CSLT TECHNICAL REPORT-20150018 [Thursday 15th October, 2015]

Chinese Song Iambics Generation with Neural Attention-based Model

Qixin Wang[1]
, Tianyi Luo[1]
, Dong Wang[1*]
and Chao Xing[1]

*Correspondence: wangdong99@mails.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

Chinese ancient poems are unique in tonal patterns, rhyme schemes and imaginary beauty. Learning and generating Chinese poems automatically is a charming yet challenging task. Traditional approaches involve various language modeling and machine translation techniques, while recent studies focus on deep learning methods, particularly with the recurrent neural network (RNN) model. Although promising, current RNN-based approaches seem 'forget' too much and quickly during training and inference, which prevents it from dealing with the complex tonal, rhyme and sentiment patterns.

This paper studies the attention-based sequence-to-sequence learning in Chinese poem generation. Specifically, we encode the input key words by a bi-directional Long-Short Term Memory (LSTM) model and then predict the entire poem by an attention-based LSTM model. Both the LSTM and attention-based LSTM are well known in preserving long-term dependency and thus suitable for learning the complex patterns of Chinese poems. Our study shows that the new approach indeed can learn long-distance complex structural and rhythmic patterns very well, and thus can be used to generate Song iambics, a particular type of Chinese classical poems that involve flexible structures and complex rhythmical patterns and have never been successfully generated by machines before.

Keywords: Poem generation; Attention neural network

1 Introduction

The ancient Chinese classical poetry is an important and special cultural heritage with over 2,000 years of history. There are many genres of Chinese classical poetry e.g. Tang poetry, Song iambics, Ming poetry and Qing poetry. Different types of classical Chinese poetry have their own specific structural, rhythmical and tonal patterns. The structural pattern regulates how many lines and how many characters for each line; the rhythmical pattern requires that the last characters of certain lines hold similar vowels; the tonal pattern requires characters in particular positions hold particular tones, i.e., "Ping"(level tone), or "Ze"(downward tone). A good poem should follow all these three pattern regulations (in a descendant order of priority), and has to hold consistent semantic meaning and emotional characteristics. For this reason, it is very difficult to create Chinese classical poetry.

Roughly speaking, Chinese classical poems can be classified into regulated verses and iambics. Regulated verses involve were mostly popular in Tang dynasty (therefore often called 'Tang poem'), and iambics gained the most popularity in Song dynasty (so often called 'Song Iambics', or 'Song Ci' in Chinese). Compared to regulated verses that hold very strict structures (fixed number of lines and fixed number of characters per line) and rhythmical patterns, Song Iambics are more flexible: their structures and rhythmical patterns are not necessarily identical, instead each lyric may follow one of some pre-defined 'tunes'. Song Iambics were originally lyrics of songs that young female artists perform thousands years ago, and the different tunes actually correspond to different melodies. An example of Song Iambics is shown in Table 1. The tune name is '虞美人(Beauty Yu)'. The rhythmical patterns are labelled as bold characters, and the tonal patterns are provided after each line, where 'P' represents level tone and 'Z' represents downward tone.

虞美人
Beauty Yu
春花秋月何时**了**， (*P*ZPPZ)
When will there be no more moon and spring flowers,
往事知多**少**。 (*ZPPZ)
For me who had so many memorable hours?
小楼昨夜又东**风**， (*P*ZZPP)
My attic which last night in vernal wind did stand,
故国不堪回首月明**中**。(***P*ZZPP)
Reminds cruelly of the lost moonlit land.
雕阑玉砌应犹**在**， (*P*ZPPZ)
Carved balustrades and marble steps must still be there,
只是朱颜**改**。 (*ZPPZ)
But rosy faces cannot be as fair.
问君能有几多**愁**， (*P*ZZPP)
If you ask me how much my sorrow has increased,
恰是一江春水向东**流**。(***P*ZZPP)
Just see the overbrimming river flowing east!

