Chinese Poetry Generation with Planning based Neural Network

Zhe Wang[†], **Wei He**[‡], **Hua Wu**[‡], **Haiyang Wu**[‡], **Wei Li**[‡], **Haifeng Wang**[‡], **Enhong Chen**[†]
[†]University of Science and Technology of China, Hefei, China
[‡]Baidu Inc., Beijing, China
xiaose@mail.ustc.edu.cn, cheneh@ustc.edu.cn
{hewei06, wu_hua, wuhaiyang, liwei08, wanghaifeng}@baidu.com

Abstract

Chinese poetry generation is a very challenging task in natural language processing. In this paper, we propose a novel two-stage poetry generating method which first plans the sub-topics of the poem according to the user's writing intent, and then generates each line of the poem sequentially, using a modified recurrent neural network encoder-decoder framework. The proposed planning-based method can ensure that the generated poem is coherent and semantically consistent with the user's intent. A comprehensive evaluation with human judgments demonstrates that our proposed approach outperforms the state-of-the-art poetry generating methods and the poem quality is somehow comparable to human poets.

1 Introduction

The classical Chinese poetry is a great and important heritage of Chinese culture. During the history of more than two thousand years, millions of beautiful poems are written to praise heroic characters, beautiful scenery, love, friendship, etc. There are different kinds of Chinese classical poetry, such as Tang poetry and Song iambics. Each type of poetry has to follow some specific structural, rhythmical and tonal patterns. Table 1 shows an example of quatrain which was one of the most popular genres of poetry in China. The principles of a quatrain include: The poem consists of four lines and each line has five or seven characters; every character has a particular tone, Ping (the level tone) or Ze (the downward tone); the last character of the second and last line in a quatrain must belong to the same rhyme category (Wang, 2002). With such strict restrictions, the well-written quatrain is full of rhythmic beauty.

In recent years, the research of automatic poetry generation has received great attention. Most approaches employ rules or templates (Tosa et al., 2008; Wu et al., 2009; Netzer et al., 2009; Oliveira, 2009; Oliveira, 2012), genetic algorithms (Manurung, 2004; Zhou et al., 2010; Manurung et al., 2012), summarization methods (Yan et al., 2013) and statistical machine translation methods (Jiang and Zhou, 2008; He et al., 2012) to generate poems. More recently, deep learning methods have emerged as a promising discipline, which considers the poetry generation as a sequence-to-sequence generation problem (Zhang and Lapata, 2014; Wang et al., 2016; Yi et al., 2016). These methods usually generate the first line by selecting one line from the dataset of poems according to the user's writing intents (usually a set of keywords), and the other three lines are generated based on the first line and the previous lines. The user's writing intent can only affect the first line, and the rest three lines may have no association with the main topic of the poem, which may lead to semantic inconsistency when generating poems. In addition, topics of poems are usually represented by the words from the collected poems in the training corpus. But as we know, the words used in poems, especially poems written in ancient time, are different from modern languages. As a consequence, the existing methods may fail to generate meaningful poems if a user wants to write a poem for a modern term (e.g., Barack Obama).

In this paper, we propose a novel poetry generating method which generates poems in a two-stage procedure: the contents of poems ("what to say") are first explicitly planned, and then surface realization ("how to say") is conducted. Given a user's writing intent which can be a set of keywords, a sentence or

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1051–1060, Osaka, Japan, December 11-17 2016.

静夜思		Thoughts in a Still Night
床前明月**光**，	(P P Z Z P)	The luminous moonshine before my bed,
疑是地上**霜**。	(* Z Z P P)	Is thought to be the frost fallen on the ground.
举头望明月，	(* Z P P Z)	I lift my head to gaze at the cliff moon,
低头思故**乡**。	(P P Z Z P)	And then bow down to muse on my distant home.

