Chinese Poetry Generator with Extended Input Format

Yujian Zhang, Yanjie Ren, Xiaoxiao Shang

Abstract

Chinese poetry generation is very challenging as it requires both semantically meaningfulness and beauty. Here, we propose a two-stage method to generate Chinese poetry, planning stage then generating stage. In the planning stage, keywords were selected as the sub-topics of the poem according to writing intent. Poem lines were generated sequentially using keywords and previous lines as recurrent neural network framework input. This two-stage method ensures that the generated poem is coherent with extended input.

Introduction

Classical poetry is a significant part of China's cultural heritage. Their popularity manifests itself in many aspects of everyday life. Amongst the many different types of classical Chinese poetry, quatrain(4-line) and regulated verse(8-line) are perhaps the best-known ones. Both types of poem must meet a set of structural, phonological, and semantic requirements, rendering their composition a formidable task left to the very best scholars.

相思
Missing You
红豆生南国，(*ZPPZ)
Red berries born in the warm southland.
春来发几**枝**？(PPZZP)
How many branches flush in the spring?
愿君多采撷，(*PPZZ)
Take home an armful, for my sake,
此物最相**思**。(*ZZPP)
As a symbol of our love.

Table 1: An example of a 5-char *Tang Poem* exhibits one of the most popular tonal patterns. The tone of each character is shown at the end of each line (within parentheses); P and Z are representing for *Ping* and *Ze* tones, respectively; * indicates that the tone is not fixed and can be either. Rhyming characters are shown in boldface.

An example of a quatrain is shown in Table 1. Quatrains have four lines, each line contains five or seven characters. Characters in turn follow specific tonal patterns, within each line and across lines. For instance, the final characters in the even lines must rhyme while there is no constraint for the odd line.

• What is the problem

The words used in poems written in ancient time, are different from modern languages, which means that the existing methods may fail to generate meaningful poems if a user inputs a

modern phrase. For instance, airplane. Also, Current Chinese Poetry generation algorithm only allows fixed form of input.

• Why this is a project related to this class

The generating of the poem requires the machine to understand the meaning of the user input then it could produce the poem which reflects the user intense. This is when NLP kicks in. We need to somehow deduce the user input (words, sentences or even paragraph) into the true intense of the user. For example, the input “peach blossom” in the ancient Chinese poems often means spring, thus the theme of the poem with this input keyword usually is closed to spring.

• Why other approach is no good

Most approaches employ templates or rules.
And most RNN based algorithm usually generate the first line by selecting one line from the dataset of poems according to the user’s writing intents, 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.

• Why you think your approach is better

Our approach allow a wider scope of input, such as longer sentences, a set of words, modern words and refines the meaningfulness of poems generated by machine

• Statement of the problem

Chinese poetry generation is a very challenging task in natural language processing. 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.

Related works

• Related research to solve the problem

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).

• Advantage/disadvantage of those research

Poetry generation is a challenging task in NLP. Oliveira et al. (2009; 2012; 2014) proposed a 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 constraints. 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 previous 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).

• Your solution to solve this problem

We propose a two-stage poetry generating method which first plans the sub-topic 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.

• Where your solution different from others

The input format is extended. Our method would accept not only single sentence or single word, the whole paragraph is also acceptable.

The hidden meaning we extracted from the input would be more comprehensive, thus the final poem would be closer to what the user really wants to express.

• Why your solution is better

The existing method would first convert the input into at most 4 keywords (for Quatrain), then based on the 4 keywords, they would generate corresponding 4 sentences. However, sometimes the user might want to express more feelings or theme, thus 4 keywords would not be enough. The information loss is definitely unavoidable.
Our method would treat all the extracted meaning from the user input as the starting point to generate the poems, thus the information loss would be less. The final result should cover more meanings which the user wants to cover in their poems.

Hypothesis

Our modified generating method could take more kinds of input formats, even a whole paragraph would be acceptable.
The poems generated by our modified method be no worse than the existing method. Also, we anticipate the information loss from the input would be less than the existing method.