Table 1: An example of a Song Iambics with a popular tune 'Beauty Yu'. The rhyming characters are in boldface, and the tonal pattern is shown at the end of each line, where "P" indicates level tone and "Z" indicates downward tone, and '*' indicates that the tone of this character can be either level or downward.

In this paper, we are concerned with automatic generation for Chinese classical poems. Although some research has been conducted, most of the existing approaches focus on regulated verses. For more flexible genres, e.g., Song Iambics, little progress has been achieved. There are many difficulties in Song Iambics generation compared to generating regulated verses, e.g., Tang poems. Firstly Song Iambics are often much longer than Tang poems, which makes it not easy to control the theme (e.g., topics, emotional status) and the semantic flow (i.e., relations between consecutive lines); secondly, there are much more regulations in structures, rhythms and tones, and learning so many regulations is not trivial; thirdly, the remained works that can be used as training data are very limited for each tune, which makes the model training challenging.

In this paper we propose a novel attention-based Long-Short Term Memory (LSTM) model for Song Iambics generation. Specifically, we follow the sequence-to-sequence learning architecture, and use the LSTM model as the encoder and decoder. It is well-known that the LSTM model is capable of learning long-distance information and so can largely alleviate the quick-forgetting problem associated with the traditional RNN [1]. Additionally, the attention-based approach proposed recently by [2] is adopted to provide fine-grained supervision. The attention-based approach generates each character by referring to all the characters of the input, and automatically locates the most relevant character that the generation should be based on. This is a powerful supervision mechanism that enables accurate character-level supervision and thus can model the strict structural regulations and the subtle emotional states of Song Iambics. Since the generation always looks at the input sentences, the entire generation is strongly enforced to follow the same theme defined by the input, which is more important for Song Iambics generation that often suffers from severe 'concept drift' after generating a few sentences.

We employ our approach to generate Song Iambics in two tunes which we can find reasonable training data. The experimental results show that the new approach can generate Song Iambics so well that the subjective test could not tell the difference between human-created and machine-generated iambics.

2 Related Work

Poem automatic generation is a hot but challenging research topic over the past decades. The first category of methods is based on rules and templates. For example, [3, 4] employed a phrases search approach for Japanese poem generation, and [5] proposed an approach based on word association norms. [6, 7] used semantic and grammar templates for Spanish poem generation.

The second approach is based on various genetic algorithms [8, 9, 10]. For example, [10] proposed to use a stochastic search approach to obtain the best matched sentences. The search is based on four standards proposed by [9]: fluency, i.e., a poem must be read fluently; meaningful, i.e., a poem must intentionally convey some conceptual message that is meaningful under some interpretation; and poetic, i.e., a poem must exhibit features that distinguishes it from non-poetic text; coherent, i.e., a poem should discuss some focused topics. These four standards are also used as the evaluation metric in our experiments.

The third approach to poem generation is by various statistical machine translation(SMT) methods. This approach was used by [11] to generate Chinese couplets, which can be regarded as simple regulated verses with only two lines. Their approach formulates the couplet generation as a phrase-based SMT where the second sentence of the couplet is regarded as a translation of the first one. [12] extended this approach to generate Chinese quatrains (four-line Tang poem), where each line of the poem is generated by translation from the previous line.

Another approach to poem generation is based on summarization. For example, [13] proposed a method that retrieves high-ranking candidates of sentences out of a large poem corpus given users' queries. These candidates are segmented into constituent terms which are then grouped into clusters. By re-organizing the terms from different clusters iteratively, sentences that conform the regulation patterns are selected as the generation.

More recently, deep learning methods gain much interest in poem generation. For example, [14] proposed an RNN-based approach to generate Chinese quatrains. By this approach, the first line is generated by a character-based RNN language model [15] given some input keywords, and then the subsequent lines are generated incrementally by accumulating the status of the sentences that have been generated so far. This approach performs well when generating quatrains (a particular type of Tang poems); however the model seems rather complicated (involving four RNN components in total) so is not easy to be extended to generate poems with more complex patterns, e.g., Song Iambics.

