Context-Tuning: Learning Contextualized Prompts for Natural Language Generation

Tianyi Tang[1], **Junyi Li**[1] and **Wayne Xin Zhao**[1,2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
steventianyitang@outlook.com, lijunyi@ruc.edu.cn, batmanfly@gmail.com

Abstract

Recently, pretrained language models (PLMs) have made exceptional success in language generation. To leverage the rich knowledge encoded by PLMs, a simple yet powerful mechanism is to use *prompts*, in the form of either discrete tokens or continuous embeddings. In existing studies, manual prompts are time-consuming and require domain expertise, while continuous prompts are typically independent of the inputs. To address this issue, we propose a novel continuous prompting approach, called *Context-Tuning*, to fine-tuning PLMs for natural language generation. Firstly, the prompts are derived based on the input text, so that they can elicit useful knowledge from PLMs for generation. We refer to such prompts as *contextualized prompts*. Secondly, to further enhance the relevance of the generated text to the inputs, we utilize *continuous inverse prompting* to refine the process of natural language generation by modeling an inverse generation process from output to input. Moreover, we propose a lightweight context-tuning, fine-tuning only 0.4% of parameters while retaining well performance.

1 Introduction

Natural language generation (*a.k.a.* text generation) aims to produce plausible and readable text in human language from input data [Yu *et al.*, 2020]. Recently, by involving a massive number of parameters, pretrained language models (PLMs) such as T5 [Raffel *et al.*, 2020] have made exceptional success in language generation. To leverage the knowledge from PLMs, prompting methods have been proposed [Liu *et al.*, 2021], where the original input to PLMs has been extended by prepending discrete tokens or continious embeddings (called *prompts*). Following this paradigm, this work aims to study how to develop more effective prompting methods for text generation based on PLMs.

Early methods focused on human-written (discrete) prompts by manually constructing task-specific prompt templates [Raffel *et al.*, 2020; Radford *et al.*, 2019] such as "`TL;DR:`" for text summarization task. However, it is time-consuming and laborious to construct human-written prompts for various generation tasks. Furthermore, it has been reported that PLMs are sensitive to human-written prompts and improperly-constructed prompts may cause the performance decrease [Gao *et al.*, 2021]. In light of these problems, recent works proposed to utilize continuous prompts [Li and Liang, 2021] for text generation, consisting of free parameters which do not correspond to real tokens, and can be easily optimized during fine-tuning. However, existing prompting approaches typically adopt *static prompts* for generation, which cannot be adjusted according to input context. In some generation tasks such as story generation, there exists very limited information in the input. In such cases, it is difficult to provide sufficient semantic information for capturing the related content aspects only with static prompts.

To address above issues, we propose *Context-Tuning*, a novel continuous prompting approach to fine-tuning PLMs for natural language generation. There are three major technical contributions in the proposed context-tuning. Firstly, the prompts are derived based on input text, so that they can enrich the input by eliciting task- and input-related knowledge from PLMs, *e.g.,* commonsense and background information. Since the prompts are highly related to input context, we refer to them as *contextualized prompts*. Secondly, to further enhance the relevance between the generated text and the input text, we utilize a novel *continuous inverse prompting* [Zou *et al.*, 2021] to refine generation process. By maximizing the likelihood of predicting inputs conditioned on generated text and continuous prompts, context-tuning can generate texts highly relevant to the input text. Moreover, to ease the burden of training, we propose a lightweight context-tuning. By fine-tuning only 0.4% of the parameters, our method achieves competitive performance or even exceeds strong baselines.

Specifically, inspired by the masked language modeling (MLM) task in BERT [Devlin *et al.*, 2019], we adopt BERT as the prompt generator to derive the contextualized prompt vectors. Concatenating limited input and a sequence of "`[MASK]`" tokens as the input of BERT, we leverage the excellent mask-filling ability of BERT to predict them, and the last hidden state of these "`[MASK]`" tokens can be used as prompt vectors. Based on the contextualized prompts and input text, we adopt the PLM, *i.e.,* BART [Lewis *et al.*, 2020], to generate output text. Finally, we select the best candidate output text based on continuous inverse prompting.

To our knowledge, we are the first to encode input-related information into continuous prompts for text generation. Our context-tuning method can elicit relevant knowledge according to the specific input text. For evaluation, we compare our method and several baseline models on three natural language generation tasks. Extensive experiments demonstrate the effectiveness of our proposed context-tuning.

2 Related Work

Natural Language Generation. Natural language generation is one of the most challenging fields in natural language processing (NLP). It aims to produce human-readable text from input text. Current state-of-the-art results for many generation tasks, are based on fine-tuning PLMs, such as text summarization [Lewis *et al.*, 2020], dialogue system [Zhang *et al.*, 2020] and data-to-text generation [Ribeiro *et al.*, 2020]. As mentioned in Liu *et al.* [2021], controlled text generation is relevant to our input-dependent method. Controlled text generation aims to guide the generated texts in specific style [Hu *et al.*, 2017], length [Kikuchi *et al.*, 2016], or keywords [Dou *et al.*, 2021]. In contrast, our contextualized prompts elicit knowledge from BERT to enrich the input rather than controlling the specific property of generated text.

Prompting Methods. Prompting means prepending task instructions to the input and generating the output from PLMs. Typical methods mainly utilize manually designed task-specific prompts to adapt for different generation tasks [Radford *et al.*, 2019; Raffel *et al.*, 2020]. While, hand-crafted prompts are not flexible and cannot be applied to more kinds of tasks. Thus, recent works have concentrated on automating the search of discrete prompts [Shin *et al.*, 2020; Gao *et al.*, 2021]. However, searching prompts over discrete space is difficult to optimize due to the non-differentiable issue and continuous nature of neural networks. To handle these problems, many studies propose to optimize continuous prompts [Lester *et al.*, 2021; Li and Liang, 2021], which are more expressive and flexible to any tasks. Among these works, Prefix-Tuning [Li and Liang, 2021] and Prompt Tuning [Lester *et al.*, 2021] are two representatives focused on text generation and natural language understanding (NLU) tasks, respectively. Compared with these continuous approaches, our context-tuning encodes the context information of inputs into the contextualized prompts, and adopts continuous inverse prompting to further enhance the relevance.

There exist various studies on prompting methods [Jiang *et al.*, 2020; Qin and Eisner, 2021; Zhong *et al.*, 2021], whereas they almost focus on NLU tasks, which are choice questions and can be easily convert to filling "`[MASK]`" tasks. However, text generation aims to generate a sequence of tokens, in contrast to few options in limited space. Prefix-Tuning [Li and Liang, 2021] and GENPET [Schick and Schütze, 2021] have employed prompting methods for text generation, however, they mainly focus on lightweight fine-tuning or few-shot learning and do not achieve great performance under vanilla fine-tuning settings. In contrast, our context-tuning can improve performance under vanilla fine-tuning settings, and the lightweight strategy fine-tunes only 0.4% of parameters while retaining well performance.

