

Context-Tuning: Learning Contextualized Prompts for Natural Language Generation

Tianyi Tang^{1,4}, Junyi Li^{1,3}, Wayne Xin Zhao^{1,4,5}✉ and Ji-Rong Wen^{1,2,4}

¹Gaoling School of Artificial Intelligence, Renmin University of China

²School of Information, Renmin University of China

³DIRO, Université de Montréal

⁴Beijing Key Laboratory of Big Data Management and Analysis Methods

⁵Beijing Academy of Artificial Intelligence, Beijing, 100084, China

steventianyitang@outlook.com lijunyi@ruc.edu.cn batmanfly@gmail.com

Abstract

Recently, pretrained language models (PLMs) have had exceptional success in language generation. To leverage the rich knowledge encoded by PLMs, a simple yet powerful paradigm is to use *prompts* in the form of either discrete tokens or continuous embeddings. In existing studies, these prompting methods are typically independent of the inputs, lacking sufficient consideration of input semantics. 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 to elicit useful knowledge from PLMs for generation. We refer to such prompts as *contextualized prompts*. Secondly, we use *continuous inverse prompting* to improve the process of natural language generation by modeling an inverse generation process from output to input, making the generated text more relevant to the inputs. Furthermore, we utilize a lightweight context-tuning method that fine-tunes only 0.12% of the parameters while maintaining good performance. Our code is publicly available at <https://github.com/RUCAIBox/Context-Tuning>.

1 Introduction

Natural language generation (*a.k.a.* text generation) aims to produce plausible and readable text in human language from input data (Li et al., 2022). Recently, large-scale pretrained language models (PLMs) such as BART (Lewis et al., 2020) have had exceptional success in language generation. To leverage the encoded 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 continuous embeddings (called *prompts*). Following this paradigm, this work aims to study how to develop

✉ Corresponding author

Title	<u>Live-action medium</u> is inferior to <u>animation medium</u>
Static Prompts:	Write a story about: Title
Contextualized Prompts:	$p_{1\dots k}$ Title $p_{k+1\dots 2k}$
Story	I think that live-action works can't be considered art. They feel more like documentaries or theater pieces with CGI combined. The superiority of animated works is that they are more abstract and imaginative and characters show more emotion and variety of designs.

Table 1: Example inputs (titles) and outputs (stories) of generation dataset CMV. Static prompts are human-written instructions, which are independent of input titles. $p_{1\dots k}$ and $p_{k+1\dots 2k}$ are contextualized prompts, which are derived conditioned on the input title. The wavy line and underline denote the corresponding information between input and output.

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 the summarization task. Recent work has further proposed utilizing continuous prompts (Li and Liang, 2021; Lester et al., 2021) for text generation. Continuous prompts consist of trainable parameters that do not correspond to real tokens and can be easily optimized during fine-tuning. However, existing prompting approaches typically adopt *static prompts* for generation, *i.e.*, the prompts contain task-related information but remain the same for different input texts.

In this work, we mainly focus on challenging open-ended generation, such as story generation (Fan et al., 2018) and review generation (Li et al., 2019). Under this setting, the input text usually contains very limited information, while the task goal is to generate an output sequence containing informative contents based on the given limited input. The example in Table 1 aims to gen-

erate a story about the topics of “*live-action*” and “*animation*”. In such a case, it requires in-depth background knowledge about the two topics. As we can see, static prompts such as “*Write a story about:*” are independent of the input title, making it difficult to capture the related aspects for this generation task. Instead of static prompts, we argue that *contextualized prompts* (as shown in Table 1) derived based on the input title will be more suited for this setting.