Methodology

• How to generate/collect input data
The training data for our model would use Tang Poems, Song Poems, Ming Poems, Qing Poems, Tai Poems, etc.
The input for our generating machine would be random choosing from internet article, keywords; from some main theme from the existing ancient Chinese poems.

• How to solve the problem
We propose a two-stage poetry generating method which first plans the sub-topic 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 planning-based method can ensure that the generated poem is coherent and semantically consistent with the user's intent.

• Language used
Python 3 is used to construct the majority of the architecture since the tools described in the next section have libraries available in this language.

•Tools used
NLTK This library serves as a toolkit for computational linguistics.
Tensorflow was the biggest tool used to train and test the data.
Jieba "Jieba" (Chinese for "to stutter") Chinese text segmentation tool.
Gensim Gensim is specifically designed to handle large text collections, using data streaming and efficient incremental algorithms.
NumPy NumPy is the fundamental package for scientific computing with Python.

• How to generate output
Our model would take input words or sentence and output our desired output (poems).

• How to test against hypotheses

We would test our modified method against the baseline method. We would use the BLEU matrix to evaluate the quality of the poems. We would use human readers to verify the hidden meanings behind the poems, which would be later comparing with the user input hidden meanings. This would be a indicating of the information loss.

Approaches

• Poem Planning and Keyword extraction

The user's writing intent can be represented by a sequence of words.

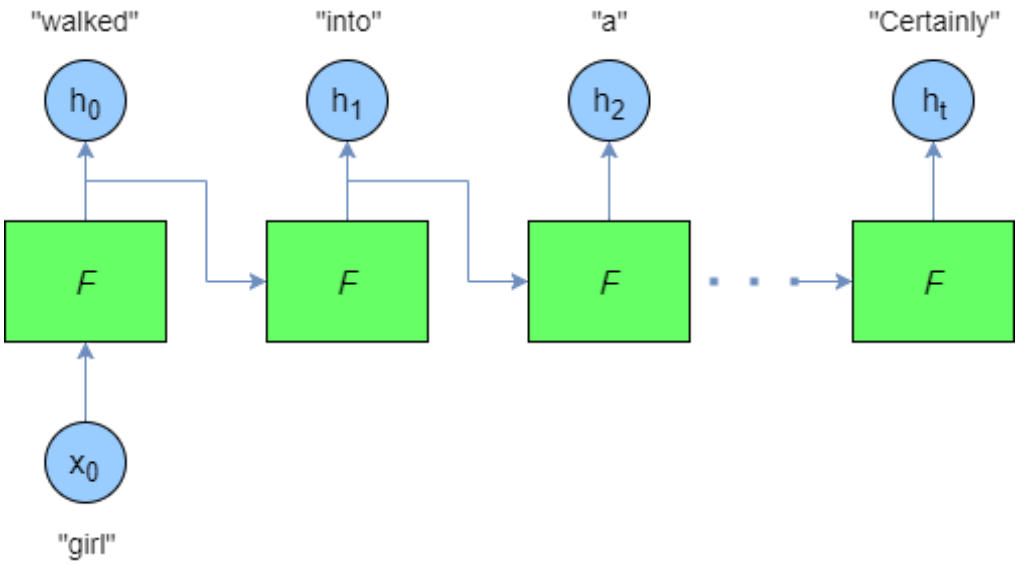
To ensure that each poem line takes exactly one keyword as the sub-topic, the keywords are selected by their relevance to the topic using a TextRank algorithm, which is a graph-based ranking algorithm based on RageRank.

• Poem Generating

In the Poem Generating 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 previous generated text as input. This procedure can be 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 previous generated text of the poem.

• Design document and flow chart

A RNN (Recurrent Neural Network) model is used to train the input data. Tang poem and Song poem were collected from the website as the training dataset. 90% of the total dataset is used to serve as our training data, 5% for validation, and the rest 5% for testing. A One-to-many version of the RNN is also used to generate poem.



