Generating Poetry Title Based on Semantic Relevance with Convolutional Neural Network

Z Li¹, K Niu¹ and Z Q He¹

¹Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, No.10 Xitucheng Road, Haidian District, Beijing, China

jaylee1992@foxmail.com

Abstract. Several approaches have been proposed to automatically generate Chinese classical poetry (CCP) in the past few years, but automatically generating the title of CCP is still a difficult problem. The difficulties are mainly reflected in two aspects. First, the words used in CCP are very different from modern Chinese words and there are no valid word segmentation tools. Second, the semantic relevance of characters in CCP not only exists in one sentence but also exists between the same positions of adjacent sentences, which is hard to grasp by the traditional text summarization models. In this paper, we propose an encoder-decoder model for generating the title of CCP. Our model encoder is a convolutional neural network (CNN) with two kinds of filters. To capture the commonly used words in one sentence, one kind of filters covers two characters horizontally at each step. The other covers two characters vertically at each step and can grasp the semantic relevance of characters between adjacent sentences. Experimental results show that our model is better than several other related models and can capture the semantic relevance of CCP more accurately.

1. Introduction

CCP is a great treasure of Chinese culture. According to the structural, rhythmical and tonal patterns, CCP can be divided into many genres. Among all the genres, the quatrain perhaps is the most popular one. There are four lines in the quatrain and each line has five or seven characters. An example of a 5-char quatrain written by Li Bai, a famous poet in Tang Dynasty of China, is shown in figure 1.

Figure 1. An example of a 5-char quatrain.

In recent years, there are several studies using deep learning to write CCP, for examples, the RNNPG model[1] with three neural networks and the encoder-decoder model[2] with a recurrent neural network (RNN) as the encoder and another RNN as the decoder. Given some keywords, those models can generate poems line by line meeting the structural, rhythmical, tonal constraints. However,
it is still difficult to automatically generate the title of CCP. The difficulties mainly come from the differences between the CCP and the modern Chinese articles, which will be described in detail in section 2.

In this paper, we focus on making machines grasp the semantic meaning of the CCP and automatically generate a semantic relevance title. We treat this as an automatic text summarization problem. The research of automatic text summarization is very active in recent years and the encoder-decoder model is widely used recently. Rush, Chopra and Weston[3] first introduce the encoder-decoder model into the automatic text summarization problem and they name their model as ABS. [4] uses a RNN for encoding and attention mechanism is used in the RNN encoder. Many techniques have been proposed based on the encoder-decoder model to improve the summarization result, such as feature-rich encoder[5], minimum risk training[6] and copy mechanism[7]. Our work is based on the encoder-decoder model and we improve the structure of the encoder, which can capture the semantic relevance in poems and is more suitable for the title generation of CCP.

The paper is organized as follows: In section 2, we describe the problem of poem title generation. The details of poem title generation (PTG) model we proposed are introduced in section 3. At last, the evaluation result and conclusion are shown in section 4 and 5 respectively.

2. Problem description

In this paper, we focus on the task of automatically generating the title of CCP. In this task, each input is aquatrain, which is a famous genre of CCP. We first train word2vec[8] model using our dataset to represent each characters as a $k$-dimensional vector. Let $x_{ij} \in \mathbb{R}^k$ be the $k$-dimensional column vector of the $j$-th character in the $i$-th sentence. Let $m$ be the total number of sentences in one quatrain and $n$ be the total number of characters in one sentence, then a quatrain can be represented as a matrix $X \in \mathbb{R}^{mn \times k}$ as below:

$$X = \begin{bmatrix}
    x_{11} & x_{12} & \cdots & x_{1n} \\
    x_{21} & x_{22} & \cdots & x_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{m1} & x_{m2} & \cdots & x_{mn}
\end{bmatrix}$$

(2.1)

Given the content of a poem $X$, the model needs to generate the title $y = [y_1, y_2, \ldots, y_k]$ with maximum probability, where $y_k$ is the $k$-th character in the title. The process can be denoted as below:

$$y = \arg \max_y p(y | X)$$

(2.2)

This task can be taken as an automatic text summarization problem. However, there are several differences between our problem and the summarization problem of modern articles. First, the words in CCP are very different from the words in the modern Chinese and there are no valid word segmentation tools we can use. Just using characters is not enough and we need to make use of the information of words. Second, there are structural, rhythmical, tonal constraints in CCP, so it's better for our model to take advantage of that information. Third, the title of CCP is extremely short (usually less than five characters), so the popular evaluation metrics such as ROUGE[9] are not suitable for this problem. All the differences bring new challenges to the poem title generation problem. For the first and second difficulties, we proposed our poem title generation (PTG) model. We also introduce a new kind of evaluation metric which can reflect the semantic differences between titles to solve the third difficulty.