To address the above issues, we propose **Context-Tuning**, a novel continuous prompting approach to fine-tuning PLMs for natural language generation. Our approach has three major technical contributions. Firstly, the prompts are derived based on input text to enrich the input by eliciting related knowledge from PLMs. Specifically, by concatenating limited input and a sequence of “[MASK]” tokens into BERT (Devlin et al., 2019), we leverage its excellent mask-filling ability to predict these tokens, and the last hidden state of tokens can be used as prompt vectors. Since the prompts are highly related to the input context, we refer to them as *contextualized prompts*. Secondly, to further enhance the relevance between the generated text and the input text, we extend inverse prompting (Zou et al., 2021) by incorporating continuous prompts. We refer to them as *continuous inverse prompting*. By maximizing the likelihood of predicting inputs conditioned on the generated text and continuous prompts, context-tuning can generate texts highly relevant to the input text. Moreover, to ease the training burden, we propose to use a lightweight context-tuning method (Ben Zaken et al., 2022) that only fine-tunes the bias term of all model parameters. In this way, we can achieve a comparable performance (98.0% of the full-tuned performance) by only tuning 0.12% of the parameters, compared to full-tuned context-tuning.

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

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., 2021). As mentioned in Liu et al. (2021), controlled text generation is relevant to our input-dependent method. The goal of controlled text generation is to direct the generated texts into specific styles (Hu et al., 2017), lengths (Kikuchi et al., 2016), or keywords (Dou et al., 2021). In contrast, our contextualized prompts elicit knowledge from PLMs to enrich the input rather than control the specific properties of generated text.

Prompting Learning. Prompting methods prepend task instructions to the input and generate the output from PLMs. Most typical methods utilize manually designed task-specific prompts to adapt to different generation tasks (Radford et al., 2019; Raffel et al., 2020). However, it is time-consuming and laborious to construct human-written prompts for various generation tasks. As a result, recent research has concentrated on automating the search for discrete prompts (Shin et al., 2020; Gao et al., 2021). Nonetheless, searching for prompts over discrete space is challenging to optimize due to the non-differentiable issues and continuous nature of neural networks. To handle these problems, many studies propose optimizing continuous prompts (Lester et al., 2021; Li and Liang, 2021), which are more expressive and flexible for any task. 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 enhance relevance further.

Most existing prompting methods (Schick and Schütze, 2021a; Shin et al., 2020; Lester et al., 2021) focus on NLU tasks, which are choice questions that can easily be converted into filling

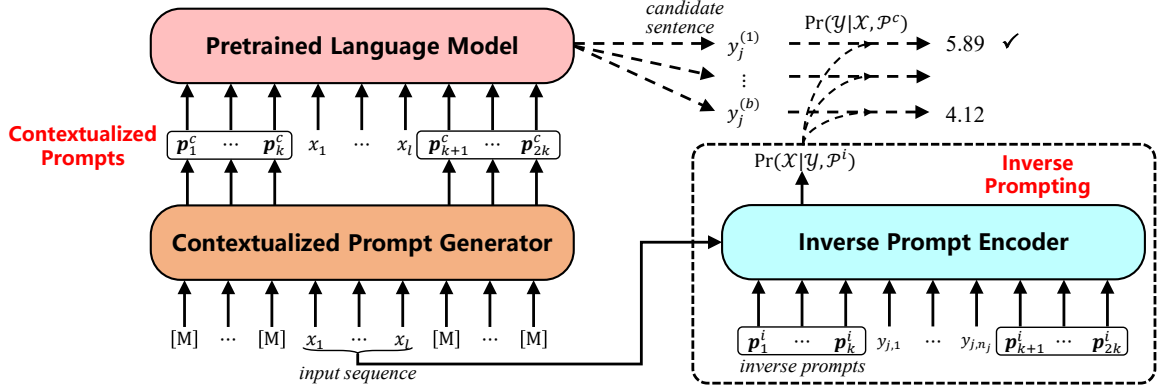


Figure 1: The overview of the proposed context-tuning. “[M]” denotes the mask token “[MASK]”. By combining the forward probability $\Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c)$ (the left part) and backward probability $\Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i)$ (the right part), we select the sequence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

“[MASK]” tasks. However, text generation aims to generate a sequence of tokens, in contrast to a few options in limited space. For example, prefix-tuning (Li and Liang, 2021) and GENPET (Schick and Schütze, 2021b) 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 full tuning settings. In contrast, our context-tuning can improve performance under full tuning settings, and the lightweight strategy tunes only 0.2% of the parameters while retaining good performance.