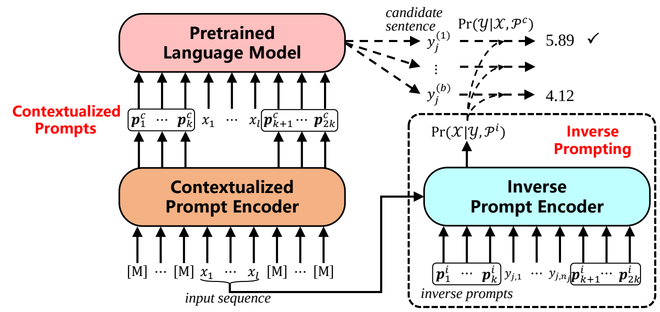


Figure 1: The overview of the proposed Context-Tuning. "[M]" denotes the mask token "[MASK]". By combining the forward probability $\Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c)$ and backward probability $\Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i)$, we select the sequence $y_j^{(i)}$ with the highest combined scores from all the candidates.

3 The Proposed Approach

In this section, we present the proposed *context-tuning* to fine-tune PLMs for natural language generation. We first introduce the contextualized prompts based on the input text for generating informative text. To further enhance the relevance of generated text to the input, we utilize a continuous inverse prompting to enforce the prediction of inputs given the generated text and continuous prompts. Figure 1 presents an overall illustration of the proposed context-tuning approach.

For natural language generation, we consider a general task setting, where the model generates the output sequence $\mathcal{Y}$ conditioned on the input sequence $\mathcal{X} = \langle x_1, \ldots, x_l \rangle$. The output text is usually composed of multiple sentences: $\mathcal{Y} = \{y_j : \langle y_{j,1}, \ldots, y_{j,t}, \ldots, y_{j,n_j} \rangle\}_{j=1}^{m}$. In context-tuning, we introduce contextualized prompts $\mathcal{P}^c = \langle \boldsymbol{p}_1^c, \ldots, \boldsymbol{p}_k^c \rangle$ into the input side. Thus, the prompt-based generation task can be formulated as:

$$\Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c) = \Pr(y_1, \ldots, y_m | x_1, \ldots, x_l, \mathcal{P}^c). \quad (1)$$

3.1 Contextualized Prompts

Instead of using static prompts [Lester *et al.*, 2021] (*irrelevant to input*), we use *contextualized prompts* specially for the input, which are expected to provide additional information such as world knowledge and task information extracted from PLMs to enrich the limited input text.

Masked Prompt Learning. Specifically, unlike Prefix-Tuning [Li and Liang, 2021] that prepends a sequence of static vectors to each layer of PLMs, we append a sequence of k continuous vectors on both the left and right sides of the input sequence $\mathcal{X}$ ($2k$ vectors in total). Inspired by the MLM pretraining task in BERT, we adopt BERT as the prompt generator to derive the contextualized prompt vectors. We first place a sequence of k "`[MASK]`" tokens on both sides of the input text $\mathcal{X}$ as $\widetilde{\mathcal{X}} = \texttt{[MASK]}_1, \ldots, \texttt{[MASK]}_k, \mathcal{X}, \texttt{[MASK]}_{k+1}, \ldots, \texttt{[MASK]}_{2k}$. By feeding $\widetilde{\mathcal{X}}$ into the prompt generator, we can obtain the top-layer representations of these "`[MASK]`" tokens:

$$\underbrace{\tilde{\boldsymbol{p}}_1^c, \ldots, \tilde{\boldsymbol{p}}_k^c}_{\text{prefix prompts}} / \underbrace{\tilde{\boldsymbol{p}}_{k+1}^c, \ldots, \tilde{\boldsymbol{p}}_{2k}^c}_{\text{suffix prompts}} = \text{Prompt-Generator}(\widetilde{\mathcal{X}}). \quad (2)$$

After conducting sensitivity analyses in the Appendix D, we set k to 150 with the best performance. Compared with randomly-initialized prompts, our BERT-based prompt learning method is more powerful to build the dependency between the prompts and input texts.

Aligning to Word Embeddings. Since these prompt vectors are latent embeddings, we further align them to the semantic space of word embeddings, by designing a two-step *semantic mapping* operator. For the first step, BERT predicts the probability distribution over its vocabulary based on these top-layer representations:

$$\Pr(w|\widetilde{\mathcal{X}}) = \text{softmax}(\boldsymbol{W}^V \tilde{\boldsymbol{p}}_k^c), \tag{3}$$

where $\boldsymbol{W}^V$ is a trainable matrix. For the second step, we multiply the probability distribution $\Pr(w|\widetilde{\mathcal{X}})$ with the word embedding matrix $\boldsymbol{E}$ and obtain the final contextualized prompt vectors:

$$\boldsymbol{p}_k^c = \boldsymbol{E} \cdot \Pr(w|\widetilde{\mathcal{X}}). \tag{4}$$

We consider these mapped vectors as *contextualized prompts*. Intuitively, the above semantic mapping can be considered as a weighted average of word embeddings according to their probabilities. Intuitively, the above *semantic mapping* can be considered as a weighted average of word embeddings according to their probabilities. Compared with existing continuous prompts, our contextualized prompt vectors better correspond to real word embeddings in semantic space, as shown in Section 4.5.

Applying the Prompts. After obtaining the contextualized prompts, we combine these prompt vectors and the word embeddings of $\mathcal{X}$ as the input of PLMs for generating the output text $\mathcal{Y}$. Specifically, we utilize BART as the base PLM to generate text by minimizing the cross-entropy loss function:

$$\begin{aligned}\mathcal{L}_c &= -\log \Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c) \\ &= -\log \Pr(\mathcal{Y}|\boldsymbol{p}_1^c, \dots, \boldsymbol{p}_k^c, \boldsymbol{x}_1, \dots, \boldsymbol{x}_l, \boldsymbol{p}_{k+1}^c, \dots, \boldsymbol{p}_{2k}^c).\end{aligned} \tag{5}$$

With the extracted knowledge from PLM, the contextualized prompts are helpful to generate informative output texts.

3.2 Continuous Inverse Prompting

Although contextualized prompts improve the informativeness of output, PLMs still have a tendency to digress from input text and generate off-topic texts as the text length increases [Zou *et al.*, 2021]. To deal with this issue, we propose *continuous inverse prompting* to enhance the relevance in an inverse manner from output to input. Compared to previous inverse prompting that depends on artificial construction [Zou *et al.*, 2021], our inverse prompting is based on continuous prompts, which can be optimized during fine-tuning.

