Chinese Poem Generator with Prefix Control

Stanford CS224N Custom Project

Yitong Lu
ICME
Stanford University
ylu236@stanford.edu

Abstract

Motivated by the criticism that poetry-specific models merely accumulate poetic vocabularies and follow rhyme schemes without embodying profound emotion and thought, I aimed to develop a Chinese classic poetry generation model that emphasizes coherence and meaningfulness. I have demonstrated that the sliding window technique is a valid method for enhancing the coherence of generated poems. Additionally, reinforcement learning with a guided auto-grader has proven to be a feasible approach to boosting the meaningfulness of the poetry.

1 Key Information to include

- Mentor: Moussa Doumbouya
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Poetry generation has always been an intriguing topic and a popular test for evaluating the capabilities of large language models (LLMs). Despite its appeal, poetry-specific models often face criticism for being mere accumulations of poetic vocabularies that follow rhyme schemes without capturing the profound emotions, thoughts, and ambitions a poet should express. Furthermore, these models frequently lack basic consistency. Consider the following example of AI-generated poetry:

江上春风吹客衣，扁船载酒入烟霏。桃花落处人家远，燕子衔将柳絮飞。

On the river, the spring breeze blows upon the traveler's clothes, A small boat carries wine into the misty haze. Where peach blossoms fall, houses are far away, Swallows fly, carrying willow catkins in their beaks

This example showcases typical AI-generated poetry, filled with poetic words such as "spring breeze," "small boat," "misty haze," and "peach blossoms." While these elements are evocative, they often fail to coalesce into a coherent and meaningful whole. The aim of this project is, based on a Chinese version of GPT2 training code, build a poetry generator that focus on ensuring the coherence and meaningfulness. In other word, during the process of my experiment, I am trying to tackle the following two questions:

- how to improve the coherence and meaningfulness of a poem.
- how to evaluate the quality of poetry.

3 Related Work

There exist many research in Chinese poetry generation. Jiuge Group, a NLP Lab at Tsinghua University, produced a paper (Yi et al., 2018) talking about evaluate the quality of classical Chinese poem using fluency, coherence and meaningfulness rewarders and used them in mutual reinforcement learning, which give me some enlightenment for my own approach.

Fluency Rewarder $R_1(O)$

$$r(L_i) = \max(|P_{lm}(L_i) - \mu| - \delta_1 * \sigma, 0)$$
$$R_1(O) = \frac{1}{n}\sum_{i=1}^{n} e^{-r(L_i)}$$

Motivate the language model probability of generated lines to fall into a reasonable range.

Coherence Rewarder $R_2(O)$

$$MI(L_{1:i-1}, L_i) = \log P_{seq2seq}(L_i|L_{1:i-1}) - \lambda \log P_{lm}(L_i)$$
$$R_2(O) = \frac{1}{n-1}\sum_{i=2}^{n} MI(L_{1:i-1}, L_i)$$

Use Mutual Information to measure the coherence and expect higher MI.

Meaningfulness Rewarder $R_3(O)$

$$R_3(O) = \frac{1}{n}\sum_{i=1}^{n} F(L_i)$$

$F(L_i)$ is a neural network to estimate the TF-IDF value of a line. TF-IDF is a rough attempt to generate more infrequent/meaningful words

Another approach to evaluate quality is using similarity between different lines of a poem, and tone/rhythm predicted accuracy(Deng et al. 2020). The tone accuracy is the percentage that the tone level is predicted correct to all the generated samples, and the rhythm accuracy is similar about the last character of each poem line that the rhyme is predicted correct.