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 the generated text to the input, we utilize 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 $\mathcal{X} = \langle x_1, \dots, x_l \rangle$. The output text is usually composed of multiple sentences: $\mathcal{Y} = \{y_j : \langle y_{j,1}, \dots, y_{j,t}, \dots, y_{j,n_j} \rangle\}_{j=1}^m$. In context-tuning, we introduce contextualized prompts $\mathcal{P}^c = \langle p_1^c, \dots, p_k^c \rangle$ into the input side. Thus, the prompt-based generation task can be formulated as:

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

3.1 Contextualized Prompts

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

Masked Prompt Learning. Specifically, unlike prefix-tuning (Li and Liang, 2021), which 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 masked language modeling task of BERT (Devlin et al., 2019), we use 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 \mathcal{X} as:

$$\tilde{\mathcal{X}} = [\text{MASK}]_{1,\dots,k}, \mathcal{X}, [\text{MASK}]_{k+1,\dots,2k}. \quad (2)$$

By feeding $\tilde{\mathcal{X}}$ as the input of the prompt generator, we can obtain the top-layer representations of these “[MASK]” tokens:

$$\underbrace{\tilde{p}_1^c, \dots, \tilde{p}_k^c}_{\text{prefix prompts}} / \underbrace{\tilde{p}_{k+1}^c, \dots, \tilde{p}_{2k}^c}_{\text{suffix prompts}} = \text{Prompt-Generator}(\tilde{\mathcal{X}}). \quad (3)$$

After shown in Section 4.4, we set k to 150 with the best performance. Compared with randomly-initialized prompts, our BERT-based prompt learning method can better learn 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:

$$\Pr(w|\tilde{\mathcal{X}}) = \text{softmax}(\mathbf{W}^V \tilde{\mathbf{p}}_k^c), \quad (4)$$

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

$$\mathbf{p}_k^c = \mathbf{E} \cdot \Pr(w|\tilde{\mathcal{X}}). \quad (5)$$

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. Compared with existing continuous prompts, our contextualized prompt vectors can better correspond to real word embeddings in semantic space, as shown in Section 4.6.

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:

$$\begin{aligned} \mathcal{L}_c &= -\log \Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c) \\ &= -\log \Pr(\mathcal{Y}|\mathbf{p}_{1\dots k}^c, \mathbf{x}_{1\dots l}, \mathbf{p}_{k+1\dots 2k}^c), \end{aligned} \quad (6)$$

where $\mathbf{p}_{1\dots k}^c$ denotes $\mathbf{p}_1^c, \dots, \mathbf{p}_k^c$, $\mathbf{x}_{1\dots l}$ denotes $\mathbf{x}_1, \dots, \mathbf{x}_l$, and $\mathbf{p}_{k+1\dots 2k}^c$ denotes $\mathbf{p}_{k+1}^c, \dots, \mathbf{p}_{2k}^c$. By leveraging the encoded knowledge from PLM, the contextualized prompts are helpful to generate informative output texts.

3.2 Continuous Inverse Prompting

Although contextualized prompts can improve the informativeness of output, it still suffers from the off-topic generation issue 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 the previous inverse prompting that depends on artificial construction (Zou et al., 2021), our inverse prompting is based on continuous prompts, which can be flexibly optimized during fine-tuning.

Algorithm 1 The algorithm procedure for generation process of context-tuning.

