Controllable Generation from Pre-trained Language Models via Inverse Prompting

Xu Zou[†*], Da Yin[†*], Qingyang Zhong[†*], Hongxia Yang[◇], Zhilin Yang[‡*],Jie Tang[†*]
[†] Department of Computer Science and Technology, Tsinghua University
[*] Beijing Academy of Artificial Intelligence, [‡] Recurrent AI, Ltd, [◇] Alibaba Inc.
{zoux18,yd18,zqy20}@mails.tsinghua.edu.cn
yang.yhx@alibaba-inc.com,kimi_yang@rcrai.com,jietang@tsinghua.edu.cn

ABSTRACT

Large-scale pre-trained language models have demonstrated strong capabilities of generating realistic texts. However, it remains challenging to control the generation results. Previous approaches such as prompting are far from sufficient, and lack of controllability limits the usage of language models. To tackle this challenge, we propose an innovative method, inverse prompting, to better control text generation. The core idea of inverse prompting is to use generated text to inversely predict the prompt during beam search, which enhances the relevance between the prompt and the generated text and thus improves controllability. Empirically, we pre-train a large-scale Chinese language model to perform a systematic study using human evaluation on the tasks of open-domain poem generation and open-domain long-form question answering. Results demonstrate that our proposed method substantially outperforms the baselines and that our generation quality is close to human performance on some of the tasks.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**.

KEYWORDS

Language Modeling, Machine Question Answering, Poem Generation, Controllable Generation, Beam Search

ACM Reference Format:
Xu Zou[†*], Da Yin[†*], Qingyang Zhong[†*], Hongxia Yang[◇], Zhilin Yang[‡*],Jie Tang[†*] . 2021. Controllable Generation from Pre-trained Language Models via Inverse Prompting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467418

1 INTRODUCTION

The field of text generation has made tremendous progress in recent years. Large-scale auto-regressive Transformer models [23] optimized with maximum likelihood estimation have shown the ability of generating realistic text [2, 4, 15]. For real-world applications of text generation such as essay writing and story generation, it is essential for the users to be able to control the generation results. One of the most common approaches is to use prompting; i.e., a user shall manually write a few sentences to serve as the prompt and the language model generates the subsequent tokens given the prompt. For example, a user might input "this is a sad story about a disease named COVID-19" as a prompt to expect the generation of a COVID-19 story.

However, prompting is far from sufficient for controllable text generation. It is not uncommon for a language model to deviate the generation process from the original prompt and start generating text of unrelated topics. Table 1 shows an example of how language models fail to maintain the coherence between the prompt and the generated text. In the example, the language model is asked to answer the question "which moment do you want to live in forever". Using conventional prompting generates a story that deviates a lot from the prompt; i.e., most of the generated content is irrelevant to the question.

To tackle this challenge, we propose a novel method, inverse prompting, to refine the process of text generation from pre-trained language models. Inverse prompting can be decoupled into three steps. First, given a piece of generated text, an inverse prompt is constructed using the generated text. Second, the conditional likelihood of the original prompt given the inverse prompt is computed based on the pre-trained language model. Third, the conditional likelihood is used as a score in beam search for selecting the best generation candidates. As a result, inverse prompting ensures that the pre-trained language model predicts the prompt given the generated text with high likelihood, which encourages the relevance of the generated text to the prompt. For example, in the case of question answering, a language model generates an answer given the question; during beam search, we use the same language model to compute the likelihood of the question given the generated answer for candidate selection. In the example of Table 1, text generated by inverse prompting describes a beautiful moment of high school times which is closely related to the question. As a result, inverse prompting achieves an average score of 8.60 out of 10 under human evaluation on this example, compared to the baseline's 5.40.

To systematically evaluate the performance of inverse prompting, we conduct human evaluation on the tasks of open-domain long-form question answering and open-domain traditional Chinese poem generation. We pre-train a Chinese language model to serve as the base model in our experiments. The task of long-form question answering is similar to answering questions on Quora

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467418

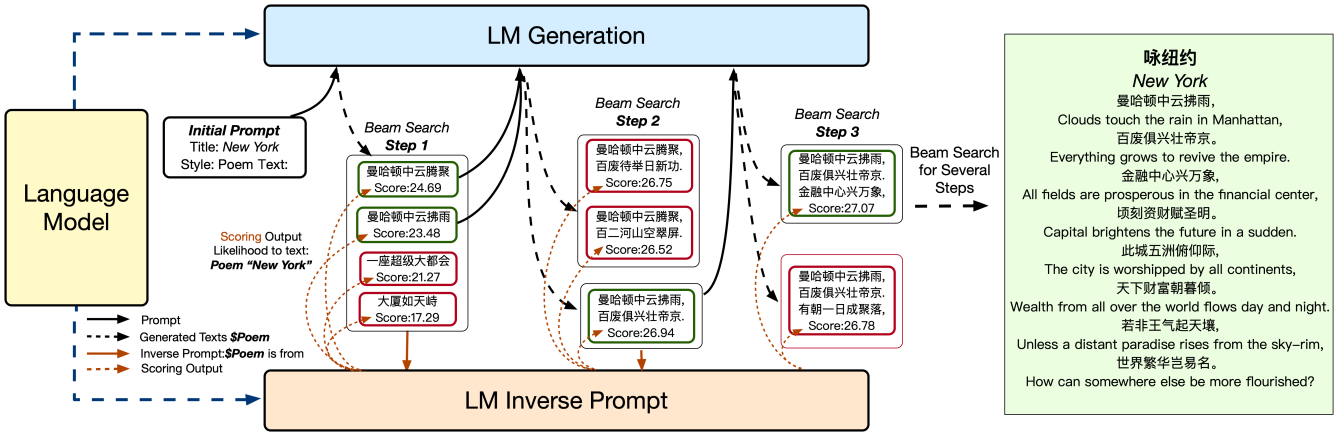

Figure 1: The generation process of open-domain traditional Chinese poems under inverse prompting. Using title *New York* (《咏纽约》) as an example.

Table 1: An example showing how the prompting baseline model may fail to maintain relevance in generated text, and how inverse prompting alleiates this issue. The relevance and overall scores were obtained from human evaluation.