Table 1: An example of Tang poetry. The tone is shown at the end of each line. P represents the level-tone, and Z represents the downward-tone; * indicates that the tone can be either. The rhyming characters are in boldface.

even a document described by natural language, the first step is to determine a sequence of sub-topics for the poem using a poem planning model, with each line represented by a sub-topic. The poem planning model decomposes the user's writing intent into a series of sub-topics, and each sub-topic is related to the main topic and represents an aspect of the writing intent. Then the poem is generated line by line, and each line is generated according to the corresponding sub-topic and the preceding generated lines, using a recurrent neural network based encoder-decoder model (RNN enc-dec). We modify the RNN enc-dec framework to support encoding of both sub-topics and the preceding lines. The planning based mechanism has two advantages compared to the previous methods. First, every line of the generated poem has a closer connection to user's writing intent. Second, the poem planning model can learn from extra knowledge source besides the poem data, such as large-scale web data or knowledge extracted from encyclopedias. As a consequence, it can bridge the modern concepts and the set of words covered by ancient poems. Take the term "Barack Obama" as the example: using the knowledge from encyclopedias, the poem planning model can extend the user's query, Barack Obama, to a series of sub-topics such as outstanding, power, etc., therefore ensuring semantic consistency in the generated poems.

The contribution of this paper is two-fold. First, we propose a planning-based poetry generating framework, which explicitly plans the sub-topic of each line. Second, we use a modified RNN encoder-decoder framework, which supports encoding of both sub-topics and the preceding lines, to generate the poem line by line.

The rest of this paper is organized as follows. Section 2 describes some previous work on poetry generation and compares our work with previous methods. Section 3 describes our planning based poetry generation framework. We introduce the datasets and experimental results in Section 4. Section 5 concludes the paper.

2 Related Work

Poetry generation is a challenging task in NLP. Oliveira (2009; 2012) proposed a Spanish poem generation method based on semantic and grammar templates. Netzer et al. (2009) employed a method based on word association measures. Tosa et al. (2008) and Wu et al. (2009) used a phrase search approach for Japanese poem generation. Greene et al. (2010) applied statistical methods to analyze, generate and translate rhythmic poetry. Colton et al. (2012) described a corpus-based poetry generation system that uses templates to construct poems according to the given constrains. Yan et al. (2013) considered the poetry generation as an optimization problem based on a summarization framework with several constraints. Manurung (2004; 2012) and Zhou et al. (2010) used genetic algorithms for generating poems. An important approach to poem generation is based on statistical machine translation (SMT). Jiang and Zhou (2008) used an SMT-based model in generating Chinese couplets which can be regarded as simplified regulated verses with only two lines. The first line is regarded as the source language and translated into the second line. He et al. (2012) extended this method to generate quatrains by translating the previous line to the next line sequentially.

Recently, deep learning methods achieve great success in poem generation. Zhang and Lapata (2014) proposed a quatrain generation model based on recurrent neural network (RNN). The approach generates the first line from the given keywords with a recurrent neural network language model (RNNLM) (Mikolov et al., 2010) and then the subsequent lines are generated sequentially by accumulating the status of the lines that have been generated so far. Wang et al. (2016) generated the Chinese Song iambics using an end-to-end neural machine translation model. The iambic is generated by translating the pre-

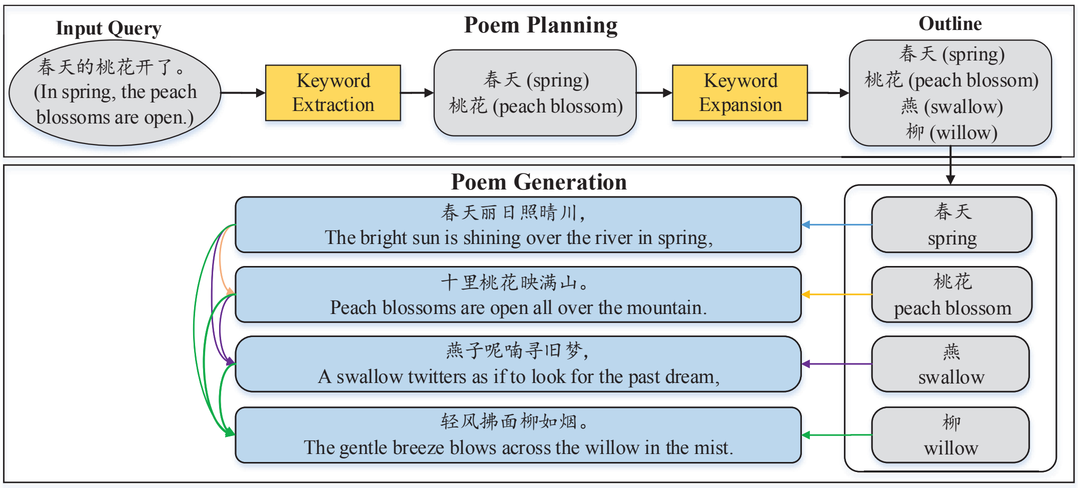

Figure 1: Illustration of the planning based poetry generation framework.