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

- 1: **Input:** An input sequence \mathcal{X}
- 2: **Output:** A generated sequence \mathcal{Y}
- 3: Initialize step $j = 0$
- 4: **while** $j < n_m$ **do**
- 5: Derive contextualized prompts \mathcal{P}^c based on \mathcal{X}
- 6: Generate b candidate sentences $y_j^{(1)}, \dots, y_j^{(b)}$ according to Eq. 6
- 7: Utilize continuous inverse prompts \mathcal{P}^i to compute the likelihood of candidate sentences according to Eq. 7
- 8: Choose the best sentence as y_s based on Eq. 8
- 9: Terminate the loop if y_s contains the end of sentence token
- 10: Update $j = j + 1$
- 11: **end while**
- 12: Concatenate y_1, \dots, y_j as generated sequence \mathcal{Y}
- 13: **return** \mathcal{Y}

Output-to-Input Relevance Enhancement. To model the relevance of output \mathcal{Y} to input \mathcal{X} , we hypothesize 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 utilize prompts to mitigate this issue. We introduce continuous 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:

$$\begin{aligned} \mathcal{L}_i &= -\log \Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i) \\ &= -\sum_{j=1}^m \log \Pr(\mathcal{X}|\mathbf{p}_{1\dots k}^i, \mathbf{y}_{j,1}, \dots, \mathbf{y}_{j,n_j}, \mathbf{p}_{k+1\dots 2k}^i), \end{aligned} \quad (7)$$

where $\mathbf{p}_{1\dots k}^i$ denotes $\mathbf{p}_1^i, \dots, \mathbf{p}_k^i$ and $\mathbf{p}_{k+1\dots 2k}^i$ denotes $\mathbf{p}_{k+1}^i, \dots, \mathbf{p}_{2k}^i$. Unlike contextualized prompts in Section 3.1, we expect inverse prompts to reflect better the relationship between \mathcal{Y} and \mathcal{X} , which is dependent on the task rather than the input. Thus, the inverse prompts are static and continuous in our approach.

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

$$\mathcal{Y} = \underset{\mathcal{Y}}{\operatorname{argmax}} \log \Pr(\mathcal{Y}|\mathcal{X}, \mathcal{P}^c) + \lambda \log \Pr(\mathcal{X}|\mathcal{Y}, \mathcal{P}^i), \quad (8)$$

where λ is a hyper-parameter to balance these two probabilities. 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 makes the generation process more controllable. Even for latter generated sentences, it can still enforce them to adhere to the input topic.

3.3 Discussion and Learning

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

Discussion and Comparison. We use contextualized prompts (Eq. 6) 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 static prompts, which are irrelevant to the input. Besides, we propose continuous inverse prompting (Eq. 7) to enforce the relevance of long output text by considering a generation process from output to input. Different from the original inverse prompting, our inverse prompting is based on continuous prompts, which can be optimized during fine-tuning.

Considering that our method involves another PLM and more parameters, we propose a lightweight context-tuning approach. Following Ben Zaken et al. (2022), we only fine-tune the bias term of each parameter, resulting in fine-tuning only 0.12% of the parameters of complete 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 parameters of prompts. Prefix-tuning fine-tunes prompts in each layer and tunes 16.4% of the BART parameters, while prompt tuning only fine-tunes the prompt concatenated to the input, resulting in fine-tuning 0.05% of the BART parameters.

Optimization. We use 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. 8 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. 6. Meanwhile, we optimize $\Theta^{(i)}$ according to the inverse generation loss using Eq. 7. During inference, we combine them and select sentences that are both

Dataset	#Train	#Valid	#Test	#Input	#Output
WP	53,516	4,000	2,000	25.48	150.64
ROC	176,688	9,816	4,909	9.02	40.72
CMV	42,462	6,480	7,562	17.89	104.10
WIKIP	69,288	8,661	8,662	3.38	194.72

Table 2: 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 and output text.

informative and relevant to the input text based on Algorithm 1 and Eq. 8.