Our approach follows the RNN-based approach and thus closely related to the work [14], however there are several important differences that make our proposal novel. Firstly, we use the LSTM rather than the ordinary RNN to obtain long-distance memory; secondly, we use the attention-based framework to enforce theme consistency; thirdly, our model is a simple sequence-to-sequence structure, which is much simpler than the model proposed by [14] and can be easily extended to generate poems with different genres. Particularly, we employ this model to generate Song iambics that involve much more complex structures than regulated and regulations and thus have never been successfully generate by machines.

3 Mehtod

In this section, we first present the attention-based framework, and then describe our implementation for the encoder and decoder models that have been tailored for Chinese classical poem generation.

3.1 Attention-based architecture

The attention-based sequence-to-sequence learning is a general framework where the input sequence is converted to a sequence of latent variables (hidden states) that represent the semantic status at each position of the input, and this sequence is then used to regulate the generation for a new sequence, where at each generation step, the most relevant input is discovered by comparing the 'current' status of the generation and the latent variables of the input. The hidden states are generated by an encoder, and the generation is performed by a decoder.

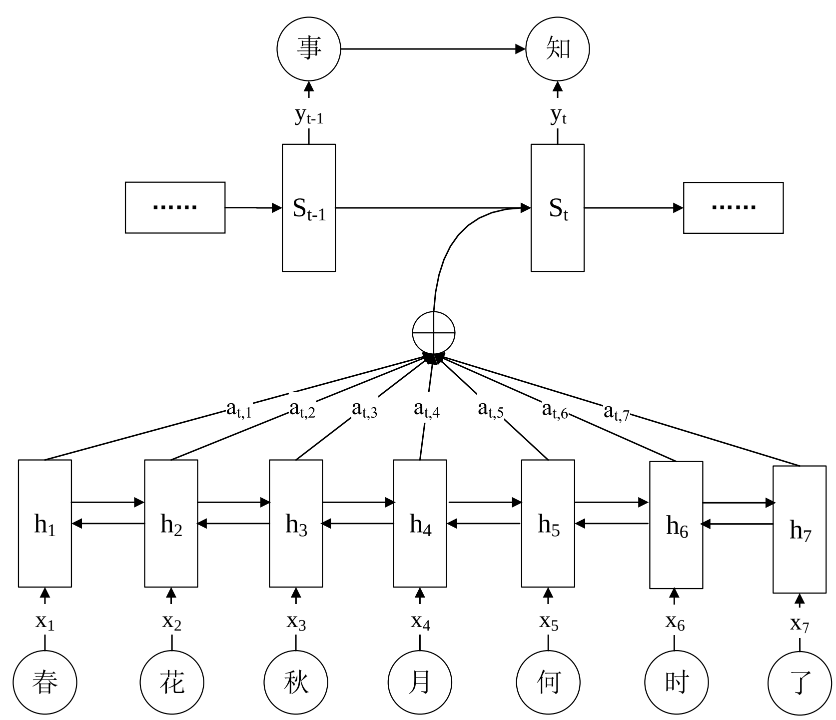

Figure 1: The attention-based sequence-to-sequence learning framework for Song iambics generation. The picture is largely reproduced from [2].

The entire framework is illustrated in Figure 1 when applied to generate Chinese classical poems. The encoder (a bi-directional LSTM as will be discussed shortly) converts the input Chinese characters denoted by $(x_1, x_2, ...)$ into a set of hidden states $(h_1, h_2, ...)$, and the decoder uses this hidden states to generate the poem character by character, denoted by $(y_1, y_2, ...)$. The first step of the generation uses the last hidden state h_T, and at each step t of the subsequent generation, the prediction for the next character y_t is based on the 'current' status s_t of the generation as well as $(h_1, h_2, ...)$ of the original input. An important concept is that the dependence on the hidden states $(h_1, h_2, ...)$ are computed by the relevance between s_t and each of h_i, denoted by $\alpha_{t,i}$. This means that the prediction will pay more attention to the history that is more similar to the current status.