vious line into the next line sequentially. This procedure is similar to SMT, but the semantic relevance between sentences is better. Wang et al. (2016) did not consider the generation of the first line. Therefore, the first line is provided by users and must be a well-written sentence of the poem. Yi et al. (2016) extended this approach to generate Chinese quatrains. The problem of generating the first line is resolved by a separate neural machine translation (NMT) model which takes one keyword as input and translates it into the first line. Marjan Ghazvininejad and Knight (2016) proposed a poetry generation algorithm that first generates the rhyme words related to the given keyword and then generated the whole poem according to the rhyme words with an encoder-decoder model (Sutskever et al., 2014).

Our work differs from the previous methods as follows. First, we don't constrain the user's input. It can be some keywords, phrases, sentences or even documents. The previous methods can only support some keywords or must provide the first line. Second, we use planning-based method to determine the topic of the poem according to the user's input, with each line having one specific sub-topic, which guarantees that the generated poem is coherent and well organized, therefore avoiding the problem of the previous method that only the first line is guaranteed to be related to the user's intent while the next lines may be irrelevant with the intention due to the coherent decay problem (He et al., 2012; Zhang and Lapata, 2014; Wang et al., 2016; Yi et al., 2016). Third, the rhythm or tone in (Zhou et al., 2010; Yan et al., 2013; Zhang and Lapata, 2014; Yi et al., 2016; Marjan Ghazvininejad and Knight, 2016) is controlled by rules or extra structures, while our model can automatically learn constrains from the training corpus. Finally, our poem generation model has a simpler structure compared with those in (Zhang and Lapata, 2014; Yi et al., 2016).

3 Approaches

3.1 Overview

Inspired by the observation that a human poet shall make an outline first before writing a poem, we propose a planning-based poetry generation approach (PPG) that first generates an outline according to the user's writing intent and then generates the poem. Our PPG system takes user's writing intent as input which can be a word, a sentence or a document, and then generates a poem in two stages: Poem Planning and Poem Generation. The two-stage procedure of PPG is illustrated in Figure 1.

Suppose we are writing a poem that consists of N lines with l_i representing the i-th line of the poem. In the Poem Planning stage, the input query is transformed into N keywords $(k_1, k_2, ..., k_N)$, where k_i is the i-th keyword that represents the sub-topic for the i-th line. In the Poem Generation stage, l_i is generated by taking k_i and $l_{1:i-1}$ as input, where $l_{1:i-1}$ is a sequence concatenated by all the lines generated previously, from l_1 to l_{i-1}. Then the poem can be generated sequentially, and each line is generated according to one sub-topic and all the preceding lines.

3.2 Poem Planning

3.2.1 Keyword Extraction

The user's input writing intent can be represented as a sequence of words. There is an assumption in the Poem Planning stage that the number of keywords extracted from the input query Q must be equal to the number of lines N in the poem, which can ensure each line takes just one keyword as the sub-topic. If the user's input query Q is too long, we need to extract the most important N words and keep the original order as the keywords sequence to satisfy the requirement.

We use TextRank algorithm (Mihalcea and Tarau, 2004) to evaluate the importance of words. It is a graph-based ranking algorithm based on PageRank (Brin and Page, 1998). Each candidate word is represented by a vertex in the graph and edges are added between two words according to their co-occurrence; the edge weight is set according to the total count of co-occurrence strength of the two words. The TextRank score $S(V_i)$ is initialized to a default value (e.g. 1.0) and computed iteratively until convergence according to the following equation:

$$S(V_i) = (1 - d) + d \sum_{V_j \in E(V_i)} \frac{w_{ji}}{\sum_{V_k \in E(V_j)} w_{jk}} S(V_j), \tag{1}$$

where w_{ij} is the weight of the edge between node V_j and V_i, $E(V_i)$ is the set of vertices connected with V_i, and d is a damping factor that usually set to 0.85 (Brin and Page, 1998), and the initial score of $S(V_i)$ is set to 1.0.

3.2.2 Keyword Expansion

If the user's input query Q is too short to extract enough keywords, we need to expand some new keywords until the requirement of keywords number is satisfied. We use two different methods for keywords expansion.