4 Approach

I started with a open sourced Chinese version training code. I used BERT tokenizer during my several training process as it can handle rare words better and its bidirectional context can be useful for capturing the subtleties of classical poetry. I formatted and cleaned a comprehensive collection of 300,000 classical poetry to training my model. I pre-trained my model with Masked Language Modeling (MLM) method, randomly masking 15% of the character of each poems. MLM can be beneficial for understanding the nuances and meanings in Chinese classical poetry, where context containing symmetrical schemes and appropriate historical reference is crucial. I then fine-tune it with normal autoaggressive next token prediction for generation. Based on the poems generated by my first model, I observed that there usually exist inconsistency between the first half of the poem (line1 and line2) and the second half (line3 and line4), comparing to the relatively good coherence inside each half, possibly thanks to the frequent symmetrical scheme exist in classic poetry. To improve cohenrence, I thus use two type of sliding window method.

- using the first half of poetry to predict the second half
- using the first three lines to predict the last line

The first type of sliding window turns out to be effective in my experiment as after the fine-tuning the model incline to tell a more centralized story with some main idea.
To further improve the coherence and meaningfulness of my model, I thus implemented reinforcement learning with human evaluation and automatic grading. I randomly generated 100 samples of poetry. I graded their qualities focusing on their coherence and meaningfulness. Score were ranged in $[1, 3]$ where 1 represent poor and 3 represent perfect. I then gave these 100 poems and theirs corresponding scores, as well as the general grading rubric to OpenAI, leading it to grade the subsequent generated poems x_i, and the score is the value of reward function $R(x_i)$. I then used simple policy gradient

method for my reinforcement learning.

$$\nabla_\theta J(\theta) \approx \sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(x_i) R(x_i)$$

$\log \pi_\theta(x_i)$ is the log probability of generating poem x_i given current model.
The based line of my project were samples generated by Jiuge system, a poetry specific model developed by Tsinghua NPL Lab. I would make a comparison between the quality our samples judged by the auto-grader below.
My project started from a open sourced Chinese version training code, including the auto-aggressive training code and text generating code. Starting from there, the other part of the code are all revised and written by myself, including: MLM method training code, Sliding window method training code, auto-grader, reinforcement learning and various code of cleaning and reformatting my data into different formats of Json file suitable for different training.

5 Experiments

5.1 Data

I downloaded several online classical poetry collections from Kaggle, Baidu and Jiuge system, and gathered around 30,000 poems in total. Poems were appropriately cleaned and formatted for BERT tokenization. For sliding window fine-tuning, I divided the poems and let the first half (or first three lines) to be the input and the remaining one to be the output.

Trying to fine-tune for better meaningfulness and artistry, I also selected a small collection of around 8,000 poems written by around 50 most outstanding poets in Chinese history.

5.2 Evaluation method

The Human Evaluation is inevitable for poem evaluation, which is more reliable and credible than the automatic evaluation metrics. Some papers trivially use BLEU as the evaluation metrics, but poetry generation is different from translation, and reference is thus hard to choose. Furthermore, similarity to reference is essentially not a qualified standard as creativity and diversity are the cornerstones of poetry generation.
We measure the quality of poetry in four different aspect:

- **Fluency (F)**: The smoothness, readability, and natural flow of the language in the generated poem.

- **Coherence (C)** The logical and thematic consistency within the poem. How well the lines and stanzas connect to form a unified piece.
- **Meaningfulness (M)** The depth, clarity, and significance of the message conveyed by the poem.
- **Aesthetics (A)** The beauty and artistic quality of the poem, including imagery, metaphor, and other poetic devices.

During human evaluation, each score could be integer 1 (poor), 2(average) or 3(perfect). The final score is the average of these four sub-scores above.
Then, during each auto-grading process, I feed the initial 100 samples of poetry and their scores into the judge prompt, as well as the general grading rubric. I also put 20 outstanding poems written by real poets in the history and graded all of them as perfect.
The feedbacks that OpenAi auto-grader gave to me are the evaluation results in my project, as well as the rewarder $R(x_i)$ for my reinforcement learning.