3.2 Implementation details

The encoder of the attention-based framework shown in Figure 1 converts the input characters into a sequence of hidden stats, so that each hidden state encapsulates the semantic meaning of the input a particular position. Typically an RNN model is used as the encoder, however conventional RNNs tend to forget history information quickly in the forward propagation, and so is not suitable to learn long-distance patterns that are often observed in Song Iambics. To improve the memory for complex pattern regulations, we use a bi-directional LSTM model as the encoder. It is well-known that LSTM is capable of learning long-term patterns and using the bi-directional structure further improves the information each hidden state involves.

The decoder of the attention-based framework generates poems depending on the hidden states generated by the encoder. It can be a simple RNN model that maintains an internal status vector s_t, and for each generation step t, a context input c_t is accepted and the most probable output y_t is generated based on s_t. This can be formulated as follows:

$$y_t = argmax_y p(y|s_t, c_t, y_{t-1})$$

where s_t is updated after each prediction, formally written by

$$s_t = f(s_{t-1}, c_{t-1}, y_t)$$

where $f(\cdot)$ is the update function that is determined by the model structure. In this study, we the LSTM model to implement the decoder. As has been discussed, the LSTM model is capable of remaining long-distance information. By this model, the update function involves a couple of linear projections and non-linear activations.

The context vector c_t represent the external input during the generation, and is often used to provide some global information. For example in [14], c_t is derived from all the sentences that have been generated so far. In the attention-based approach, c_t is derived from all the hidden states of the input sequence, e.g., keywords or the first sentence provided by users, formulated as:

$$c_t = \sum_{j=1} \alpha_{tj} h_j,$$

where h_j is the state variable of the j-th input character, and $\alpha_{i,j}$ is the 'attention' paied on h_j, derived by:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$$

$$e_{ij} = \tanh(Ws_{i-1} + Uh_j),$$

where W and U are two matrixes that need to be optimized during model training.

3.3 Global representation

Although the attention-based approach can discover the most interesting part to pay attention and accordingly generate more focused sentences, the global representation h_T derived at the end of the input sequence is still important and deservers special treatment. This is because only at the end of the sentence, the true intention of the sentence can be derived. There are two possible ways to involve h_T: global once (GO) and global always (GA).

The GO method involves h_T as the initial input for the generation. This strategy has been taken in ([16, 17]) for building the intermediate representation for machine translation. The method proposed by [14] belongs to this category. The drawback of the global once method is that the supervision will be gradually lost when generating more and more characters, leading to the problem of concept drift. On the other hand, this is also an advantage, as it allows more creative generation. In contrast, the GA method involves h_T at each step of the generation, by either influence the LSTM gates or the output directly. GA offers stronger supervision and helps consistent generation, but it also constrains the innovation.

It should be highlighted that the attention-based approach collaborates pretty well with the global representation: the global representation focuses on general themes while the attention discovers fine-grained matching. This leads to interesting trade-off between consistency, regulation and innovation in Song Iambics generation, as we will show in the next section.

4 Experiments

4.1 Data and evaluation

We use the corpus provided by [14]. This corpus involves Chinese classical poems of various genres, including Tang poems, Song Poems, Song iambics, Mining Poems, Qing poems, and Tai Poems. There are 284,899 poems in total. The main task focus on Song Iambics generation, for which we train models for two tunes: 'Beauty Yu' and 'Butterflies Love Flowers'. The training data for the two models are 216 and 349 iambics written by the poets in the history, respectively. Another 21 iambics in 'Beauty Yu' and 35 iambics in 'Butterflies Love Flowers' are used for test.

We choose the subjective evaluation and ask for 18 people to evaluate the quality of the generated poems. The participants are all graduate students. Although not experts, they all accepted basic training in reading and analyzing Chinese classical poems for many years. We choose pair-wised evaluation, in which the participants are asked to compare pairs of poems that are generated by different models, where the model generated for a particular poem is unknown and random. The evaluation is based on four metrics: Poeticness (if words are poetry like), Meaning (if the poem is meaningful), Coherence (if the meaning/style is consistent across lines) and Fluency (if a line reads fluently). The participants are asked to choose from three options, i.e., either A or B better, or the two are equal. The averaged proportion of the selections for the three options directly reflects which model is better in a comparison group.