Question	什么瞬间让你想留住那一刻? Which moment did you want to live in forever?
Prompting Baseline (relevance:2.83 overall:5.40)	我和男朋友在一起两年。我去国外留学, 他留在国内。一人在国外, 寂寞心酸都是真的。念书的时候, 觉得常在一起也没什么, 两个人一起努力, 吵吵闹闹就过来了。我一直这样想, 以至于到了高中, 到了最后几个月, 我都觉得他就是我行走的小弟儿。其实最快乐的时候, 就是一起在外头的“孤儿院”里。 I have been in a relationship with my boyfriend for two years. I went to study abroad and he stayed in our home country. Being alone in a foreign country, I felt really lonely and sad. When I was in school, I felt that it was ok to be together often. We worked hard together and we also quarrelled with each other. I kept thinking in this way. In the last few months of our time at the high school, I felt that he was my walking little brother. In fact, the happiest moment was when we were together in the “orphanage”.
Inverse Prompting (relevance:4.53 overall:8.60)	一瞬间想到高二的时候吧, 那天下午在操场晒了一中午太阳, 和同学一起在楼下打球, 下午的夕阳徐徐洒下来, 我们三个人一起站在走廊上, 倚着栏杆可以面对着夕阳。一直想留下那一瞬间, 可惜我以后再也没遇到过那样的时光。就是想留一下那一刻吧。 In an instant I recall the second year of high school. That afternoon, I took a sunbath on the playground for a long time. I then played basketball downstairs with my classmates. As the sun set slowly, the three of us stood together on the corridor, leaning on the railing to face the sunset. I've always wanted to live in that moment forever, but unfortunately I never encountered a similar moment again. I just want to live in that moment.

or Zhihu. On this task, we show that inverse prompting achieves much higher scores in all aspects than the prompting baseline and the previous state-of-the-art Chinese language model CPM [27]. The task of traditional Chinese poem generation targets generating poems of traditional form but with contemporary topics including rocket science, relativity, or artificial intelligence, which tests the generalization ability of different approaches. Figure 1 illsutrates an example of traditional Chinese poem generation under the title *New York*. It combines contemporary notions of New York like Manhattan and the financial center with a traditional form and traditional poetic imagery of cloud and rain. On this task, human expert evaluation demonstrates that inverse prompting performs significantly better than the beam search baseline and comparable to Jiuge [28], a well-known state-of-the-art system for traditional Chinese poem generation. When we combine inverse prompting with self training, i.e., fine-tuning the model with self-generated poems, our system outperforms Jiuge under human evaluation by a large margin. Our results of human evaluation demonstrate that inverse prompting improves the controllability and quality of text generation significantly and achieves close-to-human results.

2 RELATED WORK

2.1 Pre-training and Language Modeling

Language modeling has been widely used as an objective for pre-training and demonstrates strong generalization abilities. Originating from word embedding methods such as word2vec [13] and GloVe [14], pre-training methods have displayed an increased level of importance in the field of natural language processing [4, 7, 11]. These models are more general and require less domain-specific data to achieve strong performance.

Most of the well-known pre-training models are auto-regressive language models. Generative pre-training (GPT) [2, 15, 16] and Transformer-XL [4] achieve substantial improvements in terms of perplexity and also improve generation quality. These approaches have also been adapted to different languages [6, 27].

Although realistic text can now be generated automatically by large-scale pre-trained language models, it is challenging but essential for users to be able to control the generated results. Prompting [4, 15] has been widely used but is rather limited in controlling the generation results. CTRL [9] proposes the usage of control codes to provide conditions for a language model. PPLM [5] performs backpropagation during test time to adjust generation to maximize the scores given by attribute models. Different from CTRL, inverse prompting does not rely on modification of pre-training paradigms or human-designed attributes. Compared with

PPLM, inverse prompting does not require any gradient updates to the original model and is free of any additional attribute models.

The idea of using dual process to strengthen the quality of AI generation has long been researched. The dual property suggests that outputs and inputs are inputs and outputs under an inverse perspective. [24] introduces dual learning for the task of machine translation. The method uses multiple different models to form a translation loop and expects the context to remain unchanged after passing through the loop. CycleGAN [3] and VAE [1] also share the similar idea of inverse reconstruction in their applications. Different from these works which use different forward and inverse models, in this paper, we exploit the existence of inverse format in natural languages and use the same language model for prompting and inverse prompting.

2.2 Open-Domain Long-Form Question-Answering

Question answering is a well-studied problem in artificial intelligence [21]. There are various paradigms of question answering. Short-form question answering focuses on using a short phrase or sentence to answer the question [17, 25]. On the other hand, long-form question answering targets generating multiple sentences or paragraphs to answer a question in a more comprehensive way. Online question answering platforms such as Quora and Zhihu can be viewed as good examples of long-form question answering. While short-form question answering is easier to evaluate and more widely studied, we are interested in investigating the ability of open-domain long-form question answering using text generation models. It is challenging to figure out a metric to evaluate the qualities of long-form question answering, therefore we employ human evaluation in our experiments.

2.3 Traditional Chinese Poem Generation

Traditional Chinese poetry is an important genre of Chinese literature with a history of dozens of centuries [10]. A few years ago, researchers experimented with generating traditional Chinese poems using statistical machine learning methods [8]. Later, Jiuge [26, 28] advanced traditional Chinese poem generation to a new level. As the well-recognized state of the art for open-domain Chinese poem generation, Jiuge is able to generate multiple forms of poems under any given titles, keywords or even images. Despite its ability to handle arbitrary open-domain inputs, Jiuge performs well on domain-specific contexts such as giant deserts or iron horses but does not generalize well to contemporary notions such as Donald Trump, quantum computation, and Europe. Different from Jiuge, we employ a large-scale language model pretrained on a general-purpose corpus and leverage inverse prompting to enhance its generation quality.

3 METHODOLOGY

In this section, we discuss the proposed inverse prompting method. The problem of text generation is modeled as generating text sequence of tokens c_g given the prompt sequence of tokens c_p.

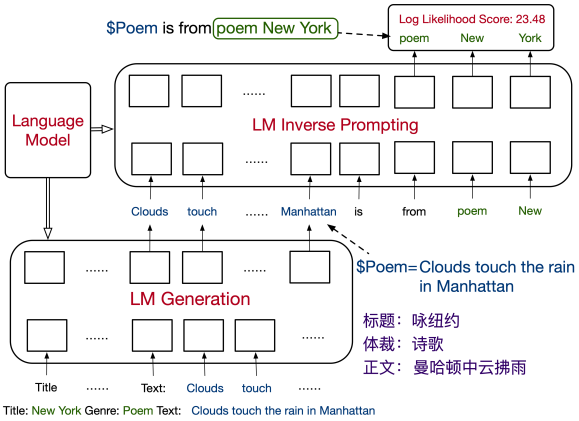

Figure 2: Language model generation and language model inverse prompting scoring for generating a poem sentence.

3.1 Baseline: Prompting and Beam Search

Given a language model with probability distribution p, a simple and widely-used approach is to generate text by maximizing the conditional probability $p(c_g|c_p)$. This is usually achieved with beam search [12]. With a beam size of n, beam search keeps the top-n sequences during decoding time according to a beam scoring function $f(\cdot)$. An illustration is shown in Algorithm 1. The baseline method uses the log likelihood to define the scoring function, i.e., $f(c_g|c_p) = \log p(c_g|c_p)$

Algorithm 1: Beam search. Inverse prompting follows the beam search framework with a novel scoring function f being the inverse log likelihood of the prompt given the generated text.

Result: Generated Context c_g

Given a language model p, a prompt c_p, the number of beams n, the number of steps s, exploration steps for each beam m. Initialize current step $k = 0$. For each beam j, initialize the generated context for this beam $c_j = ''$.

while $k<s$ **do**

  For each c_j, generate m next token $t_{j1}...t_{jm}$ sampled from $p(\cdot|c_p + c_j)$, update $c_{jl} = c_j + t_{jl}$

  Choose the best n texts $c_{b1}, c_{b2}...c_{bn}$ with highest $f(c_{jl}|c_p)$.