RNNLM-based method. We use a Recurrent Neural Network Language Model (RNNLM) (Mikolov et al., 2010) to predict the subsequent keywords according to the preceding sequence of keywords: $k_i = \arg\max_k P(k|k_{1:i-1})$, where k_i is the i-th keyword and $k_{1:i-1}$ is the preceding keywords sequence.

The training of RNNLM needs a training set consisting of keyword sequences extracted from poems, with one keyword representing the sub-topic of one line. We automatically generate the training corpus from the collected poems. Specifically, given a poem consisting of N lines, we first rank the words in each line according to the TextRank scores computed on the poem corpus. Then the word with the highest TextRank score is selected as the keyword for the line. In this way, we can extract a keyword sequence for every poem, and generate a training corpus for the RNNLM based keywords predicting model.

Knowledge-based method. The above RNNLM-based method is only suitable for generating sub-topics for those covering by the collected poems. This method does not work when the user's query contains out-of-domain keywords, for example, a named entity not covered by the training corpus.

To solve this problem, we propose a knowledge-based method that employs extra sources of knowledge to generate sub-topics. The extra knowledge sources can be used include encyclopedias, suggestions of search engines, lexical databases (e.g. WordNet), etc. Given a keyword k_i, the key idea of the method is to find some words that can best describe or interpret k_i. In this paper, we use the encyclopedia entries as the source of knowledge to expand new keywords from k_i. We retrieve those satisfying all the following conditions as candidate keywords: (1) the word is in the window of $[-5, 5]$ around k_i; (2) the part-of-speech of the word is adjective or noun; (3) the word is covered by the vocabulary of the poem corpus. Then the candidate words with the highest TextRank score are selected as the keywords.

3.3 Poem Generation

In the Poem Generation stage, the poem is generated line by line. Each line is generated by taking the keyword specified by the Poem Planning model and all the preceding text as input. This procedure can be

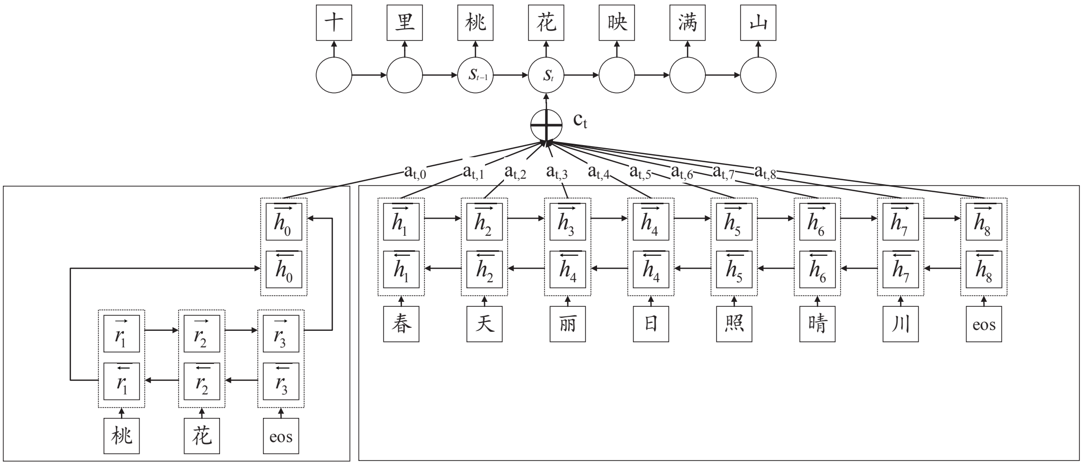
十
里
桃
花
映
满
山
s_{t-1}
s_t
c_t
$a_{t,0}$
$a_{t,1}$
$a_{t,2}$
$a_{t,3}$
$a_{t,4}$
$a_{t,5}$
$a_{t,6}$
$a_{t,7}$
$a_{t,8}$
$\overrightarrow{h_0}$
$\overleftarrow{h_0}$
$\overrightarrow{h_1}$ $\overrightarrow{h_2}$ $\overrightarrow{h_3}$ $\overrightarrow{h_4}$ $\overrightarrow{h_5}$ $\overrightarrow{h_6}$ $\overrightarrow{h_7}$ $\overrightarrow{h_8}$
$\overleftarrow{h_1}$ $\overleftarrow{h_2}$ $\overleftarrow{h_4}$ $\overleftarrow{h_4}$ $\overleftarrow{h_5}$ $\overleftarrow{h_6}$ $\overleftarrow{h_7}$ $\overleftarrow{h_8}$
春 天 丽 日 照 晴 川 eos
$\overrightarrow{r_1}$ $\overrightarrow{r_2}$ $\overrightarrow{r_3}$
$\overleftarrow{r_1}$ $\overleftarrow{r_2}$ $\overleftarrow{r_3}$
桃 花 eos