4.2 Experimental results

We choose some iambics that were created by poets in the history and use their first lines as the input to different models to generate the entire iambics. Three models are compared: the GO model which involves global once information only; the GA model which involves global always information; the GO + Att model which involves both global once information and the attention mechanism. We also compare the generation from the GO + Att model with the poems created by poets (denoted by ORG). The experimental results are presented in three group of comparisons: GA vs GO, GO vs GO + Att, and GO + Att vs ORG, as shown in Figure 2, 3 and 4.

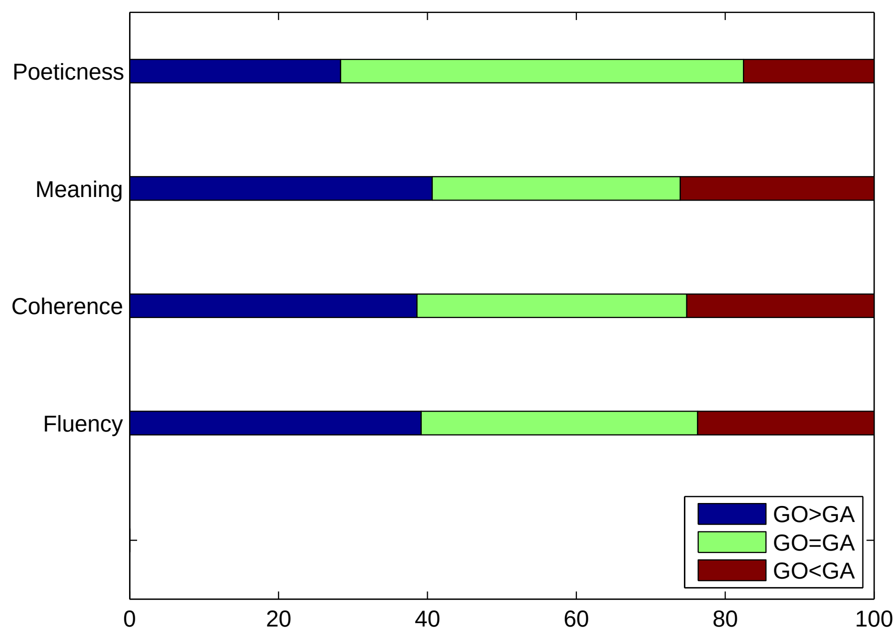

Figure 2: Comparison between GA and GO.

Several interesting observations can be obtained from these results. First of all, from Figure 2 it seems that the GO approach outperforms the GA approach in terms of all the four metrics. This is consistent to the observations that [18] obtained in image recognition, and suggests that for Song iambics that involve relatively flexible structures, a reasonable initial state is sufficient for the decoder to generate reasonable sentences. Too strong regulation from the global information may lead over strong regulation that hurts the innovation for ne expressions and detriments the contribution of the pattern regulations learned by the decoder RNN.

Additionally, Figure 3 shows that by introducing the attention mechanism, the quality of the generated poems is reduced. This is not expected as the attention mechanism tends