  For all j, update $c_j = c_{bj}$.

  Update k=k+1.

end

Output the best beam $c_g = c_1$.

3.2 Inverse Prompting

In this paper, we introduce a new method based on a new scorer f in beam search, called *inverse prompting*. Unlike previous controllable methods such as CTRL or PPLM which needs additional attribute model training or manually-defined codes, inverse prompting directly uses the original language model itself to improve its generation.

One main issue that reduces the quality of the generated text is the run-away phenomena shown in Table 1. The text gradually becomes irrelevant to the given prompt as the sentences being generated. As the distance between the given prompt and the generated sentence becomes larger, it hinders the generator to keep a close connection with the prompt.

To alleviate this issue, our main idea is to design a novel beam search scoring function that evaluates the log likelihood in an inverse direction; e.g., if the prompt can be generated back from the text, they ought to be very related with each other:

$$f(c_g|c_p) = \log p(c_p|c_g). \tag{1}$$

Texts are not always fluent if we read them in an inverse way. In question-answering, the prompt may be "Question:${**Question**} Answer:". It is natural to follow the answer after the prompt, yielding "Question:${**Question**} Answer:${**Answer**}". However, it is very unlikely that in natural language the order will present in the inverse way "${**Answer**} Question:${**Question**} Answer:". Simply using equation 1 only results in failure.

However, thanks to the nature of natural language, there do exist ways to rearrange contexts to make them appear in an inverse order properly. Let's continue with the above instance: For "Question:${**Question**} Answer:${**Answer**}", there do exist a way in natural language to inverse it: "${**Answer**} answers the question:${**Question**}".

To achieve the core idea of Eqn. 1, we simply need to alter the format of the prompts and texts:

$$f(c_g|c_p) = \log p(c'_p|c'_g), \tag{2}$$

where c'_g is inverse prompt under a new format, and c'_p being the inverse text. Table 2 displays some examples of this transformation format. For c_p = "Question:${**Question**} Answer:" and c_g = "${**Answer**}", we only need to set c'_p = "${**Question**}" and c'_g = "${**Answer**} answers the question:", equation 2 shall work. Inverse prompting ranks different beams by their likelihood to generate back the original prompt in an inverse way, promoting the most relevant generated texts. Inverse prompting can be used as long as the language supports an inverse structure to rearrange the prompt and context in a proper way. Detailed illustration for language model generation and language model inverse prompting is presented in Figure 2.

The top rows in table 2 are the formats of inverse prompting in the two experiments of our paper—long-form question answering and poem generation, while the bottom rows are additional examples of how inverse prompting can be used in other text generation tasks.

Inverse prompting is a simple method and easy to implement. The method requires no additional models or data processing, as the inverse prompting score can be simply computed by the same language model used for generation. However, inverse prompting offers large improvements to the qualities of the generated texts, which we will show in Sections 4 and 5.

4 IMPLEMENTATION

We mainly use two long-term text generation tasks, Open-Domain Long-Term Chinese Question-Answering, and Open-Domain Chinese Poem Generation, which require the AI to generate long, in-depth contexts according to relatively short prompts, to demonstrate the fantastic performance of inverse prompting.

We believe that as the relevance between generated texts and the given prompt (questions/titles) improves, the generation quality will increase too. So we conduct inverse prompting on questions/titles(问题/标题) in our experiments, as shown in the first four rows in Table 2.

4.1 Base Language Model

Our base Chinese language model is trained using Megatron-LM[19] with Transformer-XL[4].[1] The model has 2.86 billion parameters. The training set contains 302GB of raw Chinese data abstracted from multiple Chinese websites including Baidu, Zhihu and Sougou. We train the base model using 64 GPUs for 160,000 steps. Details of training settings and datasets are displayed in Appendix A.1.

4.2 Open-Domain Long-Form Question-Answering

Long-Form Question-Answering, like the QAs on Quora, Zhihu or Sougou, is a form of question-answering that questions are short and described in one or two sentences, while the answers have to be long, informative and in-depth. The questioner expects a thorough answer answering the question he asks in detail.

We apply inverse prompting in this way to generate Long-Form Answers given question prompts: We generate sub-sentences randomly according to language model $\mathcal{LM}$, and do beam-search with inverse prompting in sub-sentence level. To ensure the answer follows the question, we apply inverse prompting (Equation 2 for each sub-sentence and sum up their scores. To keep the generated context fluent, we combine the scores with length-normalized forward perplexity,

$$f(c_g|c_p) = \frac{1}{n}\sum_{s\in c_g} \lambda_1 \log p(c'_p|s') + \lambda_2 \frac{\log p(c_g|c_p)}{n(c_g)^\lambda}. \tag{3}$$

4.3 Open-Domain Poem Generation

Traditional Chinese Poem generation is the pearl of domain-specific long-form Chinese text generation tasks. Traditional Chinese poems have their specific complex format and word usages different from modern Chinese language. Most of the poems are written by ancient poets to express their feelings, describe things they are doing, or ideas on different items. Generation of meaningful text under the poem format given open-domain information is very hard for both state-of-the-art AI models and humans.

In this paper, in addition to Open-Domain Long-Form QA, we challenge our inverse prompting for a seemingly impossible task

[1] The model is released at https://github.com/THUDM/Chinese-Transformer-XL.

Table 2: Inverse prompting transformation Table. The first rows represents the inverse prompts used in experiments.(in Chinese and English) Some additional examples of inverse prompting format are also displayed.

Prompt c_p	Generated Text c_g	Inverse Prompt c'_g	Inverse Text c'_p
问题：**${Question}**? 描述：**${Description}** 回答： Question: **${Question}**? Description: **${Description}** Answer:	**${Answer}** **${Answer}**	"**${Answer}**" 回答了问题 " "**${Answer}**" answers the question "	**${Question}**?" **${Question}**?"
标题：**${Title}** 体裁：诗歌 正文： Title: **${Title}** Genre: Poem Text: **${Title}**	**${Poem}** **${Poem}** **${Text}**	"**${Poem}**" 出自 "**${Poem}**" is from **${Text}** discusses the topic	诗歌《**${Title}**》 poem **${Title}** **${Title}**
Title: **${Title}** Genre: Poem Author: **${Author}** Text: Title: **${Title}** Emotion: **${Emotion}** Text: Title: **${Title}** Keywords: **${Keywords}** Text: Title: **${Title}** Text:	**${Poem}** **${Text}** **${Text}** **${Text}**	"**${Poem}**" is written by **${Text}** expresses the emotion **${Text}** studies the research field of **${Text}** is from news	**${Author}**. **${Emotion}** **${Keywords}** **${Title}**

– Using the language model trained on modern texts to generate Open-Domain Traditional Chinese Poems.