Figure 2: An illustration of poem generation model.

considered as a sequence-to-sequence mapping problem with a slight difference that the input consists of two different kinds of sequences: the keyword specified by the Poem Planning model and the previously generated text of the poem. We modify the framework of an attention based RNN encoder-decoder (RNN enc-dec) (Bahdanau et al., 2014) to support multiple sequences as input.

Given a keyword $\mathbf{k}$ which has T_k characters, i.e. $\mathbf{k} = \{a_1, a_2, ..., a_{T_k}\}$, and the preceding text $\mathbf{x}$ which has T_x characters, i.e. $\mathbf{x} = \{x_1, x_2, ..., x_{T_x}\}$, we first encode $\mathbf{k}$ into a sequence of hidden states $[r_1 : r_{T_k}]$, and $\mathbf{x}$ into $[h_1 : h_{T_x}]$, with bi-directional Gated Recurrent Unit (GRU) (Cho et al., 2014) models. Then we integrate $[r_1 : r_{T_k}]$ into a vector r_c by concatenating the last forward state and the first backward state of $[r_1 : r_{T_k}]$, where

$$r_c = \begin{bmatrix} \overrightarrow{r_{T_k}} \\ \overleftarrow{r_1} \end{bmatrix}. \tag{2}$$

We set $h_0 = r_c$, then the sequence of vectors $\mathbf{h} = [h_0 : h_{T_x}]$ represents the semantics of both $\mathbf{k}$ and $\mathbf{x}$, as illustrated in Figure 2. Notice that when we are generating the first line, the length of the preceding text is zero, i.e. $T_x = 0$, then the vector sequence $\mathbf{h}$ only contains one vector, i.e. $\mathbf{h} = [h_0]$, therefore, the first line is actually generated from the first keyword.

For the decoder, we use another GRU which maintains an internal status vector s_t, and for each generation step t, the most probable output y_t is generated based on s_t, context vector c_t and previous generated output y_{t-1}. This can be formulated as follows:

$$y_t = \arg\max_y P(y|s_t, c_t, y_{t-1}). \tag{3}$$

After each prediction, s_t is updated by

$$s_t = f(s_{t-1}, c_{t-1}, y_{t-1}). \tag{4}$$

$f(\cdot)$ is an activation function of GRU and c_t is recomputed at each step by the alignment model:

$$c_t = \sum_{j=1}^{T_h} a_{tj} h_j. \tag{5}$$

h_j is the j-th hidden state in the encoder's output. The weight a_{tj} is computed by

$$a_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T_h} \exp(e_{tk})}, \tag{6}$$

where

$$e_{tj} = v_a^T \tanh(W_a s_{t-1} + U_a h_j). \tag{7}$$

e_{tj} is the attention score on h_j at time step t. The probability of the next word y_t can be defined as:

$$P(y_t|y_1, ..., y_{t-1}, \mathbf{x}, \mathbf{k}) = g(s_t, y_{t-1}, c_t), \tag{8}$$

where $g(\cdot)$ is a nonlinear function that outputs the probability of y_t.

The parameters of the poem generation model are trained to maximize the log-likelihood of the training corpus:

$$\arg\max \sum_{n=1}^{N} logP(\mathbf{y_n}|\mathbf{x_n}, \mathbf{k_n}). \tag{9}$$

4 Experiments

4.1 Dataset

In this paper, we focus on the generation of Chinese quatrain which has 4 lines and each line has the same length of 5 or 7 characters. We collected 76,859 quatrains from the Internet and randomly chose 2,000 poems for validation, 2,000 poems for testing, and the rest for training.

