model-irrelevant metrics which approximate corresponding human criteria. They

[2]Due to length limit, we only display the better of the two simultaneously trained generators. Our source code will be available at https://github.com/XiaoyuanYi/MRLPoetry.

Models	Fluency	Coherence	Meaning	Overall Quality
Base	3.28	2.77	2.63	2.58
Mem	3.23	2.88	2.68	2.68
MRL	**4.05****	**3.81****	**3.68****	**3.60****
GT	4.14	4.11^{++}	4.16^{++}	3.97^{++}

Table 3: Human evaluation results. Diacritic ** ($p < 0.01$) indicates MRL significantly outperforms baselines; ++ ($p < 0.01$) indicates GT is significantly better than all models.

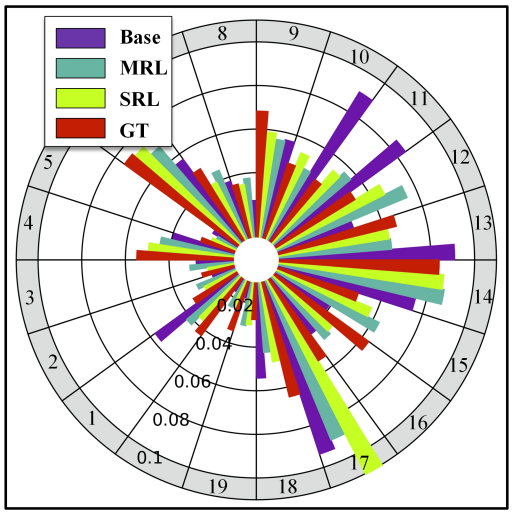

Figure 3: Topic Distributions of different models.

can reflect poetry quality to some extent. As shown in Table 1, on each criterion, GT gets much higher rewards than all these models. Compared to Base, MRL gets closer to GT and achieves 31% improvement on the weighted average reward. Mem outperforms Base on the criteria except for meaningfulness ($\tilde{R}_3$). This is mainly because Mem generates more distinct words (Table 2), but these words tend to concentrate on the high-frequency area, resulting in unsatisfactory TF-IDF reward. We also test different strategies of MRL. With naive single-learner RL, the improvement is limited, only 14%. With mutual RL, the improvement increases to 27%. Combining local MRL and global MRL leads to another 4% improvement. The results demonstrate our explicit optimization (RL) is more effective than the implicit ones and MRL gets higher scores than SRL.

Diversity and Innovation. Poetry is a kind of literature text with high requirements on diversity and innovation. Users don't expect the machine to always generate monotonous poems. We evaluate innovation of generated poems by distinct bigram ratio as (Li et al., 2016b). More novel generated bigrams can somewhat reflect higher innovation. The diversity is measured by bigram-based average Jaccard similarity of each two generated poems. Intuitively, a basic requirement for innovation is that, with different inputs, the generated poems should be different from each other.

As shown in Table 2, Mem gets the highest bigram ratio, close to GT, benefiting from its specially designed structure for innovation. Our MRL achieves 43% improvement over Base, comparable to Mem. We will show later this satisfactory performance may lie in the incorporation of TF-IDF (Figure 2). On Jaccard, MRL gets the best result due to the utilization of MI. MI brings richer context-related information which can enhance diversity as shown in (Li et al., 2016a). In fact, human-authored poems often contain strong diversity of personal emotion and experience. Therefore, despite prominent improvement, there is still a large gap between MRL and GT.

TF-IDF Distribution. As mentioned, the basic model tends to generate common and meaningless words. Consequently, we use TF-IDF as one of the rewards. Figure 2 shows the TF-IDF distributions. As we can see, Base generates poems with lower TF-IDF compared to GT, while MRL pulls the distribution towards that of GT, making the model generate more meaningful words and hence benefiting innovation and diversity.

Topic Distribution. We run LDA (Blei et al., 2003) with 20 topics on the whole corpus and then inference the topic of each generated poem. Figure 3 gives the topic distributions. Poems generated by Base center in a few topics, which again demonstrates the claim: MLE-based models tend to remember the common patterns. In contrast, human-authored poems spread on more topics. After fine-tuning by our MRL method, the topic distribution shows better diversity and balance.