3. The PTG model
In this section, we first describe the semantic features of CCP, and then the framework for poem title generation is presented. After that, we show the implementation details of our model and how the model can capture the semantic features.

3.1. The semantic features of CCP

The coherence is an important feature of CCP. When writing a CCP, the poet pays attention to not only the semantic relevance of characters in one sentence, but also the semantic relevance of characters between adjacent sentences. So the semantic features of CCP are mainly reflected in two aspects, the intra-sentence semantic relevance and the inter-sentence semantic relevance.

The intra-sentence semantic relevance of characters means some characters usually appear together in one sentence and form words. In CCP, the words usually consist of two characters. In figure 1, “moon” and “light” in the first sentence of the quatrain are an example of the intra-sentence semantic relevance, which form the word “moonlight” together.

The inter-sentence semantic relevance of characters is the antithesis used between two adjacent sentences in the poem. If two characters appear in the same position of two adjacent sentences and the meanings of the two characters are related to each other, then there is inter-sentence semantic relevance between the two characters. In the quatrain in figure 1, the first character “raise” of the third sentence and the first character “stoop” of the fourth sentence are an example of the inter-sentence semantic relevance.

There are difficulties for a model to capture the two kinds of semantic features. The words used in the CCP have many differences with the model Chinese words and there are no valid segmentation tools. So our model should learn the information of words directly from the characters. In addition, the inter-sentence semantic relevance is reflected from the characters in the same position of adjacent sentences, so the input form of our model should not simply be a sequence of characters, which usually is the input form of traditional encoder-decoder models. We will show how our model overcomes those difficulties and captures both the two kinds of semantic features of CCP in the below sections.

3.2. The framework for poem title generation

The poem title generation problem is similar to the automatic text summarization problem. Given an input, which is the quatrain in our problem, the PTG model needs to generate an output which represents the main ideas of the input and is the title of the quatrain here. The entire framework for poem title generation is shown in Figure 2. The title of a CCP can be generated through three steps in our PTG model. Step 1, by looking up the word embedding matrix trained by word2vec model, each character in the poem is transformed into a \( k \)-dimensional vector. Step 2, the model encoder which is a CNN encodes the poem into a fixed length vector, which represents the semantic meaning of the original poem. Step 3, the model decoder which is a RNN generates the title character by character as the above vector being the context.

Our model is similar to the encoder-decoder model raised in [10], but the encoder is tailored CNN for our task. We will describe the details of the CNN encoder in the next section.

3.3. The CNN encoder
The encoder is to encode the input into a fixed length vector. Here, each input is a quatrain $X$ shown in equation (2-1). The quatrain has four sentences and each sentence consists of five or seven characters, so here $m$ is 4 and $n$ is 5 or 7. Because of the special structure of quatrains, we can take quatrains as squared pictures.

![Figure 3. The operations in CNN encoder](image)

Our encoder is a convolutional neural network. The convolutional operation is associated with the filter $W \in \mathbb{R}^{ab}$. To capture the two kinds of semantic features mentioned above, we design two kinds of filters with different shapes. The first kind of filter, which is drawn yellow in figure 3, is the horizontal filter (h-filter) denoted as $W_h$ with $a=1$ and $b=2k$. Another kind of filter, which is drawn red in figure 3, is the vertical filter (v-filter) denoted as $W_v$ with $a=2$ and $b=k$. Then the convolutional operations are induced as below to get an intra-sentence feature $c_{ij}^h$ and an inter-sentence feature $c_{ij}^v$:

$$c_{ij}^h = f \left( \sum_{p=1}^{a} \sum_{q=1}^{b} (W_h \odot X_{i,j,p+q}) + b_h \right)$$  \hspace{1cm} (3.1)$$

$$c_{ij}^v = f \left( \sum_{p=1}^{a} \sum_{q=1}^{b} (W_v \odot X_{i+1,j,p+q}) + b_v \right)$$  \hspace{1cm} (3.2)$$

where, $X_{i,j,p+q} = [x_{i,j}^T x_{i+1,j}^T]$ and $X_{i+1,j} = \left[ x_{i,j}^T x_{i+1,j}^T \right]$ are parts of the input $X$. The mark $\odot$ indicates the element-wise multiplication. $b_h \in \mathbb{R}$ and $b_v \in \mathbb{R}$ are the bias term. $f$ is a non-linear function such as the hyperbolic tangent.