Output-to-Input Relevance Enhancement. In order to model the relevance from output $\mathcal{Y}$ to input $\mathcal{X}$, we make a hypothesis that the output text is highly relevant to the input text if we can recover the input based on the output. Nevertheless, in some text generation tasks, it is non-intuitive to generate the input text given the output text. Hence, we employ prompts to alleviate this issue. We introduce continuous

Algorithm 1 The pseudo code for generation process of Context-Tuning.

Require: Model parameters $\Theta^{(c)}$ and $\Theta^{(i)}$, beam size b and maximum number of sentences n_m

Input: An input sequence $\mathcal{X}$
Output: A generated sequence $\mathcal{Y}$
Initialize step $j = 0$ and generated sentence $y_0 =$ ''
while $j < n_m$ **do**
  Derive contextualized prompts $\mathcal{P}^c$ based on $\mathcal{X}$ and previously generated sentences $y_1, \dots, y_{j-1}$
  Generate b candidate sentences $y_j^{(1)}, \dots, y_j^{(b)}$ according to Eq. 5
  Utilize continuous inverse prompts $\mathcal{P}^i$ to compute the likelihood of candidate sentences according to Eq. 6
  Choose the best sentence as y_s based on Eq. 7
  Update $j = j + 1$
end while
Concatenate $y_1, \dots, y_j$ as generated sequence $\mathcal{Y}$
return $\mathcal{Y}$

inverse prompts $\mathcal{P}^i$ and append them on both sides of the output $\mathcal{Y}$. Then, we utilize another PLM to measure the conditional probability $\Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i)$. Considering the output text $\mathcal{Y}$ might be much longer than the input text $\mathcal{X}$, we further model the probability at the sentence level:

$$\begin{aligned}\mathcal{L}_i &= -\log \Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i) \\ &= -\sum_{j=1}^{m} \log \Pr(\mathcal{X}|\boldsymbol{p}_1^i, \dots, \boldsymbol{p}_k^i, \boldsymbol{y}_{j,1}, \dots, \boldsymbol{y}_{j,n_j}, \boldsymbol{p}_{k+1}^i, \dots, \boldsymbol{p}_{2k}^i).\end{aligned} \tag{6}$$

Unlike contextualized prompts in Section 3.1, we expect inverse prompts to better reflect the relation between $\mathcal{Y}$ and $\mathcal{X}$, which is dependent on task rather than input. Thus, the inverse prompts are static and continuous in our approach.

Generation with Inverse Prompting. Finally, with the above-mentioned two techniques together in the generation process, we utilize a modified beam search algorithm shown in Algorithm 1 to generate the sequence $\mathcal{Y}$ with highest combined probability:

$$\mathcal{Y} = \underset{\mathcal{Y}}{\text{argmax}} \log \Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c) + \lambda \log \Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i), \tag{7}$$

where λ is a hyper-parameter to balance these two probabilities. After conducting sensitivity analyses in the Appendix D, we set λ to 4.0 with the best balance of performance.

In contrast to contextualized prompts that enrich the input information, continuous inverse prompting is used to make generation process more controllable. Even for later generated sentences, it can still keep them adhere to the input topic.

3.3 Discussion and Learning

In this part, we present the model discussion and optimization.

Discussion and Comparison. We learn contextualized prompts (Eq. 5) to elicit useful knowledge from PLMs for different inputs. As a comparison, previous continuous prompting methods [Li and Liang, 2021; Lester *et al.*, 2021] adopt

Datasets	Models	B-1	B-2	B-3	B-4	R-1	R-2	R-L	ME	D-1	D-2	#Para
WRITING-PROMPTS	**Fusion Model**	14.61	9.64	5.88	3.19	16.90	3.07	17.63	8.13	0.16	2.24	6.1×10^7
	GPT-2	21.41	13.89	7.88	4.08	12.73	2.80	14.92	8.14	0.25	1.26	1.2×10^8
	T5	6.76	4.07	2.44	1.46	22.36	3.07	17.13	7.65	1.48	7.37	2.2×10^8
	BART	26.98	16.16	9.61	5.71	27.28	4.19	19.35	11.08	0.57	3.80	1.4×10^8
	Prompt Tuning	27.58	16.55	9.80	5.78	27.32	**4.22**	19.34	11.26	0.56	3.65	1.4×10^8
	Context-Tuning	**28.11**	**16.83**	**10.01**	**5.95**	**27.48**	4.20	**19.41**	**11.33**	**0.61**	**3.86**	2.5×10^8
	Lightweight fine-tuning											
	Prefix-Tuning	22.52	13.43	7.98	4.74	25.42	3.64	18.49	9.66	0.50	3.33	2.4×10^7
	Prompt Tuning	4.73	2.79	1.64	0.97	4.43	0.48	3.48	1.71	0.09	0.78	2.3×10^5
	Context-Tuning	26.12	14.94	8.15	5.35	26.32	4.01	19.09	10.10	0.37	3.55	5.0×10^5
REDDITGEN	**Fusion Model**	24.44	14.70	8.78	5.22	11.00	2.66	14.09	5.45	0.01	0.03	6.1×10^7
	GPT-2	39.49	23.42	13.90	8.27	7.64	1.06	9.79	9.43	0.25	1.39	1.2×10^8
	T5	34.91	21.74	13.51	8.38	13.39	2.07	16.20	7.98	0.79	3.21	2.2×10^8
	BART	60.02	38.51	24.19	15.03	26.68	4.56	**20.83**	16.08	2.85	20.66	1.4×10^8
	Prompt Tuning	60.30	38.67	24.27	15.06	26.56	4.50	20.70	16.25	2.68	19.88	1.4×10^8
	Context-Tuning	**61.25**	**39.26**	**24.69**	**15.38**	**26.72**	**4.58**	20.57	**16.71**	2.91	20.74	2.5×10^8
	Lightweight fine-tuning											
	Prefix-Tuning	53.58	34.06	21.23	13.10	25.30	4.37	20.37	14.96	3.10	19.78	2.4×10^7
	Prompt Tuning	9.15	5.66	3.43	2.06	3.94	0.41	3.56	2.21	0.49	4.75	2.3×10^5
	Context-Tuning	54.99	35.30	22.20	13.82	24.46	4.07	20.36	14.14	**3.69**	**24.83**	5.0×10^5

Table 1: Performance comparisons of different methods for story and opinion generation tasks. B-n, R-n, ME, and D-n are short for BLEU-n, ROUGE-n, METEOR, and Distinct-n, respectively. Bold and underline fonts denote the best and the second best methods (the same as below). #Para denotes the number of fine-tuned parameters in each method.

static prompts, which are irrelevant to the input. Besides, we propose continuous inverse prompting (Eq. 6) to enforce the relevance of long output text by considering a generation process from output to input. Different from original inverse prompting, our inverse prompting is based on the continuous prompts, which can be optimized during fine-tuning.