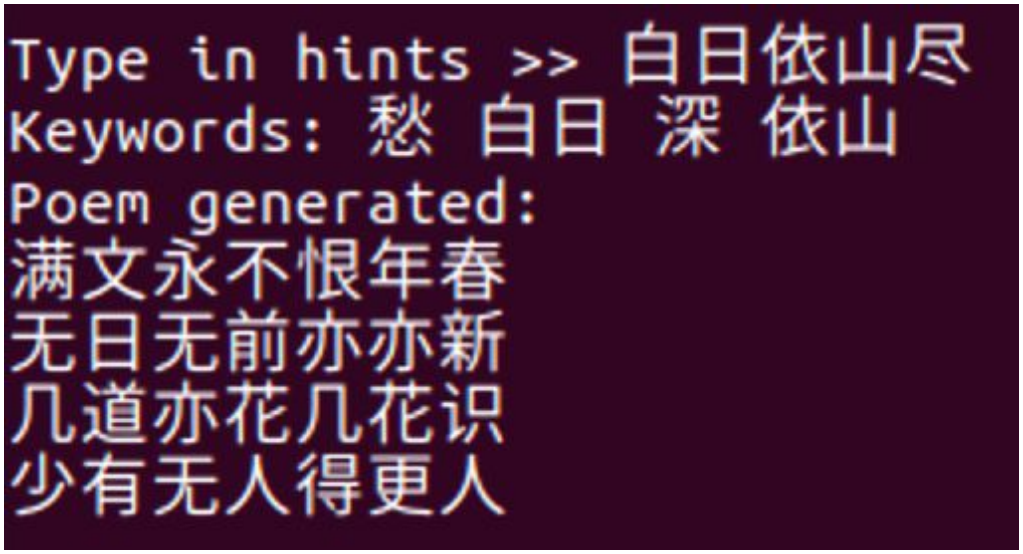
(reference from:
http://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/)

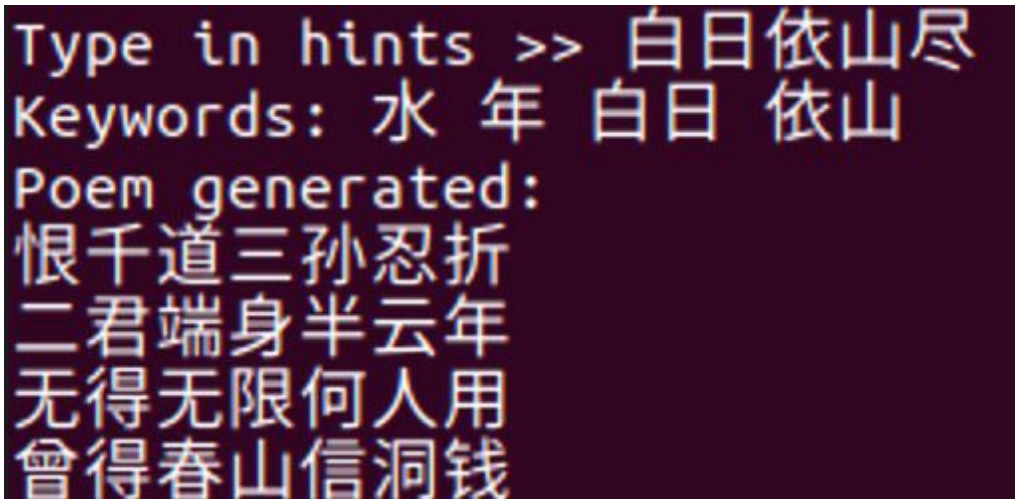
The input would be the keywords we deduces from the user input. Since we would generate the 4-sentence 7-char Quatrain, thus we need to make the input be 4 keywords.
Currently, each of the sentence is using the same RNN model.