We basically keep the inverse prompting format of Eqn. 3 while adding a poem-format term to the beam-search (as shown in Eqn 4), which penalizes generated texts by the degree of the deviation to the poem format in rhythms or tones.

$$f(c_g|c_p) = \frac{1}{n}\sum_{s\in c_g} \lambda_1 \log p(c'_p|s') + \lambda_2 \frac{\log p(c_g|c_p)}{n(c_g)^\lambda} - \lambda_3 l_{format}(c_g) \tag{4}$$

4.4 Self Training for Poem Generation

Given that the model is trained on modern Chinese texts including very few poem-format texts, it can hardly generate texts fully match the poem format while maintaining a strong relevance to the given title.

Therefore, to improve its performance, we try the generate-and-fine-tune self training protocol in AlphaGo-Zero [20] for this task.

For each self-training cycle, we randomly select 1,500 titles and let the model to produce poems based on these titles, then we fine-tune the model on these generated poems for 2,000 steps. This cycle can be repeated multiple times and in our experiments we repeat this cycle for 2 times.

We expect the self-trained model to be more likely to generate sentences with better poem formats and other poem-specific properties like aesthetics without losing their relevance to the given title, hence their quality may be improved.

5 EXPERIMENTS

In this section, we display the human-evaluation results of inverse prompting on two long-form text generation tasks, open-domain long-form QA and open-domain poem generation.

5.1 Human Evaluation Protocol

We first introduce how our human evaluation on the two tasks is conducted. Table 3 illustrates the statistics for human evaluation experiments. For open-domain long-form QA, we recruit 45 people, mostly university students, to evaluate the quality of the generated answers. As for the evaluation of poem generation, we invite 11 experts on traditional Chinese poems. Some of them previously participated in the evaluation of Jiuge, the prior state-of-the-art Chinese poem generator. The others are either similarly known to be knowledgeable on poems or guaranteed to be qualified for the evaluation by some poem contests.

Table 3: Human Evaluation Statistics.

Task	Participants			Scores
	Recruited	Finished	Valid	Collected
long-form QA	45	32	30	12,000
Poem generation	11	10	10	4,000

Each task contains 100 prompts. For each prompt, we provide 4 different contexts (answers for long-form QA and poems for Poem Generation) for evaluators to evaluate. An evaluator needs to score each context from multiple aspects and give an overall rating for each context on our online evaluation platform within one week time.

To ensure participants making evaluations seriously, for each prompt we ask the participants to select the best context. Then we will check if this answer is consistent with the overall ratings for those 4 contexts additionally. If the proportion of inconsistent answers is over 20%, the participant will be regarded as not seriously doing the experiment, therefore his scorings will be removed. Other scores will be maintained and collected to be the scoring.

As listed in Table 3, 32 evaluators in long-form QA evaluation and 10 experts for traditional Chinese poems finished his/her evaluation. 30 of the finished evaluators in long-form QA experiment are valid, while all the 10 finished experts in our traditional Chinese poem experiment are valid.

5.2 Open-Domain Long-Form Question-Answering

For open-domain long-form Chinese QA evaluation, we randomly select 100 questions from various domains in Zhihu, a Quora-like Chinese open-domain long-form QA platform. In Zhihu, users can ask questions with some background descriptions, or provide informative long-form answers to the raised questions. Besides, users can "Upvote" or "Downvote" answers based on their opinions.

In this experiment, we only select questions that are excluded in the training set of our base model. For each question, we display one highly-upvoted human answer and three AI-generated answers produced by CPM [27], prompting baseline, and inverse prompting respectively.

We shuffle the order of all answers and ask human evaluators to score the answers through four aspects including:

(1) **Fluency** Whether the answer is well-formed and logical to read. Rated from 1 to 5.
(2) **Informativeness** Whether the answer contains useful information for the given question. Rated from 1 to 5.
(3) **Relevance** Whether the answer is relevant to the given question. Rated from 1 to 5.
(4) **Overall** The overall quality of the answer. Rated from 1 to 10.

Table 4: Performance for open-domain long-form Chinese QA under Human Evaluation.

Method	Fluency (1-5)	Inform.[1] (1-5)	Relevance (1-5)	Overall (1-10)
CPM [27]	2.66	2.47	2.36	4.32
Prompting Baseline	3.44	3.25	3.21	5.97
Inverse Prompting	**3.61**	**3.43**	**3.59**	**6.51**
Human Answers	3.80	3.61	3.67	6.85

[1] Informativeness

Table 4 shows that inverse prompting outperforms both the prompting baseline and the previous SOTA Chinese language model CPM by a large margin in all individual aspects, as well as the overall quality of the generated answers.

Despite inverse prompting only forces the answer to be more related to the question in this experiment, an interesting finding is that by producing more relevant answers, inverse prompting also makes improvements on the fluency and informativeness of the generated answers, raising the overall quality as well. This supports our hypothesis in section 4.

5.3 Open-Domain Poem Generation

The second experiment is to evaluate the task of open-domain poem generation. This task is similar to the long-form QA experiment described above. We randomly design 100 poem titles including various open domains for evaluation. These poem titles never appear in any real poems in the training set or being used as randomized titles in our reinforcement learning process.

For each title, we apply four different methods to generate poems, including Jiuge (the SOTA model for open-domain Chinese poem generation), the beam search baseline with poem format loss l_{format}, inverse prompting with poem format loss (Equation 4) and inverse prompting with the self-training mentioned in section 4.4. These four poems are shuffled for evaluation. For each generated poem, we request evaluators for 5 ratings:

(1) **Format** Whether the generated poem follows the rule of rhythm in traditional Chinese poetry. Rated from 1 to 5.
(2) **Innovation** Whether the sentences are copied from existing poems or created with innovative expressions. Rated from 1 to 5.
(3) **Relevance** Whether the content of the poem is related to the given title. Rated from 1 to 5.
(4) **Aesthetics** Whether the poem has obscure meanings apart from its obvious meanings, making it aesthetically better? Rated from 1 to 5.
(5) **Overall** The overall quality of the poem. Rated from 1 to 10.

Table 5: Performance for open-domain Traditional Chinese Poem Generation under human expert evaluation.

Method	Format (1-5)	Innov.[1] (1-5)	Relevance (1-5)	Aes.[2] (1-5)	Overall (1-10)
Jiuge [28]	**3.60**	2.47	1.99	**3.12**	3.57
Beam Search Baseline	2.79	1.10	1.16	2.44	1.35
Inverse Prompting	2.56	2.71	2.92	2.33	4.00
Inverse Prompting +ST	2.42	**2.92**	**3.65**	2.18	**4.40**

[1] Innovation
[2] Aesthetics

Table 5 illustrates the experimental results. The average scores for all methods are low as all of the experts are extremely critical. They only give high scores to very few perfect poems. An estimated score for selected high-quality human-written poems is 6, while for average poems written by ancient celebrities, the average score is estimated to be around 5.

The beam search baseline can hardly generate appropriate poems. Even with the poem format loss, it only outputs unrelated sentences copied from existing famous poems that appear in modern Chinese languages.

However, with the help of inverse prompting, the overall quality of generated poems surpasses *Jiuge*. Moreover, the self-training can further improve the performance on top of inverse prompting.