Considering our method involves another PLM and more parameters, we also propose a lightweight Context-Tuning. Following IV *et al.* [2021], we only fine-tune the bias term of each parameter, resulting in fine-tuning only 0.4% of parameters of full models. In the meanwhile, Prefix-Tuning [Li and Liang, 2021] and Prompt Tuning [Lester *et al.*, 2021] freeze the PLM and only fine-tune the parameter of prompts. Prefix-Tuning fine-tunes prompts in each layer and fine-tunes 17% of parameters, while Prompt Tuning only fine-tunes the prompt concatenated to the input, resulting in fine-tuning 0.2% of parameters.

Optimization. For our model, we adopt the base version of BERT as our prompt generator. The number of prompt vectors k is set to 150. We utilize the base version of BART for text generation. The hyper-parameter λ in Eq. 7 is set to 4.0. There are two sets of trainable parameters in contextualized prompts and continuous inverse prompting, denoted by $\Theta^{(c)}$ and $\Theta^{(i)}$, respectively. First, we optimize $\Theta^{(c)}$, including BERT and BART, according to Eq. 5. Meanwhile, we optimize $\Theta^{(i)}$ according to the inverse generation loss using Eq. 6. During inference, we combine them and follow Algorithm 1 to select sentences that are informative and relevant to the input text based on Eq. 7.

4 Experiment

In this section, we first set up the experiments, and then report the results and analysis.

4.1 Experimental Setup

Construction of the Datasets

To measure the performance of our proposed context-tuning, we evaluate on three text generation tasks: WRITING-PROMPTS [Fan *et al.*, 2018] for story generation, REDDIT-GEN for opinion generation, XSUM [Narayan *et al.*, 2018] for text summarization. Specifically, WRITINGPROMPTS consists of pairs of story premise and response from Writing-Prompts forum. REDDITGEN contains pairs of post statement on a controversial issue, which are collected from Reddit. The detailed information of three datasets is listed in the Appendix A.

Baseline Methods

We consider the following baselines as comparison: Fusion model, GPT-2, BART, T5, Prefix-Tuning and Prompt Tuning. Among these baselines, Fusion model [Fan *et al.*, 2018] is a CNN-based Seq2Seq model and widely applied to story generation task; GPT-2 [Radford *et al.*, 2019], BART [Lewis *et al.*, 2020] and T5 [Raffel *et al.*, 2020] are three prevalent PLMs for natural language generation; Prefix-Tuning [Li and Liang, 2021] and Prompt Tuning [Lester *et al.*, 2021] are the recently proposed prompt-based model using continuous prompts for generation tasks.

Note that, original Prompt Tuning freezes the PLM and only fine-tunes the prompt concatenated to the input. In order for fair comparison, we also fine-tune the PLM of Prompt

Models	B-4	ME	D-2
Context-Tuning	5.95	11.33	3.86
w/o Continuous w Manual			
- human-written prompt$_1$	5.63	11.02	3.59
- human-written prompt$_2$	5.78	10.95	3.85
w/o BERT w LSTM	5.86	10.91	3.90
w/o BERT w RoBERTa	5.76	10.92	3.66
w/o Semantic Mapping	5.92	11.16	3.78
w/o Inverse Prompting	5.81	11.17	3.72

Table 2: Ablation analysis on WRITINGPROMPTS dataset.

Models	TT (%)	Flu.	Info.	Rel.	Coh.
GPT-2	81.20	3.90	3.27	3.77	3.50
T5	61.48	3.58	3.02	3.64	3.25
BART	77.17	3.82	3.27	3.74	3.59
Context-Tuning	**82.83**	**4.12**	**3.47**	**3.94**	**3.85**
Gold	94.00	4.26	3.90	4.33	4.01

Table 3: Turing test (TT) and human evaluation on WRITING-PROMPTS. "Gold" indicates the ground-truth texts. Flu., Info., Rel. and Coh. denote fluency, informativeness, relevance and coherence respectively.

Tuning as a strong baseline under vanilla fine-tuning setting. Prefix-Tuning prepends prompts in each layer of Transformer backbone, and we consider it as a lightweight fine-tuning method. The description and implementation details of baselines are in the Appendix B.

Implementation Details

In all experiments, we utilize the Adam optimizer and set $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-6}$. We adopt a linear warm-up of the learning rate over the first 2000 steps and a cosine decay after that. The peak learning rate is set to 2×10^{-5}. The batch size is set to 16. We train our model for 50 epochs and utilize the model with the best performance on validation set for generation. During inference, we apply the beam search method with a beam size of 5 and length penalty is set to 2.0.

Evaluation Metrics

To evaluate the performance of different methods on natural language generation, we adopt four automatic evaluation metrics, including BLEU [Papineni *et al.*, 2002], ROUGE [Lin, 2004], METEOR [Banerjee and Lavie, 2005], and Distinct [Li *et al.*, 2016]. Specifically, BLEU, ROUGE and METROR evaluate the quality between generated and real text, while Distinct measures the diversity of generated texts.

4.2 Performance Comparison

We present the results of different methods on story and opinion generation tasks in Table 6. The results of the text summarization are listed in the Appendix C.

First, it seems that the simple method, Fusion model, achieves poor performance among all baseline models. This result reveals that RNN models have significant limitation in generating long text.

Second, among all the baselines, we can clearly see that PLMs perform well overall. Pretrained on large-scale corpus, PLMs are able to understand natural language accurately and express in human language fluently. Note that, compared to other PLMs, BART performs best on these three generation tasks, which is owing to the encoder-decoder architecture and the DAE pretraining task. That is the major reason we select BART as our base generation model.

Third, the recently proposed continuous prompting methods, Prefix-Tuning and Prompt Tuning, do not achieve ideal performances in these tasks. This phenomenon reflects the natural language generation tasks is more challenging than NLU tasks and only fine-tuning a few parameters cannot obtain great performance compared with vanilla fine-tuning. After fine-tuning all the parameters of PLM, Prompt Tuning can achieve the second best performance. As proposed in Li and Liang [2021], Prompt Tuning can be seen as the upper bound of the discrete prompt optimization, including human-written prompts [Raffel *et al.*, 2020; Radford *et al.*, 2019] and automatically searched discrete prompts [Shin *et al.*, 2020].

Finally, our model outperforms all the baselines over three tasks. The reason lies in that our context-tuning utilizes *contextualized prompts*, which can serve as queries to elicit input-relevant knowledge from PLMs. Under the lightweight fine-tuning settings, our Context-Tuning has comparable results of Prefix-Tuning, only with 2% parameters of Prefix-Tuning and 0.4% parameters of BART. And the reason is that Prefix-Tuning and Prompt Tuning adopts static prompts, which are task-specific and not related to the context information.