4.4 Human Evaluation

From the testing set, we randomly select 80 sets of keywords to generate poems with these mod-

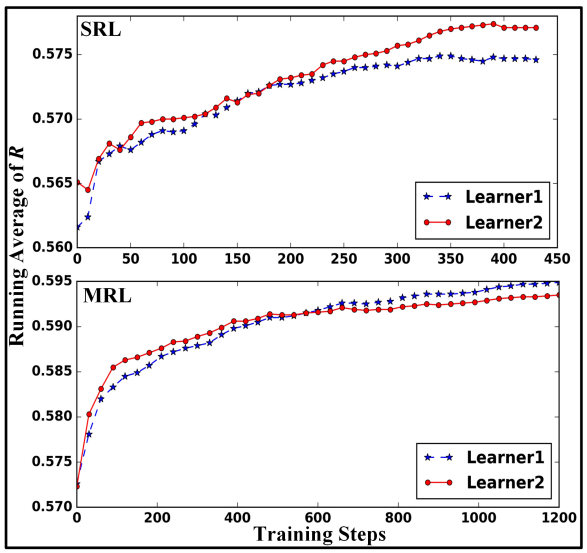

Figure 4: The learning curves of SRL and MRL. Learner (generator) 2 is pre-trained for more epoches to allow some diversity.

els. For GT, we select poems containing the given words. Therefore, we obtain 320 quatrains (80*4). We invite 12 experts on Chinese poetry to evaluate these poems in terms of the four criteria: fluency, coherence, meaningfulness and overall quality and each needs to be scored in a 5-point scale ranging from 1 to 5. Since it's tiring to evaluate all poems for one person, we randomly divide the 12 experts into three groups. Each group evaluates the randomly shuffled 320 poems (80 for each expert). Then for each model, each poem, we get 3 scores on each criterion and we use the average to alleviate individual preference.

Table 3 gives human evaluation results. MRL achieves better results than the other two models. Since fluency is quite easy to be optimized, our method gets close to human-authored poems on Fluency. The biggest gap between MRL and GT lies on Meaning. It's a complex criterion involving the use of words, topic, emotion expression and so on. The utilization of TF-IDF does ameliorate the use of words on diversity and innovation, hence improving Meaningfulness to some extent, but there are still lots to do.

4.5 Further Analyses and Discussions

In this section we give more discussions.

Learning Curve. We show the learning curves of SRL and MRL in Figure 4. As we can see, for SRL, the adequately pre-trained generator 2 always gets higher rewards than the other one during the DRL training process. With the increase of training steps, the gap between their rewards gets larger. After several hundred steps, rewards of the two generators converge.

For MRL, generator 2 gets higher rewards at the beginning, but it is exceeded by generator 1 since generator 1 learns from it and keeps chasing. Finally, the two generators converge to higher rewards compared to SRL.

Case Study. We show some generated poems in Figure 5. The Base model generates two words, 'sunset' and 'moon' in poem (1), which appear together and thus cause the conflict of time. The word 'fishing jetty' is confusing without any necessary explanation in the context. In contrast, poem (2) describes a clearer scene and expresses some emotion: a lonely man takes a boat from morning till night and then falls asleep solitarily.

In poem (3), Mem generates some meaningful words, such as 'phoenix tree', 'wild goose' and 'friend'. However, there isn't any clue to link them together, resulting in poor coherence. On the contrary, things in poem (4) are tightly connected. For example, 'moonlight' is related to 'night'; 'rain', 'frost' and 'dew' are connected with 'cold'.

Poem (5) expresses almost nothing. The first two lines seem to talk about the change of time. But the last two lines are almost unrelated to 'time change'. Poem (6) talks about an old poet, with the description of cheap wine, poem and dream, expressing something about life and time. However, the human-authored poem (7) does much better. It seems to describe a mosquito, but in fact, it's a metaphor of the author himself.

5 Conclusion and Future Work

In this work, we address two substantial problems in automatic poetry generation: lack of diversity, and loss-evaluation mismatch, which are caused by MLE-based neural models. To this end, we directly model the four widely used human evaluation criteria and design corresponding automatic rewarders. We use these explicit rewards to guide gradient update by reinforcement learning. Furthermore, inspired by writing theories, we propose a novel mutual learning schema to further improve the performance. Mimicking the poetry learning activity, we simultaneously train two generators, which will not only be taught by the rewarders but also learn from each other. Experi-