5.3 Experimental details

I trained several different version of models with different combination of methods. My major model configurations was employing the AdamW optimizer with a learning rate of $1.5 * 10^{-4}$ and a warmup phase of 2000 steps to gradually increase the learning rate before decaying it linearly. The training process spanned 5 epochs, with a batch size of 8, and employed gradient accumulation to update the model parameters every gradient accumulation steps. To ensure stability, we applied gradient clipping with a maximum gradient norm of 1.0.
For my final reinforcement learning, after several trials, I trained my model 30 epochs, and each epochs I randomly generated 15 samples of poetry. I picked a constant small learning rate of $1 * 10^5$ because policy gradient methods can be sensitive to high learning rates and my rewarded ranging $[1, 3]$ was not normalized and could be unstable.

5.4 Results

I both tried to directly train my model using normal auto-aggresive next token prediction, and firstly training with MLM method and then fine-tuning with next token prediction. Surprisingly, MLM pre-training did not improve the coherence the of the samples, but a enhancement of fluency is noticed.
I then fine-tuned my model with sliding window methods, an good improvement of coherence is detected. Hoping to improve my model's aesthetics and meaningfulness, I fine-tuned my model with a small collection of 8000 poems written by 50 outstanding poets. The result, however, turns out to be trivial when training with few epochs

and steps, or negative when training with more steps and a symptom of over-fitting is detected.

Finally, I trained my model with reinforcement learning, a good improvement of meaningfulness is observed.

表 1: Model Quality Evaluation

Model	**Fluency**	**Coherence**	**Meaningfulness**	**Aesthetics**
Real poem by LiBai	2.72	2.90	2.87	2.73
Real poem by BaiJuyi	2.93	2.70	2.92	1.95
Jiuge	**2.56**	2.31	2.15	2.37
NTP	2.29	2.23	2.01	2.43
MLM + NTP	2.45	2.21	1.99	2.17
MLM + NTP + SW	2.38	**2.42**	2.35	2.14
MLM + NTP + SW + SC	2.33	1.69	2.10	2.27
MLM + NTP + SW + RL	2.36	2.40	**2.61**	2.32

6 Analysis

I tested the reliability of my auto-grader by feeding it real poems written by eminent poets. The auto-grader's credibility in assessing aesthetics was questionable, as it sometimes gave very low feedback for historical masterpieces. However, its judgment in terms of fluency, coherence, and meaningfulness turned out to be relatively dependable. Of all of my trained models, none of them can surpass my baseline Jiuge in term of fluency, I believe that most of the poetry-specific model implements rigid rhyme patterns of different poetry types which could ensure a better fluency than my purely probabilistic generating model, as my model did not implement any hard-code any rhyme schemes, and my model has a fairly high probability to generate a poem without clear rhythm structure, which is comprehensible for me as among the corpus of my collection, fairly portion of historical poems did not follow any clear poetry structure. MLM methods, however, did noticeably improve the fluency of my model, I would attribute such improvement to that when trying to predict the masked tokens of a line, the model gradually learnt to focus its attention on the tokens of the same position of its paired lines, and thus latently gained more structural knowledge of poetry.

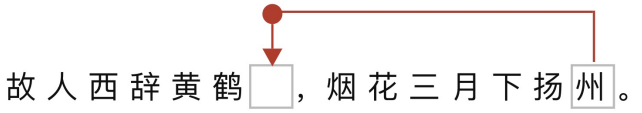

图 1: How paired lines help to predict masked character

Sliding window methods effectively improved the coherence of my samples. It helped my model to reinforce the logical connection between the first two lines to the second two lines. The difference is clear. Blow is one of my numerous examples. The left is generated by model before fine-tuning with SW, right is after fine-tuning.

Poem 1

故人西辞黄鹤楼，烟花三月下扬州。
长安五色日色白，千门万户春色留。

An old friend bids farewell to the Yellow Crane Tower in the west,
In the misty and floral March, he heads down to Yangzhou.
In Chang'an, the five-colored clouds turn pale,
Spring colors linger at every door and window.