The h-filter covers two adjoining characters in one sentence. After training on the dataset, the h-filter can learn the commonly used words of two characters without word segmentation, so the intra-sentence semantic relevance is captured by our model. The v-filter covers two characters in the same position of two adjacent sentences, so it can capture the inter-sentence semantic relevance. When applying the h-filter to row $i$ of $X$, we can get an intra-sentence feature list denoted by $c_i^h = [c_{i,1}^h \ c_{i,2}^h \ ... \ c_{i,n-1}^h]$.

After the convolutional operation, we use the max-pooling operation. The max-pooling operation is taking the max value of each feature list which is denoted as $c_i^m = \text{max}(c_i^h)$. The aim of max-pooling operation is to capture the most important feature of each feature list. So after the convolutional and max-pooling operations, a specific h-filter can generate an intra-sentence feature list.
\[ c^h = \left[ c^h_1, c^h_2, \ldots, c^h_m \right]. \]
Similarly, a specific v-filter can generate an inter-sentence feature list
\[ c^v = \left[ c^v_1, c^v_2, \ldots, c^v_n \right]. \]

We have described the process of how each filter can generate each list of features. Our CNN model uses multiple h-filters and v-filters to obtain multiple feature lists. We concatenate all feature lists to form one \( k \)-dimensional vector \( c \in \mathbb{R}^k \), which is the representation of the whole quatrain. Our model can capture both the two kinds of semantic features mentioned above, so the vector \( c \) can represent the semantic meaning of the corresponding quatrain.

4. Experiments and results
The training dataset we use is 40047 quatrains with titles. We compare our model with two baseline systems, which are both encoder-decoder models. The first is RNN encoder-decoder model raised in [11]. The second one[4] is similar with the first one, but uses the attention mechanism[12] in the encoder.

4.1. Evaluation metrics
The traditional evaluation metrics for automatic text summarization task such as ROUGE can only reflect the surface similarity between the standard title and the generated title. This is not suited for our poem title generation task. As we mentioned before, the title of CCP usually has less than five characters and directly comparing the surface similarity is ineffective. The semantically similar titles may have low scores under the traditional evaluation metric. In order to reflect the semantic similarity between the standard title and the generated title, we proposed two kinds of evaluation metrics. Let \( G = \{ (X^{(1)}, y^{(1)}), (X^{(2)}, y^{(2)}), \ldots, (X^{(M)}, y^{(M)}) \} \) denotes the standard poem-title pairs in the test dataset and the corresponding generated titles. Here \( y = \{ y_1, y_2, \ldots, y_j, \ldots, y_k \} \) denotes the standard title of the poem and \( y' = \{ y'_1, y'_2, \ldots, y'_j, \ldots, y'_k \} \) denotes the model generated title. \( y_j \in \mathbb{R}^k \) and \( y'_j \in \mathbb{R}^k \) are vectors, which are trained by word2vec model, of \( j \)-th characters in the titles. The vectors trained by the word2vec model can represent the semantic meaning of characters and the distance such as cosine distance between vectors can show the semantic similarity between corresponding characters. The first evaluation metric \( s_1 \) we proposed is calculated as below:

\[ s_1(y_T, y'_T) = s(y_T, y'_T) = \frac{y_T \cdot y'_T}{\| y_T \| \| y'_T \|} \tag{4.1} \]

where the vectors of all characters in the title are averaged to formulate the standard title vector \( y_T \in \mathbb{R}^k \) and the generated title vector \( y'_T \in \mathbb{R}^k \). \( s(\cdot) \) is the cosine similarity function between two vectors.

The \( s_1 \) can reflect the semantic differences between different titles. However, with the fact that the title of CCP is usually very short, directly comparing the title is not very effective, so we propose the second evaluation metric. Instead of comparing the standard titles and generated titles directly, the second evaluation metric compares the titles with the corresponding poems. The title represents the central thought of the poem, so a good title is usually closely related to the poem. Inspired by those ideas, we propose the \( s_2 \) calculated as below:

\[ s_2(y_T, y'_T, x_p) = \frac{s(y'_T, x_p)}{s(y_T, x_p)} \tag{4.2} \]

where the vectors of all characters in the poem are averaged to formulate the poem vector \( x_p \in \mathbb{R}^k \).

4.2. Evaluation results
Our test dataset is 1300 quatrains with titles. After the titles are generated by models, we calculate the $s_1$ score and the $s_2$ score of each generated titles. First, we compare the mean and the median of both $s_1$ and $s_2$ between different models. The results are shown in Table 1.