(1) Base	山色侵衣袂|风声入翠微|夕阳明月上|何处是渔矶 The color of mountains blends with the clothing, and the sound of the wind reverberates among green hills. In the light of sunset and bright moon, where is the fishing jetty?
(2) MRL	山色寒侵晓|溪声夜扣舷|小窗明月下|寂寞对床眠 The color of mountains in the morning seems to be cold, and during the night the sound of streams pats the boat. In the moonlight and beside the small window, I fall asleep solitarily.
(3) Mem	梧桐叶下凤凰枝|鸿雁南飞又一时|今夜月明双鬓雪|故人何处是君期 Leaves and branches of the phoenix tree grow tier upon tier. It is already the season when the wild gooses fly to the south. The bright moonlight covers my grey hair. My friend, when will you go back?
(4) MRL	梧桐叶叶已凋残|蟋蟀无声夜漏阑|明月满窗霜露冷|微风吹雨入帘寒 Leaves of the phoenix tree withered already. Crickets are silent and the night is going to end. Moonlight shines through the window, with frost and dew. The breeze blows rain to the curtains, making the house cold.
(5) Mem	三十年前事已非|敢言吾道岂无违|可怜万里归来晚|一片青山眼底飞 Thirty years have passed, and everything has changed. I dare to say that my road is not the same as before. It is a pity to come back late from tens of thousands miles away, and green hills are flying under my eyes.
(6) MRL	老去无心听管弦|一杯浊酒已醺然|诗成桦烛灯前夜|梦到西窗月满船 I don't like listening to music anymore when getting old. Just a cup of cheap wine makes me drunk. In the light of candles, I write a poem at night, and dream that through the west window, I see the boat is filled with moonlight.
(7) GT	白鸟营营夜苦饥|不堪薰燎出窗扉|小虫与我同忧患|口腹驱来敢倦飞 A mosquito is flying around and feeling too hungry at night. It flies out of the window because of the smoke. It is just like me, sharing the same worry: if driven by hunger, we both choose to fly even if we are already exhausted.

Figure 5: Sampled poems generated by different models. Poems between two solid lines are generated with the same input keywords. Some defects are shown in red boxes.

mental results show our method achieves significant improvement both on automatic rewards and human evaluation scores, outperforming the current state-of-the-art model[3].

There are still lots to do. Can we better model the meaningfulness of a whole poem? Can we quantify some other intractable criteria, e.g, poeticness? Besides, we only tried two learners in this work. Would the collaboration of more learners lead to better results? How to design the methods of communication among many generators? We will explore these questions in the future.

Acknowledgments

We would like to thank Cheng Yang, Jiannan Liang, Zhipeng Guo, Huimin Chen and anonymous reviewers for their insightful comments. This research is funded by the National 973 project (No. 2014CB340501). It is also partially supported by the NExT++ project, the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative.

[3]Our method will be incorporated into **Jiuge**, the THUNLP online poetry generation system, https://jiuge.thunlp.cn.

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations*, San Diego, CA.

Albert Bandura. 2001. Social cognitive theory: An agentic perspective. *Annual Review of Psychology*, 52(1):1–26.

David Blei, Andrew Ng, and Michael Jordan. 2003. Latent dirichlet allocation. *Machine Learning Research*, (3):993–1022.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191, Austin, Texas. Association for Computational Linguistics.

Caglar Gulcehre, Sarath Chandar, Kyunghyun Cho, and Yoshua Bengio. 2018. Dynamic neural turing machine with continuous and discrete addressing schemes. *Neural Computation*, 30(4):857–884.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630, Berlin, Germany. Association for Computational Linguistics.

Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1650–1656, Toronto, Canada.

Long Jiang and Ming Zhou. 2008. Generating chinese couplets using a statistical mt approach. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 377–384, Manchester, UK.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 2015 International Conference on Learning Representations*, San Diego, CA.

Robert P. Levy. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, Copenhagen, Denmark. Association for Computational Linguistics.

Hisar Maruli Manurung. 2003. *An Evolutionary Algorithm Approach to Poetry Generation*. Ph.D. thesis, University of Edinburgh.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of the 2017 International Conference on Learning Representations*, Toulon, France.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *Computer Science*.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. *Computer Science*, 40(4):1–5.

Paul Prior. 2006. *Handbook of Writing Research*, chapter A Sociocultural Theory of Writing. Guilford Press.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3776–3784, Phoenix, Arizona.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, and Marc Lanctot. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. 2016a. Chinese song iambics generation with neural attention-based model. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2943–2949, New York, USA.

Zhe Wang, Wei He, Hua Wu nad Haiyang Wu, Wei Li, Haifeng Wang, and Enhong Chen. 2016b. Chinese poetry generation with planning based neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics:Technical Papers*, pages 1051–1060, Osaka, Japan.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.

Rui Yan. 2016. I, poet:automatic poetry composition through recurrent neural networks with iterative polishing schema. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2238–2244, New York, USA.

Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li. 2013. 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.

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2017. Generating chinese classical poems with rnn encoder-decoder. In *Proceedings of the Sixteenth Chinese Computational Linguistics*, pages 211–223, Nanjing, China.

Jiyuan Zhang, Yang Feng, Dong Wang, Yang Wang, Andrew Abel, Shiyue Zhang, and Andi Zhang. 2017. Flexible and creative chinese poetry generation using neural memory. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1364–1373. Association for Computational Linguistics.

Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, Doha, Qatar. Association for Computational Linguistics.