All the poems in the training set are first segmented into words using a CRF based word segmentation system. Then we calculate the TextRank score for every word. The word with the highest TextRank score is selected as the keyword for the line. In this way, we can extract a sequence of 4 keywords for every quatrain. From the training corpus of poems, we extracted 72,859 keyword sequences, which is used to train the RNN language model for keyword expansion (see section 3.2.2). For knowledge-based expansion, we use Baidu Baike[1] and Wikipedia as the extra sources of knowledge.

After extracting four keywords from the lines of a quatrain, we generate four triples composed of (the keyword, the preceding text, the current line), for every poem. Take the poem in Table 1 as example, the generated triples are shown in Table 2. All the triples are used for training the RNN enc-dec model proposed in section 3.3.

Keyword	The Preceding Text	Current Line
床	–	床前明月光
霜	床前明月光	疑是地上霜
明月	床前明月光; 疑是地上霜	举头望明月
故乡	床前明月光; 疑是地上霜; 举头望明月	低头思故乡

Table 2: Training triples extracted from the quatrain in Table 1.

4.2 Training

For the proposed attention based RNN enc-dec model, we chose the 6,000 most frequently used characters as the vocabulary for both source and target sides. The word embedding dimensionality is 512 and initialized by word2vec (Mikolov et al., 2013). The recurrent hidden layers of the decoder and two encoders contained 512 hidden units. Parameters of our model were randomly initialized over a uniform distribution with support [-0.08,0.08]. The model was trained with the AdaDelta algorithm (Zeiler, 2012), where the minibatch was set to be 128. The final model is selected according to the perplexity on the validation set.

4.3 Evaluation

4.3.1 Evaluation Metrics

It is well known that accurate evaluation of text generation system is difficult, such as the poetry generation and dialog response generation (Zhang and Lapata, 2014; Schatzmann et al., 2005; Mou et al., 2016). There are thousands of ways to generate an appropriate and relative poem or dialog response given a specific topic, the limited references are impossible to cover all the correct results. Liu et al. (2016) has recently shown that the overlap-based automatic evaluation metrics adapted for dialog responses, such

[1] A collaborative online encyclopedia provided by Chinese search engine Baidu: `http://baike.baidu.com`.

Poeticness	Does the poem follow the rhyme and tone requirements ?
Fluency	Does the poem read smoothly and fluently?
Coherence	Is the poem coherent across lines?
Meaning	Does the poem have a certain meaning and artistic conception?

Table 3: Evaluation standards in human judgement.

Models	**Poeticness**		**Fluency**		**Coherence**		**Meaning**		**Average**	
	5-char	**7-char**	**5-char**	**7-char**	**5-char**	**7-char**	**5-char**	**7-char**	**5-char**	**7-char**
SMT	3.25	3.22	2.81	2.48	3.01	3.16	2.78	2.45	2.96	2.83
RNNLM	2.67	2.55	3.13	3.42	3.21	3.44	2.90	3.08	2.98	3.12
RNNPG	3.85	3.52	3.61	3.02	3.43	3.25	3.22	2.68	3.53	3.12
ANMT	**4.34**	4.04	**4.61**	4.45	4.05	4.01	4.09	4.04	4.27	4.14
PPG	4.11	**4.15**	4.58	**4.56***	**4.29***	**4.49****	**4.46****	**4.51****	**4.36****	**4.43****

Table 4: Human evaluation results of all the systems. Diacritics ** (p <0.01) and * (p <0.05) indicate that our model (PPG) is significantly better than all other systems.

as BLEU and METEOR, have little correlation with human evaluation. Therefore, we carry out a human study to evaluate the poem generation models. Following (He et al., 2012; Yan et al., 2013; Zhang and Lapata, 2014), we use four evaluation standards for human evaluators to judge the poems: “Poeticness”, “Fluency”, “Coherence”, “Meaning”. The detailed illustration can be seen in Table 3. The score of each aspect ranges from 1 to 5 with the higher score the better. Each system generates twenty 5-character quatrains and twenty 7-character quatrains. All the generated poems are evaluated by 5 experts and the rating scores are averaged as the final score.

4.3.2 Baselines

We implemented several poetry generation methods as baselines and employed the same pre-processing method for all the methods.

SMT. A Chinese poetry generation method based on Statistical Machine Translation (He et al., 2012). A poem is generated iteratively by “translating” the previous line into the next line.

RNNLM. A method for generating textual sequences (Graves, 2013), which is proposed by Mikolov et al. (2010). The lines of a poem are concatenated together as a character sequence which is used to train the RNNLM.