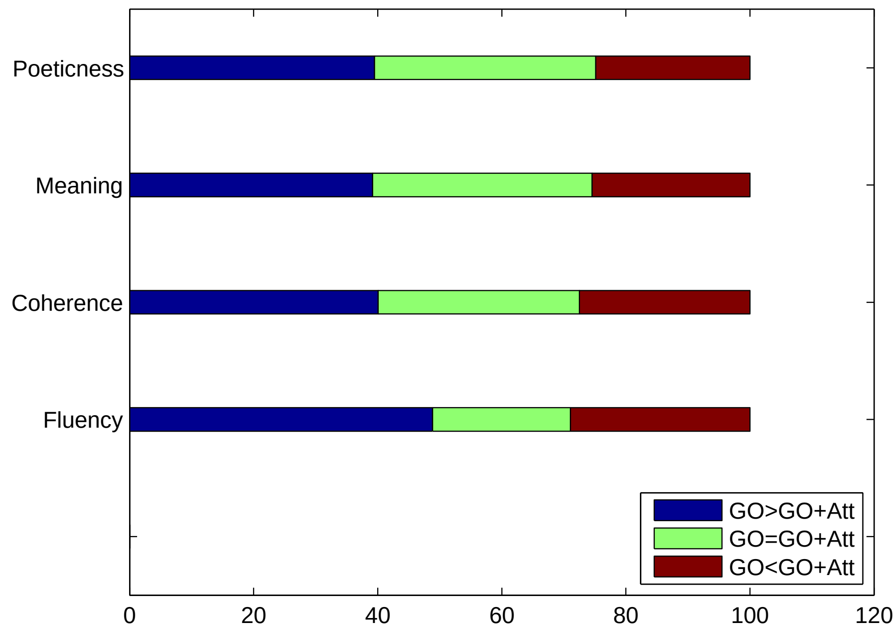

Figure 3: Comparison between GO and GO + Att.

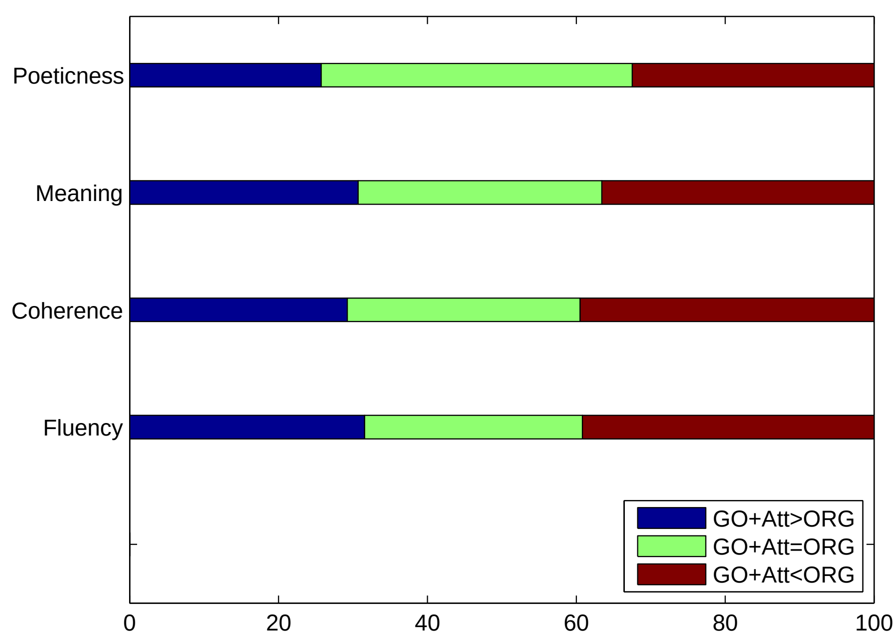

Figure 4: Comparison between GO + Att and ORG (human created poems).

to provide better regulations. We checked the generated poems, and found that that many sentences generated by the GO model were just duplicated from an existing poem in the training corpus, therefore were scored high. This is probably because the training data is relative limited, therefore it is very easy to get the training overfitted. With the attention mechanism, additional fine-grained regulations are introduced based on the input(which is not in the training set), which encourages the generation composing new sentences related to the input, instead of copying one from the training data.

Probably the most striking result is Figure 4. This figure shows the comparison between the poems generated by human and machines (the GO+Att model). The results indicate that in many cases (about 2/3 in proportion), the GO+Att model is so good that people can nearly tell which is generated by a computer and which is generated by a poet. This is very encouraging and it demonstrates the great power of machine learning techniques in modeling human languages. We show two iambics generated by the machine, and more can be found in the attachment.

蝶恋花
一霎秋风惊画扇
柳重深深
断送人遇睡
睡起又将轻别后
无愁风雨将花解
楼上凌波无处处
只解朱弦
又是人相遇
人去何须寻小院
相逢添得人肠断

Table 2: An example Song iambics in tune ‘Butterflies love flowers’.