4.3 Ablation Analysis

In this part, we construct ablation experiments on the effectiveness of our proposed context-tuning. In contrast to previous prompt-based studies, our context-tuning has made several improvements. First, compared with manual prompts, we propose a continuous prompting approach to fine-tuning PLMs. Second, we adopt BERT as the prompt generator to derive the contextualized prompt vectors with semantic mapping. Finally, we utilize inverse prompting to further enhance the relevance of the generated texts. Here, we would like to examine how each factor contributes to the final performance. To see this, we prepare several variants for comparison:

• *w/o Continuous w Manual*: the variant removes the continuous prompts, but utilizes two kinds of human-written prompts, *i.e.,* prompt$_1$: "`Title: $Input Story:`" and prompt$_2$: "`Given the title $Input, please write the following story:`".

• *w/o BERT w LSTM*: the variant replaces BERT with Bi-LSTM to form the prompt generator.

• *w/o BERT w RoBERTa*: the variant replaces BERT with RoBERTa [Liu *et al.*, 2019] to form the prompt generator.

• *w/o Semantic Mapping*: the variant does not align to word embeddings and directly utilizes the top-layer representations of "`[MASK]`" tokens in the prompt generator.

• *w/o Inverse Prompting*: the variant removes inverse prompting (Eq. 7) from our proposed context-tuning.

From Table 2, we can see that variants replacing continuous prompts with manual prompts, are worse than the model with continuous prompts. This verifies the effectiveness of utilizing continuous prompts rather than discrete ones for

Premise	Write a Nature Documentary about Pokémon .
GPT-2	<newline> it's been a long time since i've seen one of these things. <newline> i don't know how long it has been, but i know that it will be long. i have seen it before, and i will see it again. it is the only thing i can think of that makes sense to me, because it makes me feel like i am part of something greater than myself. and that is why i want to know more about it. because if i could, i would know what it was like to be human. to have a sense of purpose and purposelessness, to feel that there is a purpose in everything that you do, even if you are just a shell of what you once were. but that does not make it any less of a feeling. that feeling that i feel when i look into the eyes of another human, when they look at me and say, " i love you. "
BART	the pokémon are the most powerful of all theokémon in the world. <newline> they are the best of the best, but they are also the most dangerous. they have the ability to move faster than any otherokémon on the planet, and they are able to communicate with each other in a way that is impossible for humans to understand. however, they are not able to fight each other, because they are unable to communicate in a language that is not their own. they can only communicate through their own language, which is very different from any other language that humans can understand. they are very difficult to understand, and can not communicate in any language that can be understood by humans. they also have a very limited ability to speak, so they can not understand the language that they are speaking. in fact, they have no ability to communicate at all, because their language is very similar to that of a human.
Visualization of Contextualized Prompts: forest trees marching winds shadow food historically story movie explosive pet training pikachu animation japanese whilst critical asking we write	
Context-Tuning	" pokémon! " <newline> " what are you talking about? " the man asked. " i'm talking about a pokéball. it's called pikachu, and it lives in the forest. it lives on the edge of the forest, where it can eat anything it wants to eat. it is the only thing that can keep it alive. it can live in the woods, but it can also be eaten by any other creature. it has the ability to eat anything that it wants, and can even eat any other animal it wants. " he said. the man looked at the man, and said, " i don't know what you're talking about, but i do know that it can be eaten. " the two men looked at each other, and the man spoke, " you've got to be kidding me. "

Table 4: The generated examples of given premise from WRITINGPROMPTS dataset. Marks in visualized contextualized prompts and texts generated by context-tuning refer to four chosen aspects, *i.e.,* nature, documentary, Pokémon and the story generation task.

text generation tasks. The variants replacing the BERT-based prompt generator with LSTM or RoBERTa, are worse than the complete model. We further observe a slight performance drop by removing the semantic mapping and inverse prompting in our method. This implies that the proposed semantic mapping and continuous inverse prompting methods can indeed enforce the relevance of output text to some extent.

4.4 Human Evaluation

Generally, for text generation models, it is important to conduct human evaluation for further effectiveness verification.

We randomly select 500 input texts from the test set of the WRITINGPROMPTS dataset. We collect the generated story responses GPT-2, BART, T5 and Context-Tuning, and then shuffle them for human evaluation. Following Zou *et al.* [2021], we invite ten human judges to assign scores to a generated text with respect to four factors of quality, *i.e.,* informativeness, relevance, coherence, and fluency. We adopt a 5-point Likert scale as the scoring mechanism, in which 5-point means "very satisfying", and 1-point means "very terrible". Furthermore, inspired by Zou *et al.* [2021], we similarly design a Turing test, that a human judge is requested to distinguish whether the given text is generated by human. The description for the four factors and the human evaluation guidance can be found in the Appendix E.

We present the results of human evaluation in Table 3. It can be seen that our model is better than three baselines with a large margin. The major reason is that we utilize the contextualized prompts derived from the input text. Our contextualized prompts can extract knowledge from PLMs and serve as additional input information to be fed into PLMs, which can improve the informativeness of the generated text. Moreover, the proposed continuous inverse prompting method enhances the relevance of the generated text to the input.

4.5 Qualitative Analysis

Here, we further present intuitive explanations why our model works well through qualitative analysis.

Table 4 presents an example story from the WRITINGPROMPTS dataset and the generated story by our model and two baselines, *i.e.,* GPT-2 and BART. As we can see, there exists limited information in the input premise, besides several keywords such as nature, documentary, and Pokémon.

First, we can see that, the story generated by our context-tuning is highly relevant to the input text and conveys more rich semantic information. A major reason might be that, our contextualized prompts can elicit input-relevant knowledge from PLMs for generating more informative text. Although PLMs perform well in generating fluent text, we can see that GPT-2 and BART are still prone to generate unmeaningful and irrelevant content, such as "I love you" and "language".

Furthermore, in order to probe whether our contextualized prompts contains input-relevant knowledge, we visualize them to real words for better explanation. We use $2k$ contextualized prompts in total, and for each continuous prompt, we recall the word in the BERT vocabulary with the closest cosine distance to it. Finally, we select some words from $2k$ recalled words and showcase them grouped by four aspects in the row *Visualization of Contextualized Prompts* of Table 4. As we can see, most of the recalled keywords are included in the story generated by our context-tuning. We can infer that our contextualized prompts contain input-relevant knowledge. For example, the keywords "forest", "woods", and "animal", are closely related to the topic *nature*.

5 Conclusion

In this paper, we have presented a novel continuous prompting approach, *i.e.,* Context-Tuning, to fine-tuning PLMs for natural language generation. The core idea is to inject necessary context information into continuous prompts, called contextualized prompts, for enhancing the informativeness in generation. The contextualized prompts are able to elicit input-relevant knowledge from PLMs to enrich the input text. Furthermore, to enhance the relevance of the generated text to the inputs, we adopt a continuous inverse prompting to refine the forward generation process by modeling an inverse generation process from output to input. We also propose a lightweight method for efficient training. We have constructed extensive experiments on three generation tasks and the experimental results have demonstrated the effectiveness of our model on fine-tuning PLMs for text generation task.