RNNPG. In the approach of RNN-based Poem Generator (Zhang and Lapata, 2014), the first line is generated by a standard RNNLM and then all the other lines are generated iteratively based on a context vector encoded from the previous lines.

ANMT. The Attention based Neural Machine Translation method. It considers the problem as a machine translation task, which is similar to the traditional SMT approach. The main difference is that in ANMT, the machine translation system is a standard attention based RNN enc-dec framework (Bahdanau et al., 2014).

4.3.3 Results

The results of the human evaluation are shown in Table 4. We can see that our proposed method, Planning based Poetry Generation (PPG), outperforms all baseline models in average scores. The results are consistent with both settings of 5-character and 7-character poem generations.

The poems generated by SMT are better in Poeticness than RNNLM, which demonstrates that the translation based method can better capture the mapping relation between two adjacent lines. ANMT is a strong baseline which performs better than SMT, RNNLM and RNNPG, but lower than our approach. Both ANMT and PPG use the attention based enc-dec framework. The main difference is that our method defines the sub-topics for each line before generating the poem. The ANMT method just translates the preceding text into the next line. Without the guide of sub-topics, the system tends to generate more general but less meaningful results. In contrast, our approach explicitly considers the keywords, which

	Wrongly Identified MP as HP	Cannot Distinguish	Successfully Identified HP as HP
Normal Group	38.6%	11.3%	50.1%
Expert Group	6.3%	10.0%	83.7%

Table 5: Blind test to distinguish Human-written Poems (HP) from Machine-generated Poems (MP).

秋夕湖上	秋夕湖上
By a Lake at Autumn Sunset	By a Lake at Autumn Sunset
一夜秋凉雨湿衣，	获花风里桂花浮，
A cold autumn rain wetted my clothes last night,	The wind blows reeds with osmanthus flying,
西窗独坐对夕晖。	恨竹生云翠欲流。
And I sit alone by the window and enjoy the sunset.	And the bamboos under clouds are so green as if to flow down.
湖波荡漾千山色，	谁拂半湖新镜面，
With mountain scenery mirrored on the rippling lake,	The misty rain ripples the smooth surface of lake,
山鸟徘徊万籁微。	飞来烟雨暮天愁。
A silence prevails over all except the hovering birds.	And I feel blue at sunset .

Table 6: A pair of poems selected from the blind test. The left one is a machine-generated poem, and the right one is written by Shaoti Ge, a poet lived in the Song Dynasty.

has better controls of the sub-topic for every line. From the results of the human evaluation, it can be seen that the proposed method obtained very close performances in Poeticness and Fluency compared with ANMT but much higher Coherence and Meaning scores, which verified the effectiveness of the sub-topic prediction model.

4.4 Automatic Generation vs. Human Poet

We conducted an interesting evaluation that directly compares our automatic poem generation system with human poets, which is similar to the Turing Test (Turing, 1950). We randomly selected twenty poems from the test set, which are written by ancient Chinese poets. We used the titles of these poems as the input and generated 20 poems by our automatic generation system. Therefore, the machine-generated poems were under the same subject with human-written poems. Then we asked some human evaluators to distinguish the human-written poems from machine-generated ones. We had 40 evaluators in total. All of them were well-educated and had Bachelor or higher degree. Four of them were professional in Chinese literature and were assigned to the Expert Group. The other thirty-six evaluators were assigned to the Normal Group. In the blind test, we showed a pair of poems and their title to the evaluator at each time, and the evaluator was asked to choose from three options: (1) poem A is written by the human; (2) poem B is written by the human; (3) cannot distinguish which one is written by the human.

The evaluation results are shown in Table 5. We can see that 49.9% of the machine-generated poems are wrongly identified as the human-written poems or cannot be distinguished by the normal evaluators. But for expert evaluators, this number drops to 16.3%. We can draw two conclusions from the result: (1) under the standard of normal users, the quality of our machine-generated poems is very close to human poets; (2) but from the view of professional experts, the machine-generated poems still have some obvious shortages comparing to human-written poems. Table 6 gives an example for a pair of poems selected from our blind test.