**Table 1.** Evaluation results.

| Model                                | Mean of $s_1$ (%) | Median of $s_1$ (%) | Mean of $s_2$ (%) | Median of $s_2$ (%) |
|--------------------------------------|------------------|---------------------|------------------|---------------------|
| RNN encoder-decoder                 | 37.7             | 35.1                | 89.4             | 85.1                |
| RNN encoder-decoder + attention      | 38.2             | 35.0                | 91.1             | 86.4                |
| PTG (our model)                      | 40.4             | 37.0                | 95.4             | 91.4                |

From Table 1, we can observe that the PTG we proposed is better than the other two models under both $s_1$ and $s_2$. This indicates the CNN decoder we design is effective and the convolutional neural network is more suitable for the structure of quatrains. It can not only extract features in one line but also can learn features between lines.

Second, we divided the $s_1$ score and the $s_2$ score into five ranges (0.0-0.2, 0.2-0.4, 0.4-0.6, 0.6-0.8, 0.8-1.0) and analyse the percentage of generated titles of different models in each range. The results are shown in Figure 4 and Figure 5 respectively. From Figure 4 and Figure 5, we can observe that the percentage of titles generated in PTG model is lower in low score ranges (0.0-0.4 in Figure 4 and 0.0-0.8 in Figure 5) and higher in high score ranges (0.6-1.0 in Figure 4 and 0.8-1.0 in Figure 5). This indicates that the proposed PTG model is better in generating semantic relevance titles.

### 4.3. Case study

To demonstrate the effectiveness of our model more specifically, we analyze the different titles generated by different models corresponding to the same quatrain. The quatrain with the standard title is shown in Figure 6. It is a poem that praises the beauty of willow.

**Table 2.** The titles generated by different models

| Model                        | Title       |
|------------------------------|-------------|
| RNN encoder-decoder          | The spring  |
| RNN encoder-decoder + attention | The early spring |
| PTG (our model)              | The willow  |

Figure 6. A quatrain with standard title.
The titles generated by different models is shown in table 2. From table 2, we can observe that the title the RNN encoder-decoder model generated is “The spring” and the title the RNN encoder-decoder model with attention mechanism generated is “The early spring”. They both focus on the topic “spring”. Although the season described in the poem is the spring, but that is not the key idea this poem focuses on. In contrast, the PTG model titled this poem “The willow”. Our model gets the main idea of this poem although there is no word “willow” in the content of the poem.

5. Conclusions
In this paper, we propose an encoder-decoder model for automatically generating the title of CCP. The encoder is a CNN and it is to encode the poem into a fixed length vector. The decoder is a RNN that generates the title character by character as the above vector being the input. We also propose an evaluation metric that is more suitable for evaluating our task. Experimental results show that our model is outperformed than the other related models. The future work of this paper may be trying to combine the attention mechanism with our CNN encoder to make further improvement.

Acknowledgments
This work is supported by the National Natural Science Foundation of China (61171099 and 61671080).

References
[1] Zhang X and Lapata M, “Chinese Poetry Generation with Recurrent Neural Networks,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 670–680.
[2] Wang Q, Luo T and Wang D, “Can Machine Generate Traditional Chinese Poetry? A Feigenbaum Test,” Advances in Brain Inspired Cognitive Systems: 8th International Conference, 2016.
[3] Rush A M, Chopra S and Weston J, “A Neural Attention Model for Abstractive Sentence Summarization,” Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 379–389.
[4] Lopyrev K, “Generating news headlines with recurrent neural networks,” Computing Research Repository, 2015.
[5] Nallapati R, Zhou B, Nogueira C, Gulcehre C and Xiang B, “Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond,” In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016, pp. 280–290.
[6] Ayana, Shen S, Zhao Y, Liu Z and Sun M, “Neural headline generation with minimum risk training,” arXiv preprint arXiv:1604.01904, 2016.
[7] Zeng W, Luo W, Fidler S and Urtasun R, “Efficient Summarization with Read-Again and Copy Mechanism,” arXiv preprint arXiv:1611.03382, 2016.
[8] Mikolov T, Chen K, Corrado G and Dean J. “Efficient estimation of word representations in vector space,” International Conference on Learning Representations Workshop, 2013.
[9] Lin C Y, “Rouge: A package for automatic evaluation of summaries,” Text summarization branches out: Proceedings of the ACL-04 workshop, 2014, Vol. 8.
[10] Sutskever I, Vinyals O and Le Q V, “Sequence to sequence learning with neural networks,” Advances in neural information processing systems, 2014.
[11] Cho K, Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1724–1734.
[12] Bahdanau D, Cho K and Bengio Y, “Neural machine translation by jointly learning to align and translate,” International Conference on Learning Representations, 2015.