References

[Banerjee and Lavie, 2005] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Dou *et al.*, 2021] Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. Gsum: A general framework for guided neural abstractive summarization. In *NAACL-HLT*, 2021.

[Fan *et al.*, 2018] Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. In *ACL*, 2018.

[Gao *et al.*, 2021] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *ACL/IJCNLP*, 2021.

[Hu *et al.*, 2017] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In *ICML*, 2017.

[IV *et al.*, 2021] Robert L. Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*, 2021.

[Jiang *et al.*, 2020] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 2020.

[Kikuchi *et al.*, 2016] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *EMNLP*, 2016.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

[Lewis *et al.*, 2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP*, 2021.

[Li *et al.*, 2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*, 2016.

[Lin, 2004] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[Liu *et al.*, 2021] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[Narayan *et al.*, 2018] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP*, 2018.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[Qin and Eisner, 2021] Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. In *NAACL-HLT*, 2021.

[Radford *et al.*, 2019] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 2020.

[Ribeiro *et al.*, 2020] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. *arXiv preprint arXiv:2007.08426*, 2020.

[Schick and Schütze, 2021] Timo Schick and Hinrich Schütze. Few-shot text generation with natural language instructions. In *EMNLP*, 2021.

[Shin *et al.*, 2020] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*, 2020.

[Yu *et al.*, 2020] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *arXiv preprint arXiv:2010.04389*, 2020.

[Zhang *et al.*, 2020] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *ACL (demo)*, 2020.

[Zhong *et al.*, 2021] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: learning vs. learning to recall. In *NAACL-HLT*, 2021.

[Zou *et al.*, 2021] Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. Controllable generation from pre-trained language models via inverse prompting. In *KDD*, 2021.

Appendix

We offer some experiment-related information as supplementary materials to help readers understand and reproduce our model. The appendix is organized into five sections:

- Additional details for our datasets are presented in Appendix A;
- Additional descriptions and implement details for our baseline methods are presented in Appendix B;
- Additional result of different methods on XSUM dataset is presented in Appendix C;
- The results of model sensitivity *w.r.t.* the number of prompt vectors k and the hyper-parameter λ are presented in Appendix D; and
- Additional details for human evaluation are presented in Appendix E;

A Construction of the Datasets

We evaluate our model on three datasets for different text generation tasks: WRITINGPROMPTS for story generation, REDDITGEN for opinion generation, and XSUM for summarization.

- WRITINGPROMPTS consists of pairs of story premise and response from WritingPrompts forum, which are taken as input text and output text, respectively. We utilize the dataset provided in the page[1].
- REDDITGEN contains pairs of original post (OP) statement on a controversial issue about politics, which are collected from Reddit[2]. We use the OP title, which contains a proposition (*e.g., all public schools should have a uniform dress code*), as the input text, to generate the full post.
- XSUM contains BBC articles covering a wide variety of subjects with professionally written summaries. We utilize the dataset provided by in the page [3].

In REDDITGEN and WRITINGPROMPTS, some of the output posts and story responses are significantly long (more than 512 tokens). Due to the length limitation of PLMs, we discard examples where text contains more than 512 tokens. We summarize the statistics of three datasets after preprocessing in Table 5.

B Baseline Methods

We present the descriptions and implement details of baselines in the following:

- *Fusion model*: It integrates a language model with a convolutional sequence-to-sequence model by fusion mechanism. The fusion mechanisms can help seq2seq model build dependencies between their input and output. We utilize the codes provided in the page[4].
- *GPT-2*: It is a large transformer-based language model trained on a dataset of 8 million web pages. GPT-2 is trained with a simple objective: predict the next word, given all of the previous words within some text. We utilize the codes based on Hugging Face[5].
- *BART*: It uses a standard sequence-to-sequence architecture with a bidirectional encoder (like BERT) and a left-to-right decoder (like GPT-2). The pretraining task of BART involves randomly shuffling the order of the original sentences and a novel in-filling scheme, where spans of text are replaced with a single mask token. We utilize the codes based on Hugging Face[6].
- *T5*: It is an encoder-decoder model pretrained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. T5 works well on a variety of tasks out-of-the-box by prepending a different prefix to the input corresponding to each task. We utilize the codes based on Hugging Face[6].
- *Prefix-Tuning*: It is a lightweight alternative to fine-tuning PLMs for text generation tasks, which keeps language model parameters frozen, but optimizes a small continuous task-specific vector (called the prefix). We utilize the codes provided by in the page[7] and adopt the base version of BART.
- *Prompt Tuning*: It only fine-tunes the prompts concatenated to the input and freeze the PLM. In NLU tasks, this method can achieve satisfactory results compared with fine-tuning the whole model when the PLM is large enough. We implement this method using our own code.

To compare with existing results, we utilize the base version of all PLMs for story generation and opinion generation tasks and the large version for summarization task.

C XSUM Result

We have presented the results on two open-ended text generation tasks in Section 4.2. Although our context tuning is designed to encode rich context information for generation, we also present the result on summarization task. The results of different methods on XSUM dataset are shown in Table 6. The performance on XSUM dataset has a same tendency of two open-ended generation tasks, which proves the effectiveness of our method on general generation tasks.

D Model Sensitivity

In this part, we construct sensitivity analyses *w.r.t.* the number of prompt vectors k and the hyper-parameter λ on WRITINGPROMPTS dataset.

Model Sensitivity *w.r.t.* The Number of Prompt Vectors k. In contextualized prompts learning, the number of prompt

[1] https://github.com/pytorch/fairseq/tree/master/examples/stories

[2] https://www.reddit.com/r/changemyview/

[3] https://github.com/EdinburghNLP/XSum

[4] https://github.com/pytorch/fairseq/tree/master/examples/stories

[5] https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-generation

[6] https://github.com/huggingface/transformers/tree/master/examples/pytorch/summarization

[7] https://github.com/XiangLi1999/PrefixTuning

Dataset	#Train	#Valid	#Test	#Input	#Output
WRITINGPROMPTS	67,765	3,952	3,784	30.2	281.2
REDDITGEN	42,462	6,480	7,562	23.2	121.6
XSUM	204,045	11,332	11,334	374.2	21.1

Table 5: Statistics of our datasets after preprocessing. #Train, #Valid and #Test denote the number of examples in training, valid and test datasets, respectively. #Input and #Output denote the average number of tokens in the input text and output text.