Poem 2

故人西辞黄鹤楼，烟花三月下扬州。
莫向江头问春色，断肠多在古荆州。

An old friend bids farewell to the Yellow Crane Tower in the west,
In the misty and floral March, he heads down to Yangzhou.
Do not ask about the spring colors at the river's edge,
Heartbreak often resides in ancient Jingzhou.

The first sentence of both poems comes from a well-known, famous poem. In the left sample (Poem 1), the generated second sentence lacks a consistent theme, indicating a disjointed topic. In contrast, the model fine-tuned with Sliding Window (SW) technique, as seen in Poem 2, demonstrates a deeper understanding of the context provided by the first sentence. It recognizes the author's half-hiding sentiment of missing a departed friend and continues to convey this emotion indirectly in the subsequent lines.

Fine-tuning will small collection of high quality turns out to be a unsuccessful attempt as all aspect of score downgraded.

Reinforcement learning substantially promoted the meaningfulness of samples. During evaluation, I intentionally give poems with a strong central idea a much higher score, such as nostalgia of homeland, pursuit of Buddhism or anguish in own nation conquered. Though I doubted my auto-grader to accurately judge aesthetics, it seemed to have an good understanding of the definition of meaningfulness. After 30 iteration of reinforcement learning, it was noticeable that my samples more frequently conveyed intense emotions or thoughts.

7 Conclusion

In my project, based on a general version of Chinese training code, a reliable poetry generation model is trained without implementing any hard-coded structural rules, and still maintaining a fairly good fluency and rhythm scheme, with MLM tested to be a feasible tools boosting model's understand of poetry structure.

I accomplished to prove sliding window method good technique to improve coherence of samples.

Auto-grader empowered by judgement prompt turns out to be a effective way of grading

meaningfulness of poetry, and thus could be used in reinforcement learning implementing policy gradient method to guarantee depth and intensity of the message conveyed by the generated poem.

8 Ethics Statement

The rise of Large Language Models (LLMs) has imposed significant societal risks to humanity. Knowledge has become one of the cheapest commodities, and creativity has similarly depreciated. The entire canon of Chinese classic poetry contains only around one million works passed down through history. Yet, with a good GPU, my model can generate an equal amount of poetry in just a few days.

The field of poetry is already in decline, with few professional poets able to sustain their livelihoods. This is a stark contrast to the golden age of the 1980s, a nostalgic era when poets were respected, flourishing, and believed to be the flag bearers of idealism. In the age of LLMs, the situation for poets is likely to deteriorate even further.

Even more concerning, my model allows users to input a four-character prefix, for example, from any famous historic poem, and generate a vast amount of completed poems with this prefix, some of which are of good quality. While it may not be strictly plagiarism, such practices undeniably challenge the value of creativity.

As quoted in the biography of Samuel Johnson, "Sir, hell is paved with good intentions." This sentiment encapsulates the ethical concerns not only of my project but also of this era. As of mitigation strategies, we can only embrace the rise of LLM, promote collaboration between AI developers and creative professionals, and encourage the use of AI as a tool to enhance human creativity rather than replace it. This can involve AI assisting poets in brainstorming or enhancing their work without taking full credit.

References

Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. 2018. Automatic Poetry Generation with Mutual Reinforcement Learning. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3143 – 3153, Brussels, Belgium. Association for Computational Linguistics.*

Liming Deng, Jie Wang, Hangming Liang, Hui Chen, Zhiqiang Xie, Bojin Zhuang, Shaojun Wang and Jing Xiao. 2020. An Iterative Polishing Framework based on Quality Aware Masked Language Model for Chinese Poetry Generation. *In Proceedings of AAAI 2020.*

Huimin Chen, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang and Zhipeng Guo. 2019. Sentiment-Controllable Chinese Poetry Generation. *In Proceedings of IJCAI 2019.*

Zeyao Du. 2019. GPT2-Chinese: Tools for training GPT2 model in Chinese language. `https://github.com/Morizeyao/GPT2-Chinese`,

Jackey Gao. 2023. Chinese-poetry collection database. https://github.com/chinese-poetry/chinese-poetry