4.5 Generation Examples

Besides the ancient poems in Table 6, our method can generate poems based any modern terms. Table 7 shows some examples. The title of the left poem in Table 7 is 啤酒 *(beer)*, the keywords given by our poem planning model are 啤酒 *(beer)*, 香醇 *(aroma)*, 清爽 *(cool)* and 醉 *(drunk)*. The title of the right one is a named entity 冰心 *(Xin Bing)*, who was a famous writer. The poem planning system generates three keywords besides 冰心 *(Xin Bing)*: 春水 *(spring river)*, 繁星 *(stars)* and 往事 *(the past)*, which are all related to the writer's works.

啤酒	冰心
Beer	Xin Bing
今宵啤酒两三缸，	一片冰心向月明，
I drink glasses of beer tonight,	I open up my pure heart to the moon,
杯底香醇琥珀光。	千山春水共潮生。
With the bottom of the glass full of aroma and amber light.	With the spring river flowing past mountains.
清爽金风凉透骨，	繁星闪烁天涯路，
Feeling cold as the autumn wind blows,	Although my future is illuminated by stars,
醉看明月挂西窗。	往事萦怀梦里行。
I get drunk and enjoy the moon in sight by the west window.	The past still lingers in my dream.

Table 7: Examples of poems generated from titles of modern concepts.

5 Conclusion and Future Work

In this paper, we proposed a novel two-stage poetry generation method which first explicitly decomposes the user's writing intent into a series of sub-topics, and then generates a poem iteratively using a modified attention based RNN encoder-decoder framework. The modified RNN enc-dec model has two encoders that can encode both the sub-topic and the preceding text. The evaluation by human experts shows that our approach outperforms all the baseline models and the poem quality is somehow comparable to human poets. We have also demonstrated that using encyclopedias as an extra source of knowledge, our approach can expand users' input into appropriate sub-topics for poem generation. In the future, we will investigate more methods for topic planning, such as PLSA, LDA or word2vec. We will also apply our approach to other forms of literary genres e.g. Song iambics, Yuan Qu etc., or poems in other languages.

6 Acknowledgments

This research was supported by the National Basic Research Program of China (973 program No. 2014CB340505), the National Key Research and Development Program of China (Grant No. 2016YFB1000904), the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010) and the Fundamental Research Funds for the Central Universities of China (Grant No. WK2350000001). We would like to thank Xuan Liu, Qi Liu, Tong Xu, Linli Xu, Biao Chang and the anonymous reviewers for their insightful comments and suggestions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30:107–117.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Simon Colton, Jacob Goodwin, and Tony Veale. 2012. Full-face poetry generation. In *ICCC*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *EMNLP*.

Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Long Jiang and Ming Zhou. 2008. Generating chinese couplets using a statistical mt approach. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 377–384. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Ruli Manurung, Graeme Ritchie, and Henry Thompson. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1):43–64.

Hisar Manurung. 2004. An evolutionary algorithm approach to poetry generation.

Yejin Choi Marjan Ghazvininejad, Xing Shi and Kevin Knight. 2016. Generating topical poetry. In *EMNLP*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings the 26th International Conference on Computational Linguistics*.

Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad. 2009. Gaiku: Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 32–39. Association for Computational Linguistics.

Hugo Gonçalo Oliveira. 2009. Automatic generation of poetry: an overview. *Universidade de Coimbra*.

Hugo Gonçalo Oliveira. 2012. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence*, 1:21.

Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *6th SIGdial Workshop on DISCOURSE and DIALOGUE*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Naoko Tosa, Hideto Obara, and Michihiko Minoh. 2008. Hitch haiku: An interactive supporting system for composing haiku poem. In *Entertainment Computing-ICEC 2008*, pages 209–216. Springer.

Alan M Turing. 1950. Computing machinery and intelligence. *Mind*, 59(236):433–460.

Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. 2016. Chinese song iambics generation with neural attention-based model. *CoRR*, abs/1604.06274.

Li Wang. 2002. A summary of rhyming constraints of chinese poems.

Xiaofeng Wu, Naoko Tosa, and Ryohei Nakatsu. 2009. New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system. In *Entertainment Computing–ICEC 2009*, pages 191–196. Springer.

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 *IJCAI*.

Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2016. Generating chinese classical poems with rnn encoder-decoder. *CoRR*, abs/1604.01537.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701.

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, October. Association for Computational Linguistics.

Cheng-Le Zhou, Wei You, and Xiaojun Ding. 2010. Genetic algorithm and its implementation of automatic generation of chinese songci. *Journal of Software*, 21(3):427–437.