4 Experiment

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

4.1 Experimental Setup

4.1.1 Construction of the Datasets

To measure the performance of our proposed context-tuning, we evaluate it on four open-ended text generation tasks: WRITINGPROMPTS (WP), ROCSTORIES (ROC), CHANGEMYVIEW (CMV), and WIKILOTS (WIKIP). Specifically, WP (Fan et al., 2018) consists of pairs of story premises and responses from the WritingPrompts forum. ROC (Mostafazadeh et al., 2016) is a dataset consisting of five-sentence commonsense stories. Here, we use the first sentence as the input to generate the following four sentences. For WP and ROC, we utilize the version provided by Guan et al. (2021) for a fair comparison. CMV (Hua and Wang, 2020) contains pairs of post statements on a controversial issue, which are collected from Reddit. WIKIP¹ is a collection of story plots from Wikipedia. We use the sub-header word to generate the full story.

Since some dataset outputs are significantly long, we discard examples where the text contains more than 512 tokens due to the length limitation of PLMs. We summarize the statistics of four datasets after preprocessing in Table 2.

4.1.2 Baseline Methods

We consider the following baselines as comparisons: GPT-2, BART, T5, HINT, prefix-tuning, and prompt tuning. Among these baselines, GPT-2 (Radford et al., 2019), BART (Lewis et al.,

¹<https://github.com/markriedl/WikiPlots>

Models	WRITINGPROMPTS				ROCSTORIES				#Para
	BLUE-1	BLUE-2	Dist-1	Dist-4	BLUE-1	BLUE-2	Dist-1	Dist-4	
	<i>Full fine-tuning</i>								
GPT-2	24.94	9.03	1.40	35.38	31.45	14.26	2.21	58.63	1.2×10^8
T5	20.76	7.41	1.25	27.77	31.31	14.23	2.22	54.31	2.2×10^8
BART	<u>28.42</u>	<u>11.31</u>	<u>2.11</u>	<u>62.05</u>	32.95	15.35	<u>2.70</u>	68.88	1.4×10^8
HINT	22.40	8.40	–	31.30	<u>33.40</u>	<u>15.40</u>	–	<u>69.30</u>	1.4×10^8
Context-Tuning	29.88	11.85	2.49	67.78	34.65	16.60	3.16	75.53	2.5×10^8
	<i>Lightweight fine-tuning</i>								
Prompt Tuning	16.26	5.18	3.71	69.68	27.27	10.49	2.44	62.12	7.6×10^4
Prefix-Tuning	<u>28.39</u>	<u>10.76</u>	1.72	58.34	<u>30.62</u>	<u>13.51</u>	2.51	<u>67.19</u>	2.3×10^7
Context-Tuning	29.45	10.90	<u>1.78</u>	<u>62.89</u>	32.24	14.30	<u>2.49</u>	68.92	3.0×10^5
	CHANGEMYVIEW				WIKILOTS				
	BLUE-1	BLUE-2	Dist-1	Dist-4	BLUE-1	BLUE-2	Dist-1	Dist-4	
	<i>Full fine-tuning</i>								
GPT-2	23.39	8.32	0.75	37.18	23.74	9.33	0.90	38.39	1.2×10^8
T5	20.89	7.79	<u>1.01</u>	42.48	14.83	6.09	1.33	39.25	2.2×10^8
BART	<u>25.69</u>	<u>9.77</u>	1.11	61.21	27.12	11.54	<u>1.82</u>	<u>49.54</u>	1.4×10^8
Context-Tuning	26.11	10.00	0.99	<u>57.38</u>	27.80	11.82	2.05	51.96	2.5×10^8
	<i>Lightweight fine-tuning</i>								
Prompt Tuning	22.54	8.11	1.43	64.60	18.64	6.98	2.78	58.53	7.6×10^4
Prefix-Tuning	<u>25.72</u>	<u>9.84</u>	0.96	54.66	27.30	11.60	<u>1.95</u>	<u>51.05</u>	2.3×10^7
Context-Tuning	28.83	10.96	<u>0.97</u>	<u>57.79</u>	<u>26.93</u>	<u>11.53</u>	2.48	59.34	3.0×10^5

Table 3: Performance comparison of different methods for open-ended text generation tasks. Dist- n is short for Distinct- n . Bold and underlined fonts denote the best and second-best methods (the same below). #Para denotes the number of fine-tuned parameters in each method. The results of HINT are from its original paper (Guan et al., 2021). “–” means HINT does not compute the corresponding result.