Datasets	Models	R-1	R-2	R-L
XSUM	**BART**	45.14	22.27	<u>37.25</u>
	Prompt Tuning	<u>45.52</u>	<u>22.47</u>	37.23
	Context-Tuning	**45.56**	**22.57**	**37.42**
	Lightweight fine-tuning			
	Prefix-Tuning	43.80	22.27	36.05
	Prompt Tuning	40.65	17.48	32.40
	Context-Tuning	41.18	18.07	32.98

Table 6: Performance comparisons of different methods for summarization generation task. R-n is short for ROUGE-n. Bold and underline fonts denote the best and the second best methods.

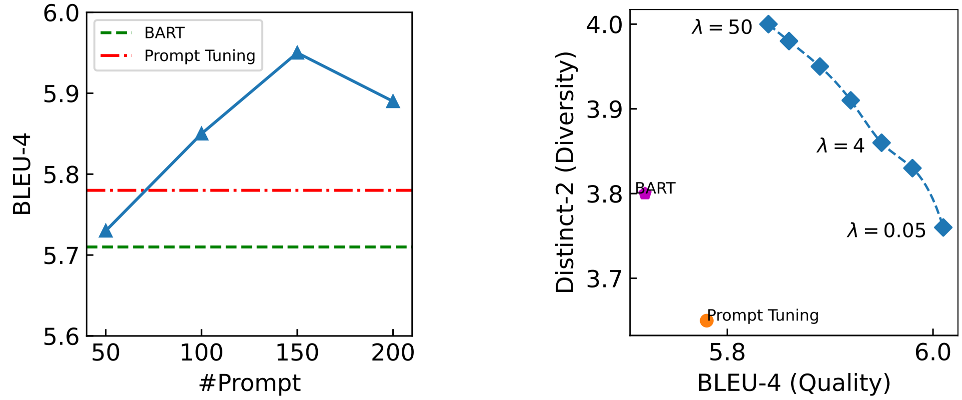
vectors is a key factor that influences the performance of our model. A longer sequence of prompt vectors means more trainable parameters, and therefore more expressive power. Here, we would examine how it affects the final performance of our Context-Tuning. Given the complete examples in Table 5, we vary the number of prompt vectors in the set $\{50, 100, 150, 200\}$. We separately train our model with different numbers of prompt vector, and report the performance on the test set. As shown in Fig. 2(a), the performance of our model gradually improves as the number of prompt vectors increases up to a threshold (150 for opinion generation), and then a slight performance drop occurs. More important, our model achieves the best performance over baselines with 150 prompt vectors.

Model Sensitivity *w.r.t.* The Hyper-Parameter λ. To generate the next sequence, we combine the forward probability $\Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c)$ and the backward probability $\Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i)$ in Eq.6 to select the best candidate sentence with a balance hyper-parameter λ. Therefore, to examine the effect of λ on generation performance, we measure the diversity-quality trade-off of our context-tuning by gradually increasing the value of λ from 0.05 to 50. We construct a similar evaluation experiment as that for the number of prompt vectors. In Fig. 2(b), the value of λ decreases from top left to bottom right. We can see that our model is consistently better than the three selected baselines in terms of diversity (high Distinct-2 results) and quality (high BLEU-4 results). This result implies that, the larger λ is, the more diversity generated text becomes.

E Human Evaluation

We have introduced four factors to evaluate the generated text in Sec 4.4, including informativeness, relevance, coherence, and fluency. Here, *informativeness* means that how much the generated text provides useful and meaningful information, *relevance* means that how relevant the generated text is according to the input contexts, *coherence* evaluates how content is coherent considering both intra- and inter-sentence correlation, and *fluency* means that how likely the generated text is produced by human.

The detailed evaluation guideline and examples are listed in Fig. 3, Fig. 4, Fig. 5 and Fig. 6.



(a) Varying the number of prompt vectors.

(b) Varying the value of hyper-parameter λ.

Figure 2: Performance tuning on WRITINGPROMPTS dataset.

Thank you for taking time out of your busy schedule to participate in our scientific research evaluation! Our research work is to let the machine generate corresponding story, for a given title, and hope that it is as close as possible to what humans write. Hence, we need to evaluate whether it meets the standards that people think.
In this task, you will see a title, idea or introduction such as:
You have a very shitty type of precognition .
Then you'll see a corresponding story or comment, either written by a human or a machine, but you don't know which one was written, and you'll have to rate it based on the following standard:

- **Turing Test:** whether the text was written by a human;
- **Fluency:** whether the text has good form, logical reading and smooth sentences;
- **Informative:** whether the text contains meaningful content and will not be boring to read;
- **Relevance:** whether the text is highly relevant to the input context;
- **Coherence:** whether the logic is coherent and not contradictory;

For Turing Test, just judge the text by instinct. For the other four factors, we adopt a 5-point likert scale as the scoring mechanism, in which 5-point means "very satisfying", and 1-point means "very terrible".

Figure 3: Evaluation guidelines for WRITINGPROMPTS dataset.

Fluency: whether the text has good form, logical reading and smooth sentences.

- **1 point example:** I was a exterminator on earth. I killed every insect and animal I killed in my life. If I win, I go to heaven, lose, and go to hell. I was an exterminator in hell. I was the exterminator of the world. I exterminated every insect, animal, and animal that I killed. I didn't have to kill them all. I just had to do it. I had to kill all of them. I couldn't let them get away with it. So I killed them all, and I did it all over again. And again and again. I don't know how long I've been doing this, but it's been a long time. **(containing some grammatical errors and repetitive contents)**
- **1 point example:** I followed you into the dark . We carved our names into the bark . " Forever and always Through short nights and through long days . " But that tree withered so long Its voice died in sweet silent song . Dead and gone We left and lost our life bond . I always remember your words Of comfort and joy like the birds . Sweet and sad ; It was all I never had . For I carved both names on that tree No kind woman would be with me . No more hope ; You and me , the end of a rope You were not ever real and I know that now . " Always to love you " , I end with that vow . **(there is no continuity between the words of the sentence, and the content is intermittent)**
- **3 point example:** I've been trying to kill my master for years. I've tried to kill him for years, but he's always been there for me. He's the only one who knows what I'm going to do, and I don't care. I 'll kill him if I have to. But I can't do it anymore. I haven't been able to do it for years now. I can not do it any more. I just want to go back to my master. I want to be with him again. But he won't let me go back. I know it's not fair, but I just need to get back to him. **(each sentence is grammatically correct and fluent, but contains certain repetitions and discontinuities in semantics)**
- **3 point example:** It's been a long time since I've seen her. She's always been there for me. I'm not sure how long I have been here, but I know she's here. I know I 'll never see her again. I don't know if she 'll ever see me again. But I know it's time. I can feel it in my bones, in my skin, in the bones of my bones. I can't help but think of her. I remember her when I first met her, when I was young. She was so beautiful, so full of life. I couldn't wait to meet her again, to see her smile again. **(sentences are fluent, but similar words are used repeatedly in the sentence, resulting in ambiguous meaning and confusing)**
- **5 point example:** Long ago his heart had warmed , three thousand years - long enough to mourn , the deeds of past and of damnation , stripped of humanity and of his station . He resided in the pits of hell the oldest friend of satan , waiting as the centuries pass watching hells inflation , resting on brimstone as passing devils chatter and laugh , who is this old man and what sin has made him . a curious young man with a glint in his eye asks his sentence , and with creaks and groans the old man rose for the first time in ages , he look at the spirit and with a heavy sigh he came out with , I 'm god and I made this .
- **5 point example:** Tell us your faults ? Really ? This was the question - the shibboleth - that unlocked the cosmos ? The Masters could have picked a scientist to answer but they feared she might mask ignorance . They could have picked from our global leaders bit they feared that they would mask deceit . They could have picked a holy man but feared he would mask violence , oppression , hate , intolerance ... the list of disqualifying sins was almost too long to enumerate . So they picked Josh Thornton , a 45 year old MBA in human resources . " Our greatest weakness ? Well , I think we work a little too hard and , as a race , we might be a bit of a perfectionist . "