Generally, *Jiuge* is good at generating sentences with beautiful words and gorgeous rhythm, since it is designed to focus strictly on poem formats. Nevertheless, according to human evaluation results, despite it sometimes does generate relevant poems, most of its generation results are not quite related to the given title and comparably weak at innovation.

Inverse prompting offers innovative and relevant expressions in the form of traditional Chinese poems. However, as the model is trained on modern Chinese texts, it is inevitably not so good in following traditional poem formats like rhythms or tones. It also doesn't handle aesthetics well, as this is common in ancient Chinese poems, but rarely appears in modern Chinese.

Despite these disadvantages, the experts still agree to give poems generated by inverse prompting a much higher average overall score than Jiuge due to their high relevance to titles and innovative expressions.

In section 4.4, we expect the self-training can bring better format and aesthetics to the generated poems. However, to our surprise, the self-training further enhances the innovation and relevance by a large margin at the cost of a minor decrease in format and aesthetics, suggesting that what the model really learns under our reinforcement learning scheme is to be more relevant. By

generating more relevant poems to the title with more innovative expressions, its average overall score becomes much higher additionally. Eventually, inverse prompting with self-training gets 4.40 average overall score, compared with Jiuge's 3.57.

One possible explanation for this phenomenon is that in order to be more relevant to open-domain titles which may never appear in the domain of Traditional Chinese Poems, the model has to be more innovative in language arrangement and less focused on formats or aesthetics.

In Appendix A.3, we discuss our deviation analysis and calculate p-values for different methods on the above two tasks.

5.4 Poem Turing Test

Apart from human evaluation for open-domain titles, we also test the performance of it on domain-specific titles. Inspired by turing test [22], we similarly designed a traditional Chinese poem turing test to further evaluate the generated poems quality of inverse prompting.

In the turing test, a human interrogator is requested to distinguish between generated poems and human poems. We implement an online game platform [2] where any netizen can participate. Each player is given several pairs of poems with each pair contains one poem written by a human poet and the other one generated by AI under the same title. The human poems are randomly selected from *Quan Tang Shi* (《全唐诗》). The player needs to figure out which poem is written by the human poet. We generate 1,500 pairs of poems and randomly display 5 pairs for each game.

For the inverse prompting+ self training model, we collect 4,592 records from 755 users. The selection ratio is 46.4%(2,132) for AI and 53.6%(2,460) for human poems.

5.5 Case Studies

In this section, we discuss explicitly the pros and cons of inverse prompting by performing series of case studies on the two tasks – open-domain long-form QA, and open-domain poem generation.

Table 6 exhibits a comparison between two answers for a question on how to deal with stress at work during pregnancy. We list the answer generated by inverse prompting and the human answer. In this case, the evaluators even give higher scores to the inverse prompting generated answer than the human answer. Both answers provide comprehensive and practical aids related to the question. The human answer is more emotional and gives advice based on the answerer's own experience. The generated answer, on the other hand, raises the point that pregnant mothers should insist on working and overcome the difficulties straightforwardly. Then it uses several sentences to enhance this point, which turns out to be more informative, reasonable and persuasive.

While the proposed method seems to be able to understand questions, integrate learned knowledge and generate logical answers, we found that numbers in the task are comparatively difficult, which often lead to some chaotic outputs. In Table 7 we show a bad case generated by inverse prompting that only receives a 4.10 score in overall quality. While the question is asking about Chapter 882 in the *One Piece* manga, the model is clearly unable to understand the differences between Chapter 885 and the asked 882.

[2]https://turing-poet.aminer.cn/index.html

Table 6: A Perfect Example of inverse prompting generating better answer than human in open-domain long-form QA.

Question	孕期工作压力大导致抑郁怎么办? I feel a lot of pressure in my work during my pregnancy, making me depressed. What shall I do?
Inverse Prompting (7.97)	孕妈妈难免会觉得工作繁重, 工作中同事有时候相互不理解, 甚至可能受到侮辱, 孕期本来就心情抑郁, 压力可想而知; 而且工作强度大, 孕妈妈也缺乏足够的休息, 导致工作状态不佳, 进而造成抑郁。孕妈妈的生活中缺少了陪伴, 老公的陪伴也少, 会让孕妈妈更加寂寞, 因此容易产生抑郁的情绪, 工作上的压力也随之压上来了。所以建议孕妈妈尽量坚持孕期去上班, 不要去吵架, 好好把手上的工作做好, 这样才能在整个孕期保持好心情, 工作受委屈, 孕妈也容易有情绪, 跟老公好好沟通, 让他也分担孕妈妈的工作。 Pregnant mothers will inevitably feel that the work is heavy. Colleagues at work sometimes do not understand each other, and may even insult pregnant mothers. Pregnant mothers are naturally with lots of pressure. Intense work and the lack of rest could lead to poor working status, which further results in depression. The lack of companionship from husbands will make pregnant mothers more lonely. Therefore, they are prone to depression and increasing pressure at work. Therefore, it is recommended that pregnant mothers try their best to go to work during pregnancy, not to quarrel, and do a good job at their hands, so as to maintain a good mood throughout the pregnancy. When aggrieved at work, pregnant mothers are also prone to emotions. Communicate with your husband and let him share the work of the pregnant mother.
Human Answer (7.30)	我也是刚升职业务经理就发现自己怀孕了, 去年已过宫外孕, 上岗一周工作强度瞬间变成以前的五倍, 压力特大, 做梦还在工作, 自身体质不是很好, 每天都有气无力的, 后来医生给开了假条在家修养一个多月了, 心态也调整好很多。工作上不求完美, 已宝宝和自己的健康为主, 只做力所能及的事情。但愿你也能调整过来, 没有什么过不了的坎, 为了宝宝一定要坚强! I found out myself being pregnant when I was just promoted to the business manager. I had an ectopic pregnancy last year. The intensity of my work during the first week became five times that before. I was stressed and even dreaming of working at night. I was not very healthy and felt weak every day. The doctor prescribed a leave note for more than a month of self-cultivation at home, and then my mentality has improved a lot. Don't seek perfection at work. Focus on the health of your baby and yourself, and only do things within you power. I hope you can also adjust to better status, as there are no obstacles that cannot be overcome. You must be strong for the baby!

Table 7: A bad case for inverse prompting generated texts. It can't overcome the barrier of maths.

Question	如何评价海贼王 882 话? What's your comment for Chapter 882 in *One Piece*?
Inverse Prompting (4.10)	海贼王 885 话: 一进新世界, 塞到剧场版的草帽一伙朋友, 咱们海贼王的忠实粉丝, 终于看到了一个完美的结局了! 感谢尾田老师, 为我们准备了这么一个好结局。让我们喜欢海贼可以一直一直, 永不结束! 海贼王 876 话: 经评论区友情提示,861 话才是大结局, 本文结束。 *One Piece* Chapter 885: As soon as entering the new world and stuffed into the theater version of the straw hat group of friends, we loyal fans of *One Piece*, finally witness a perfect ending! Thank you Mr. Oda for preparing such a good ending for us. Let us love the *One Piece* forever and never end! One Piece Chapter 876: After the friendly reminder from the comment zone, Chapter 861 is the finale, my answer ends.