虞美人
枢庭喜庆生辰到
荚糁纱厨冷
怨星虽少阳关情
不道楼头不负
可怜枝
吴枝长恨为谁惜
更挹东流别
不如今夜后花枝
莫把旧时候后
且徘徊

Table 3: An example Song iambics in tune ‘Beauty Yu’.

5 Conclusion and future work

In this paper we proposed an attention-based sequence-to-sequence learning approach for Chinese Song iambics generation. Our model can prevent the limitations of conventional RNN-based approaches from dealing with the complex tonal, rhythmic and structural patterns. Previous work on poem generation mostly in the quatrain and regulated verse with fixed number of lines and fixed number of character in each line. We remove this limitation by adopting the attention-based framework and so can conduct automation generation for arbitrary length of Chinese poems such as Song iambics. Experimental results show that our approach can learn pattern regulations of Song iambics pretty well, and the subject test can even not tell the difference between the poems generated by machines and people.

Based on the unique advantages of our model in generating poems of arbitrary length, we will extend our approach to other forms of literary genres in Chinese, e.g., Han Fu, Yuan qu, Legend and even novel. Of course, we can and want to extend the approach to other forms of literary genres in other languages, e.g., Shakespearean sonnet and fiction.

Acknowledgement

This research was supported by the National Science Foundation of China (NSFC) under the project No. 61371136, and the MESTDC PhD Foundation Project No. 20130002120011. It was also supported by Sinovoice and Huilan Ltd.

Author details
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Department of Computer Science and Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China.

References

1. Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
2. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
3. Naoko Tosa, Hideto Obara, and Michihiko Minoh, “Hitch haiku: An interactive supporting system for composing haiku poem,” in *Entertainment Computing-ICEC 2008*, pp. 209–216. Springer, 2009.
4. Xiaofeng Wu, Naoko Tosa, and Ryohei Nakatsu, “New hitch haiku: An interactive renku poem composition supporting tool applied for sightseeing navigation system,” in *Entertainment Computing–ICEC 2009*, pp. 191–196. Springer, 2009.
5. Yael Netzer, David Gabay, Yoav Goldberg, and Michael Elhadad, “Gaiku: Generating haiku with word associations norms,” in *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*. Association for Computational Linguistics, 2009, pp. 32–39.
6. H Oliveira, “Automatic generation of poetry: an overview,” *Universidade de Coimbra*, 2009.
7. Hugo Gonçalo Oliveira, “Poetryme: a versatile platform for poetry generation,” in *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*, 2012.
8. Hisar Manurung, “An evolutionary algorithm approach to poetry generation,” 2004.
9. Ruli Manurung, Graeme Ritchie, and Henry Thompson, “Using genetic algorithms to create meaningful poetic text,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 24, no. 1, pp. 43–64, 2012.
10. Cheng-Le Zhou, Wei You, and Xiaojun Ding, “Genetic algorithm and its implementation of automatic generation of chinese songci,” *Journal of Software*, vol. 21, no. 3, pp. 427–437, 2010.
11. Long Jiang and Ming Zhou, “Generating chinese couplets using a statistical mt approach,” in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 377–384.
12. Jing He, Ming Zhou, and Long Jiang, “Generating chinese classical poems with statistical machine translation models,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
13. Rui Yan, Han Jiang, Mirella Lapata, Shou-De Lin, Xueqiang Lv, and Xiaoming Li, “I, poet: automatic chinese poetry composition through a generative summarization framework under constrained optimization,” in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 2197–2203.
14. Xingxing Zhang and Mirella Lapata, “Chinese poetry generation with recurrent neural networks,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 670–680.
15. Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, “Recurrent neural network based language model.,” in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, 2010, pp. 1045–1048.
16. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
17. Ilya Sutskever, Oriol Vinyals, and Quoc VV Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
18. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan, “Show and tell: A neural image caption generator,” *arXiv preprint arXiv:1411.4555*, 2014.