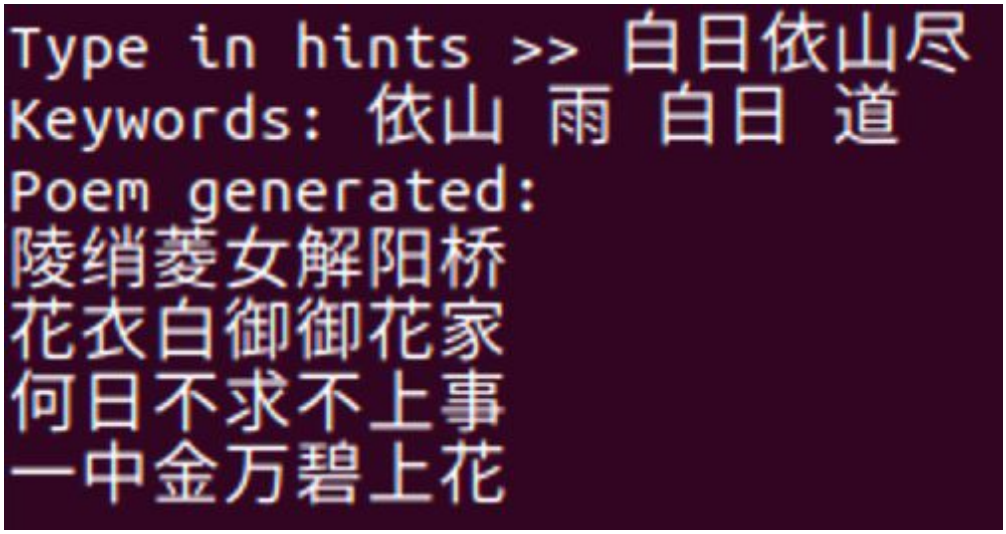
Data analysis and discussion

• sample output

The followings are our sample outputs:
With input sentences: 白日依山尽
we get:

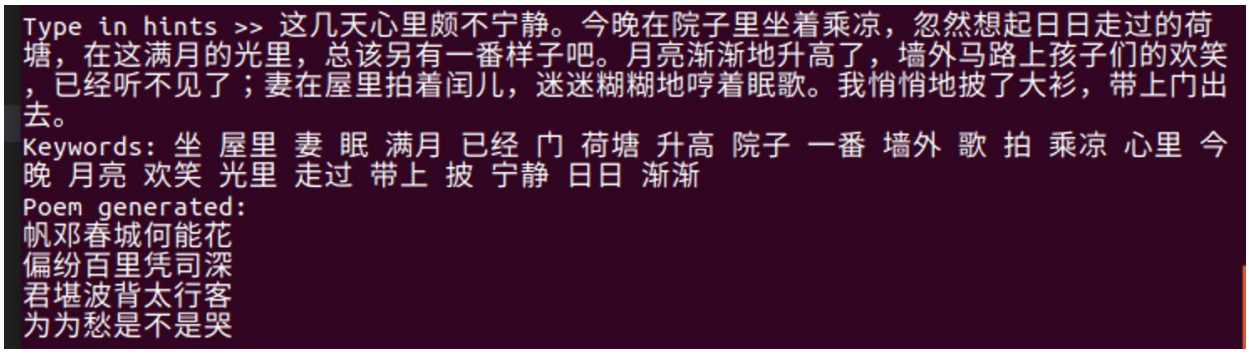


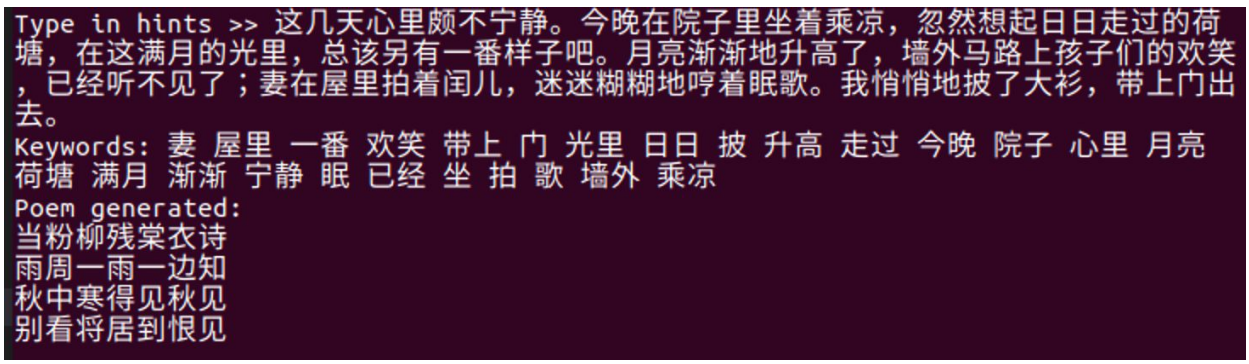




With input sentences: 这几天心里颇不宁静。今晚在院子里坐着乘凉，忽然想起日日走过的荷塘，在这满月的光里，总该另有一番样子吧。月亮渐渐地升高了，墙外马路上孩子们的欢笑，已经听不见了；妻在屋里拍着闰儿，迷迷糊糊地哼着眠歌。我悄悄地披了大衫，带上门出去。

we get:





```
Type in hints >> 这几天心里颇不宁静。今晚在院子里坐着乘凉，忽然想起日日走过的荷
塘，在这满月的光里，总该另有一番样子吧。月亮渐渐地升高了，墙外马路上孩子们的欢笑
，已经听不见了；妻在屋里拍着闰儿，迷迷糊糊地哼着眠歌。我悄悄地披了大衫，带上门出
去。
Keywords: 已经 宁静 歌 乘凉 升高 披 坐 荷塘 今晚 眠 一番 院子 心里 墙外 拍 欢笑
妻 走过 光里 带上 屋里 门 月亮 满月 渐渐 日日
Poem generated:
卦阿军客美门戟
向将见将下将河
江识三三南去去
江郎秋外负官归
```

• Output analysis

Our improvement does make it support extended input format, but could not fix the rhyme functionality bug. Also, the quality of the output poems are not as good as we expected.
The cause might be the training process is not long enough. In the original implementation, they used 1000 epochs of training, however, due to the limitation of our hardware, we only trained the model over at max 200 epochs. Thus the output seems to be meaningless sometimes.

• Compare output against hypothesis

By using the default input format in the generator, our result seems to be identical to the generator using keywords extraction. They both seem not as good as a human poet. This might due to some of the misimplementation or the training process is not getting enough number of epoch.
However, comparing with the keywords extracted by our plan based method, our result somehow could extract the keywords from the input. However, some hidden intense might be lost. For example, in ancient Chinese poem, peach blossom usually means spring, and then spring usually means a positive emotion. Our method could not extract that hidden intense correctly.

• Discussion

Since we are training with a large number of the input dataset (Tang poem, Song poem, etc.), the training would take too much time. Our hardwares are not powerful enough to finish these training and validation in a reasonable time. Thus, we used a lower number of epoch during the training phase. This would reduce the running time but might compromise the fitness of our model. The result sometimes is not readable or meaningful. However, due to we were not training the data for a large number of iteration, we could not tell whether this is because the model is not correct or the training process makes it bad.

Conclusions and Future Work

We have successfully extended the input format for the Chinese poem generator. However, due to the limitation of our knowledge base, our generator sometimes could not get meaningful

poem or the output poem does not match the intention of the user input keywords (sentences). There are still more to be done to make it practically good enough to compare with a human poet. Also, in the ancient Chinese poem structure, each sentence in different positions often represent different emotional meaning. Thus, using 4 different models for each of the 4 positions sentences seems to be a better design approach. The training is not getting too much iteration, we definite need to use a more powerful hardware and more time for the training phase. Thus, we could then distinguish the cause of the bad performance, whether it is from the bad modeling or bad training process. Also, the rhyme functionality seems not working properly, we definitely need to fix it to meet the basic requirement of Chinese poem.

Bibliography

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

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

[Cho et al. 2014] Kyunghyun Cho, Bart Van Merri¨enboer, 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.

[Colton et al. 2012] Simon Colton, Jacob Goodwin, and Tony Veale. 2012. Full-face poetry generation. In ICCC.

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

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

[He et al. 2012] 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.

[Jiang and Zhou 2008] 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.