Besides, the answer itself is self-contradictory. It is worth noticing that such a chaotic problem in maths universally exists for language models. Previous study [18] also shows that it is extremely hard for neural models to do mathematical reasoning.

In Table 8, we display all 4 answers for the question "How to wash purple sweet potato on clothes" with the average overall

Table 8: A representative case in open-domain long-form QA that the quality of answers in this problem reflects the overall performance of different methods.

Question	紫薯沾到衣服上怎么洗? How to wash my clothes when they happen to contact with purple sweet potatoes?
CPM (3.10)	毛巾内外过滤, 晾干后用肥皂搓一遍袜子抹平, 再往床上一铺就可以直挺挺睡一觉啦 问题分析: “高温” 啥表现? 为什么紫薯沾到衣服上会烂? 问题描述: 回答: “意料之外” “太惊喜” 更 “意外” 问题分析: 这个..... 这个也难得吧。毕竟紫薯那么可爱 完。 Filter the inside and outside of the towel, dry it and rub it with soap to smooth the socks, and then put it on the bed to sleep upright Problem analysis: What does "high temperature" behave? Why does purple sweet potato rot when it gets on clothes? Problem description: Answer: "Unexpected", "Too surprised" and more "surprising" Problem analysis: This...this is also rare. After all, purple sweet potato is so cute End.
Prompting Baseline (3.40)	沾上牛奶的衣服, 可用冷水先浸渍一下, 再用肥皂洗一下, 放在阴凉通风处晾干。沾上啤酒或者白酒的, 先用清水洗一下, 再用肥皂洗, 再用清水漂洗。 Clothes stained with milk can be soaked in cold water, washed with soap, and dried in a cool, well-ventilated place. If it is stained with beer or liquor, first wash it with water, then wash with soap, and then rinse with water.
Inverse Prompting (6.63)	衣服沾到紫薯上的污渍很难洗干净, 因为像紫薯这种比较软的物体, 可以很容易附着在衣服上, 后续实在是很难洗。因为紫薯比较硬, 所以先把污渍弄下来, 用洗衣液或洗衣粉在污渍处搓洗, 如果是比较硬的紫薯污渍, 需要把紫薯用搅拌机搅碎, 然后减淡溶液, 再搓洗, 挤干后漂洗干净衣服。 It is difficult to clean the stains on the purple potato on the clothes, because the softer objects like purple potato can easily adhere to the clothes, and it is really difficult to wash it afterwards. Since purple sweet potato is relatively hard, we need to remove the stain first. Use laundry detergent or washing powder to scrub the stain. If it is a hard purple sweet potato stain, you need to crush the purple sweet potato with a mixer, then lighten the solution, and then scrub. After squeezing dry, rinse the clothes clean.
Human Answer (8.67)	首先请查看衣物洗涤标签, 确认衣物是否可以水洗, 有部分衣服(例如羊毛等) 是不能水洗的, 不能水洗的衣服建议送往干洗店处理, 确认可以水洗的衣服, 可按照如下方法清洗: 首先先用工具刮走可以刮掉的紫薯, 刮完后再泥点上直接涂上蓝月亮手洗专用洗衣液, 涂的时候衣服不要沾水, 洗衣液需要完全覆盖紫薯污渍; 覆盖后等 5 分钟, 让污渍和洗衣液充分接触和反应,5 分钟后, 轻轻搓洗有紫薯泥的地方 (可用白毛巾帮助反复擦洗有泥点的地方, 直到擦洗干净), 最后再将整件衣物按照平常的习惯, 加洗衣液手洗或者机洗。 First, please check the laundry label to confirm whether the clothes can be washed. Some clothes (such as wool, etc.) cannot be washed. It is recommended that the clothes that cannot be washed should be sent to a dry cleaner. If you confirm that the clothes can be washed, you can clean them as follows: Use a tool to scrape away the purple potato that can be scraped off. After scraping, apply the blue moon hand wash special laundry detergent directly on the mud point without any water. The detergent needs to completely cover the purple potato stains. Wait 5 minutes to allow the stains to fully contact and react with the laundry detergent. After 5 minutes, gently scrub the purple potato mash (you can use a white towel to repeatedly scrub the muddy spots until they are clean), and finally wash the whole piece of clothing as usual, such as hand wash with detergent or machine wash.

scores. The best answer is written by a human, which comprehensively introduces the solution to the problem step by step. The answer generated by inverse prompting offers a similar solution but with fewer details. The prompting baseline does not give a precise answer to the original question. Instead, it tries to answer another question “How to wash out the liquids on clothes such as milk or beer?”. This tells us why we need to use inverse prompting to force the generated answer to be closely related to the original question. Finally, CPM can neither produce fluent answers nor provide useful information. This example illustrates how the difference in overall ratings for different methods in Table 4 come from in a representative way.

Table 9: A representative case in poem generation that reflects the overall qualities of different methods.

Title	咏斑马 *Zebra*
Search Baseline (2.20)	草枯鹰眼疾，雪尽马蹄轻。平沙日未没，黯黯见临洮。 Grass droops and eagle eyes sharpens, snow eliminates and horse hoofs clear. Plain desert with sun unset, Lintao is seen behind shadows.
Jiuge (5.10)	玉勒金羁下紫宸，双鬃白鬣起风尘。天生精卫才堪数，人是毛延未觉春。 曾向龙池称汉女，应随鸡厩忆吴神。不须更借千年力，长把黄云拂地新。 Jade rein golden saddles going down the sky, double white hair going through wind and dust. The creature is comparable to Jingwei bird in the sky, or Maoyan as human. It has been to dragon's pool and praise Chinese women, and also to chicken's lounge to memorize gods. It doesn't need to borrow thousands of years of strength, the yellow horse is powerful on its own.
Inverse Prompting (4.80)	胡地多豪健，蹄间刺文猩。一双如天骥，毛色壮如熊。细看前足小，低头恶鬣同。天生奇犹士，万里思奔冲。 Horses in distant lands are usually strong, with tattoos on their hoofs. The creature has a pair of sky-like hoofs and panda-like fur. Looking closely you'll find its front feet being small, and its hair being the same. It is born to be unique, and it desires to run for tens of thousands of miles.
Inverse Prompting +ST (5.70)	一双斑驳蹄，踏碎苍苔路。空蒙古兽皮，散落犹群兔。 Dappled hoofs, step through moss roads. Despite their fur being similar to ancient beasts, they are as mild as rabbits.

Table 10: A bad case that inverse prompting is not successful in understanding the given title Iroquois even under after self training.

Title	叹易洛魁 *Sigh for the Iroquois*
Inverse Prompting +ST (3.50)	白帝金城沙漠中，易生妖孽害苍生。杀人百万藏地府，放火无殃是此星。 The golden city of the White Emperor is located inside deserts, where monsters are easy to appear and damage people. The monsters kill millions and hide in hell, and they receive no damage against fire.