Figure 4: Evaluation examples for *fluency* factor.

Informative: whether the text contains meaningful content and will not be boring to read.

- **1 point example:** I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. I've been doing this for years. **(sentence repetition)**
- **1 point example:** " I'm the First ****bender. " " What's that? " I asked. " I don't know, " he replied. " But you're the first ****bender, right? " " Yeah, " I said. " You're not the first. " " Well, I guess that's true. " He paused. " What do you mean? " He looked at me. " That's right. " I looked at him. " It's not like you've ever been the first... " He trailed off. " No, it's just... " I trailed off, " I've never been the First... " **(conversation sentences are very short, and they are all safe replies)**
- **3 point example:** " What the hell is this? " " I don't know, " I said. " I'm not sure what to do with this. I've been doing this for a long time, and it's not like I 'll be able to do anything about it. " He looked at me with a confused look on his face. " What the fuck? " I asked. " You're not going to do this again. " I replied. " It's just a matter of time, " he said. I looked at the frying pan again. " Don't worry, it 'll work. It 'll do. " " What's the problem? " He asked. I nodded. " Well, I guess I 'd better get out of here. " The frying pan buzzed at me and text appeared reading " level 18 cooking required to use object. " **(containing rich dialogue, but the content of the dialogue is meaningless)**
- **3 point example:** The stars fell like rain, and we danced. The moon was bright, and the stars danced. The stars were bright, but the stars were not bright. They were bright and the sky was blue. We danced, but we didn't know what to do. We danced and we sang and we laughed and we talked and we cried and we screamed and we played and we giggled and we thought we were going to die, but it wasn't. It was just the stars falling like rain. And we danced, and it was beautiful. **(the article has a certain content at the beginning, but it is more verbose at the end)**
- **5 point example:** She is the woman you have loved since the day you met her . Everyday she has a smile on her face beautiful as ever . You love her but are afraid of what she will say when you tell her . It was raining and you ran for shelter , a small roof at the bus stop . Tired and panting you barely notice her sitting beside you , she calls your name . You jump a little bit and become nervous when you recognize her . You stare at each other not knowing what to do or say , and then she kisses you . An alarm sounds , you wake up in your room all alone , another dream **(a whole story)**
- **5 point example:** I feel like it 's worth pointing out that a lot of these are kind of situational . I think all of these rules are good to follow if what you 're writing is something you 're trying to submit to someone formally or something , but a lot of them are not important in casual writing (such as someone 's dialogue or something like that) . <u>For example , " literally " has been used for hyperbole for a very long time</u> – I know I 've heard that Nathaniel Hawthorne did it , and I hardly think he was the first . It pisses a lot of people off but it 's not like it 's a new phenomenon and it 's not like it 's a corruption of the language . Things about spelling and homonyms and stuff should probably be followed just about all the time though . **(using concrete examples)**

Figure 5: Evaluation examples for *informative* factor.

Coherence evaluates how content is coherent considering both intra- and inter-sentence correlation.

- **1 point example:** i don't know if it's a good thing or a bad thing, but i've found that if i'm going to write a story, i need to be able to get it out of my head. i think that's the most important thing. if i want to write something, i have to know what i want it to be. i have a lot of ideas, but they're just not good enough. i'll try to find a way to get them out, but if i can't find the right words to write, i will probably have to go back and re - write it. **(no semantic connection between sentences)**
- **1 point example:** " I'm sorry, " I said to myself. " I don't know what to do. " I replied. " I just want to see you again. " I looked at my reflection, and it was the same. I couldn't tell if it was a reflection or a reflection, but I knew that it was my reflection. I looked back at the mirror, and I saw that I was the one who was going to be my future soulmate. I smiled, and said, " I love you, " and walked away.
- **3 point example:** I have a very shitty type of precognition. I don't know what it is, but it's something I have to do. I'm not sure why I do it, but I do. I've been doing it for years now, and I haven't been able to figure out why. It's not like I have any control over it. I can't control what I do, or what I say, or how I act. I can only control the way I act, how I react, and how I feel. I have no control over my actions, and no control of my emotions. I just have to control my emotions, and that's all I can do. **(after careful consideration, many logical contradictions were found)**
- **3 point example:** I 'd like to take a moment to appreciate Sir Terry Pratchett. I've read a lot of his work, and I'm not sure if it's because of his writing style or because of the way he wrote it. I don't know if he's a good writer, or if he is a bad writer, but I do know that he is one of the best writers I have ever read. I think that's why I love him so much. I also think that he has a great sense of humor, and that he doesn't have a bad sense of humour. **(some repeated information, but other content is ok)**
- **5 point example:** You eagerly await your pizza to come because you ordered from this new Italian Pizza owed by two brother , you remember that one of their names are Mario but you forgot the other . The Pizza finally arrives a bit late from this tall guy dressed in green . You pay him take , take the pizza but forget to tip . When you start eating you get a bit dizzy so you lay down and fall asleep quite quickly . You wake up in a in a place covered in mushrooms with a little man dressed as a mushroom telling you that " You need to save the princess " . **(smooth connection between context)**
- **5 point example:** When 1st purge happened , no one thought people would attack each other . A desperate party know only as Al Queda broke the rules and decided that it would do what no one else would have done . Bomb Manhattan . That single move destroyed not only the Republicans and the Democrats , it also destroyed morale . Hundreds of fully armed fat Politicians fled to the streets , screaming out jibberish and shooting anyone they see . Millions lay dead as all parties Jump onto their jets towards Manhattan , preparing to be included in the giant Cesspit of a war know as the Purge . When the Morning came . There were no victors . Only that the red dawn came and claimed .

Figure 6: Evaluation examples for *coherence* factor.