Table 9 shows poems generated by different methods under title *Zebra* (咏斑马). Zebra is an open-domain concept that never appears in any traditional Chinese poems (as there's no zebra in ancient China). However, there exist lots of traditional Chinese poems for different types of animals. We would like to see how different methods generalize the traditional Chinese poem format for zebras.

Note that the direct meaning for “zebra”(斑马) in Chinese is "spotted horses", so models are likely to misuse the concept of "horse". The prompting baseline copies sentences from famous existing poems for horses and gets only 2.20 for being an awkward copycat. Jiuge treats zebras as horses and applies a lot of analogy to glorify the “horses”, with good representation and perfect format

it gets a 5.10 overall score. Inverse prompting offers a description between horses and zebras and gets a 4.80 overall score, while inverse prompting with self-training states the hoof, the fur and the behaviors of zebras in a subtle way, differing zebras from horses, this poem is scored the highest (5.70) by expert evaluators.

This is a representative instance for poem generation. The other 99 poems are also evaluated in such a way that the title is in a category that exists a lot in traditional poems. However, the precise concept of the title is new. For example, climbing different mountains or towers is popular in traditional Chinese poems, and we design open-domain titles like "Climbing Eiffel Tower" and "Climbing Mt. Everest" which never appear in them. The prompting baseline often copies from existing poems. Jiuge usually gives an poem with good format between related and unrelated. inverse prompting seems to understand the title while inverse prompting with self-training understands the title better.

However, inverse prompting does not guarantee understanding of the given title. Table 10 illustrates a bad case under title *Sigh for the Iroquois* (叹易洛魁). Inverse prompting with self-training fails to understand the meanings of the native American tribe and mistreat it as some forms of monsters living in deserts and killing people. This may due to the low occurrence of the Iroquois in modern Chinese texts that the base language model itself cannot fully understand its meanings, and the self-training process is unable to help the model understand more about such concepts as well.

6 CONCLUSION

In this paper, we present a new method, inverse prompting for text generation. Inverse prompting offers a new option for controllable generation using language models by exploiting the inverse form of natural languages.

We experiment the text generation using inverse prompting, which is to assign inverse prompts that force the generated texts to be related to the prompt. Extensive experiments of human evaluations demonstrate that as the relevance increases, the overall quality of long-form Chinese texts also improves significantly. On long-form open-domain QA, inverse prompting improves AI-generated texts one step closer towards human level. More promising results occur in open-domain Traditional Chinese Poem Generation, as with inverse prompting, language models trained on modern Chinese context can generate poems that surpass the previous SOTA despite the disadvantages in format and aesthetics. Furthermore, with self-training, the inverse prompting poem generator can do even better in human evaluations.

Acknowledgements

This work is funded by National Key R&D Program of China (2020AAA0105200) and supported by Beijing Academy of Artificial Intelligence (BAAI).

We thank Ming Ding, Zhengxiao Du and Jifan Yu for helpful discussions.

REFERENCES

[1] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015), 1–18.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

[3] Casey Chu, Andrey Zhmoginov, and Mark Sandler. 2017. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950* (2017).

[4] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2018. Transformer-xl: Language modeling with longer-term dependency. (2018).

[5] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164* (2019).

[6] Wietse de Vries and Malvina Nissim. 2020. As good as new. How to successfully recycle English GPT-2 to make models for other languages. *arXiv preprint arXiv:2012.05628* (2020).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Long Jiang and Ming Zhou. 2008. Generating Chinese couplets using a statistical MT approach. In *Coling'08*. 377–384.

[9] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[10] James JY Liu. 1966. *The art of Chinese poetry*. University of Chicago Press.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[12] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al. 1977. Speech understanding systems: Report of a steering committee. *Artificial Intelligence* 9, 3 (1977), 307–316.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546* (2013).

[14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP'14)*. 1532–1543.

[15] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[17] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[18] D. Saxton, Edward Grefenstette, Felix Hill, and P. Kohli. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. *ArXiv* abs/1904.01557 (2019).

[19] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053* (2019).

[20] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[21] Robert F Simmons. 1970. Natural language question-answering systems: 1969. *Commun. ACM* 13, 1 (1970), 15–30.

[22] Alan M Turing. 2009. Computing machinery and intelligence. In *Parsing the turing test*. Springer, 23–65.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[24] Yingce Xia, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *arXiv preprint arXiv:1611.00179* (2016).

[25] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[26] Xiaoyuan Yi, Ruoyu Li, Cheng Yang, Wenhao Li, and Maosong Sun. 2020. MixPoet: Diverse poetry generation via learning controllable mixed latent space. In *AAAI'20*, Vol. 34. 9450–9457.

[27] Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, et al. 2020. CPM: A Large-scale Generative Chinese Pre-trained Language Model. *arXiv preprint arXiv:2012.00413* (2020).

[28] Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. Jiuge: A human-machine collaborative chinese classical poetry generation system. In *ACL'19*. 25–30.

A APPENDIX

A.1 Implementation Details

Training Details for the Base Model. For training of the base model, we use a training set of 302GB, the distribution of these data is shown in Table 11. The evaluation set contains 400MB Open-Domain QA contexts that is not used during training. We select the 100 questions in human evaluation from this evaluation set.

Table 11: Dataset Distribution.

Source	Format	Size
Baidu & Sougou Baike	Online Encyclopedia	133GB
Zhihu	Open-domain QA	131GB
Baidu QA	Open-domain QA	33GB
Generated TC-Poems	Traditional Chinese Poetry	1.6MB
Evaluation	Open-domain QA	440MB

As mentioned in section 4, we use GPT framework with its transformer model substituted to Transformer-XL. For optimization, we use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 1e-6$ and a 0.1 L2-weight decay. The learning rate is warmed up linearly over the first 3,000 steps to a peak value of $1e-4$, then is tuned with cosine decay to 10% of its peak value. The total training steps is 160,000. The training process uses 8 servers with 8 Nvidia V100 GPUs on each server. Each server has 96 Intel CPU cores and 376GB Memory. Serves are connected by 100G RoCEv2 network.

For reinforcement learning, on each cycle we first generate a few poems for each of the 1500 prompts, resulting in around 800KB of generated poem data. Our fine-tuning inherits the previous conditions of the optimizer from the previous model and train on generated poem data for 2,000 steps. We repeat this process twice, so the final size of train poems generated is 1.6MB. The fine-tuning uses one server with 8 Nvidia V100 GPUs.

Table 12: Training Dataset Distribution.

Task	beam size	generations per beam	max short sentences
Long-form QA	5	5	30
Poem (train/turing)	10	7	8
Poem (eval)	10	12	8

Parameters for Beam Search. Table 12 displays the beam search parameters we use. For long-form QA, we use a beam size of 5, and for each beam we generate 5 samples for the next short sentence, and we limit the length of the answer to 30 short sentences. For Poem Generation, we use a beam size of 10, for each beam we generate 7 samples for the next short sentence in reinforcement learning and the Turing Test, and 12 samples for open-domain title human evaluation. We limit the length of the generated poems to 8 short sentences.

For $\lambda, \lambda_1, \lambda_2, \lambda_3, l_{format}$ mentioned in Section 4, we take $\lambda = \lambda_1 = \lambda_3 = 1$, $\lambda_2 = 0.75$ for poem generation and use $\lambda_2 = 1.5$ for open-domain QA.

For poem generation, we design l_{format} using various format information like number of characters in each short sentence, repetitive, rhythm, and tones. Despite it doesn't work when applied to prompting baseline directly, it cooperates well with inverse prompting.

A.2 Human Evaluation Details

Our human evaluation is conducted on a platform. The whole task of evaluating 100 prompts is divided into 10 sub-tasks, and in each sub-task, the evaluator is required to score 4 contexts for 10 prompts in multiple aspects, like an online questionnaire.

The evaluation does not necessarily need to be finished at once. People can login and logout, change their answers for already completed problems, or continue evaluation from their current positions freely in one week's time. They only need to ensure that all evaluation questions have been answered before the deadline, with the ratings being consistent.

Valid evaluators for open-domain QA are paid 150 RMB yuan each (about $25), while each TCP (Traditional Chinese Poem) evaluator receives 300 RMB yuan (about $50), as evaluation for traditional poems requires more expert reviewers. The payment is not high but due to the flexible time arrangement for online and interesting content, the task still attracted a lot of participants.

We recruit 11 experts for TCP evaluation, 10 of them finished and all of those finished provide valid evaluations, we recruit 45 people for open-domain QA, 32 of them finish their experiments and 30 of them provide consistent evaluations.

For generating baseline texts, for QA, we generate the prompting baseline using the base text generation code under the prompt format of " 问题:${**Question**} 描述:${**Description**} 回答:", for CPM we apply the same prompt format and use its recommended text generation code.

For poem generation using Jiuge, we write code that can automatically make online queries to its public generation website http://jiuge.thunlp.org/ and get generated poems. Jiuge has a lot of format modes and we choose four most general modes without additional restrictions "5-Jueju","7-Jueju","5-Lvshi","7-Lvshi" (五言绝句，七言绝句，五言律诗，七言律诗). For each title Jiuge generates one best poem for each mode. However, it offers no hint about which poem it considers the best so we randomly choose one from the 4 generated for human evaluation.

A.3 Deviation for Human Evaluators and p-values

Table 13,14 displays the deviation of the scorings for human evaluators.

The deviation is calculated in a per-capita basis that we first average the scorings for each method on for every evaluator, then we compute the deviation based on the average scores of each human evaluators.

As can be seen, evaluators agree more on the quality for Chinese QA, while less on the qualities for poems.

With those standard deviations, assuming evaluators are independent with each other, we can calculate p-score. For poems we have $N = 10$, the p-value for Jiuge $\geq$ Inverse Prompting is 0.0544 while the p-value for Jiuge $\geq$ Inverse Prompting+self-training is

Table 13: Performance and Deviation for open-domain long-form Chinese QA under Human Evaluation.

Method	Fluency (1-5)	Informativeness (1-5)	Relevance (1-5)	Overall (1-10)
CPM	2.66±0.19	2.47±0.19	2.36±0.20	4.32±0.37
Prompting Baseline	3.44±0.19	3.25±0.20	3.21±0.22	5.97±0.42
Inverse Prompting	**3.61±0.17**	**3.43±0.19**	**3.59±0.20**	**6.51±0.38**
Human Answers	3.80±0.18	3.61±0.19	3.67±0.21	6.85±0.39

Table 14: Performance and Deviation for open-domain Traditional Chinese Poem Generation under human expert evaluation.

Method	Format (1-5)	Innovation (1-5)	Relevance (1-5)	Aesthetics (1-5)	Overall (1-10)
Jiuge	**3.60±0.25**	2.47±0.28	1.99±0.31	**3.12±0.31**	3.57±0.54
Search Baseline	2.79±0.37	1.10±0.13	1.16±0.16	2.44±0.38	1.35±0.27
Inverse Prompting	2.56±0.28	2.71±0.28	2.92±0.37	2.33±0.28	4.00±0.52
Inverse Prompting +ST	2.42±0.29	**2.92±0.28**	**3.65±0.33**	2.18±0.28	**4.40±0.47**

Table 15: Selected examples of inverse prompting poetry, poems with heading, and SongCi.

Title	比特币
Inverse Prompting +ST	外挖无穷洞，机神犹未休。卡中窥币影，河里验沙流。 屡载吸金主，孤深渍盗求。方知区块链，本是古来游。
Title	赠美国友人
Inverse Prompting +ST	来往逢知己, 天涯海角期。一杯深谢酒, 千里念相思。 此别不容问, 他方复见时。持来沙漠手, 赠尔拂云丝。
Title	三体
Inverse Prompting +ST	二向箔掀天地震, 长城弯折世无伦。 凡人举火我张目, 毕昂开图看此文。
Title	夜过虹桥机场 (藏头：清华大学)
Inverse Prompting (Heading)	清河暮霭中, 华馆灯明起。大道夹城通, 学宫横野里。 单航已远飞, 余迹犹存立。星火尚寻踪, 海霞空扑日。
Title	江城子 · 通胀
Inverse Prompting (SongCi)	混全钢铁伏完坚。铸山钱, 水淹天。蛇吞象箸, 狐食虎餐前。半化半人残骨贱, 丸美药, 不传偏。 饱谙此术雇员闲。算来年, 利究颠。元轻钞重, 市物贵颠连。通缩预期成祸兆, 君看取, 券如烟。
Title	浣溪沙 · 一曲新词酒一杯
Inverse Prompting (SongCi)	一曲新词酒一杯，十年兵态渐成非, 满街风絮忆归飞。 垂柳阑干溜索女, 寻常巷陌笑牵谁。如今想却最关眉。

0.0009, suggesting that under $p < .05$ we cannot fully reject the hypothesis that Jiuge is not worse to Inverse Prompting. However, Inverse Prompting with self-training is statistically better than Jiuge.

For QA, with $N = 30$ the p-value for Prompting Baseline ≥ Inverse Prompting is < .00001, while the p-value for Inverse Prompting ≥ Human is 0.0006. So inverse prompting is statistically better than the prompting baseline but is still worse than human.

A.4 Online Demo Platforms

We further developed the poem generation and add some other formats of poems, including heading, which pre-defines the first word of each short sentence before poem generation, and SongCi, which is another form of traditional Chinese context that involves much higher format standard. All of these downstream tasks are based on the inverse prompting+self training protocol , with tiny format adjustments for each downstream task.

We display these applications on our demo Wudao Poetry（悟道作诗）[3]. Users can also submit their customized titles and generate poems of their own. There is also a QA demo named Wudao QA (悟道问答) [4] where users can submit their own questions and descriptions to get an AI answer.

Table 15 displays some of the generated poems for these downstream tasks on the platform. More cases can be found on the platform, or generated according to users' submissions.

[3]https://pretrain.aminer.cn/apps/poetry.html
[4]https://pretrain.aminer.cn/os/qa