2020), and T5 (Raffel et al., 2020) are three prevalent PLMs for natural language generation; HINT (Guan et al., 2021) is a strong baseline model specially designed for generating long and coherent texts; prefix-tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) are the recently proposed lightweight models using continuous prompts for generation tasks. We utilize the base version for all PLMs for a fair comparison.

4.1.3 Implementation Details

For fine-tuning settings, we consider two strategies: full fine-tuning and lightweight fine-tuning, to compare our methods with different baselines. In the lightweight fine-tuning settings, our context-tuning only tunes the bias term of each parameter.

In all experiments, we utilize the Adam optimizer and set $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$. We train our model for 20 epochs and utilize the model with the best performance on the validation set for generation. During inference, we apply the nucleus sampling with $p = 0.9$ and temperature

of 0.7. We train our model using NVIDIA A100 GPUs on Ubuntu 18.04 and employ the NLP open-source library Transformers (Wolf et al., 2020) and text generation library TextBox (Li et al., 2021).

4.1.4 Evaluation Metrics

To evaluate the performance of different methods of natural language generation, we adopt two automatic evaluation metrics, including BLEU (Papineni et al., 2002) and Distinct (Li et al., 2016). Specifically, BLEU evaluates the quality of generated and real text, while Distinct measures the diversity of generated texts.

4.2 Performance Comparison

We present the results of different methods on generation tasks in Table 3.

First, we can see that BART performs best compared to other PLMs on these generation tasks. Pre-trained on the large-scale corpus, PLMs can better understand natural language and fluently express human language. We consider the better perfor-

Models	B-1	B-2	D-1	D-4
Context-Tuning	29.88	11.85	2.49	67.78
w/o Continuous w Manual				
- human-written prompt ₁	27.96	10.93	1.92	59.97
- human-written prompt ₂	29.14	11.56	2.09	61.69
w/o BERT w RoBERTa	27.31	10.73	1.56	57.51
w/o Semantic Mapping	29.19	11.33	1.72	58.57
w/o Inverse Prompting	29.31	11.45	2.19	64.09

Table 4: Ablation analysis on WRITINGPROMPTS dataset.

mance of BART is due to the encoder-decoder architecture and the DAE pretraining task. That is the major reason we adopt BART as our base generation model.

Second, the recently proposed continuous prompting methods, prefix-tuning, and prompt tuning do not achieve ideal performance in these tasks. This finding shows that natural language generation tasks are more challenging than NLU tasks. Only fine-tuning a few parameters cannot outperform full fine-tuning.

Finally, our model outperforms all the baselines (including the strong baseline HINT) over four tasks under both full and lightweight tuning settings. The reason is 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 superior results to prefix-tuning, only with 1.3% of the parameters of prefix-tuning and 0.2% of the parameters of BART. Some of the performance under lightweight settings can even outperform the full tuning, which may be a solution to catastrophic forgetting (Wang et al., 2021). And a major reason is that prefix-tuning and prompt tuning adopt static prompts, which are task-specific and unrelated to the context information.

4.3 Ablation Analysis

In this part, we construct ablation experiments to test 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 enhance the relevance of the generated texts further. Here, we would like

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 5: Turing test (TT) and human evaluation on WRITINGPROMPTS. “Gold” indicates the ground-truth texts. Flu., Info., Rel., and Coh. denote fluency, informativeness, relevance, and coherence, respectively.

to examine how each factor contributes to the final performance. To see this, we prepare several variants for a comparison:

- *w/o Continuous w Manual*: the variant removes the continuous prompts but utilizes two kinds of human-written prompts, *i.e.*, prompt₁: “Title: **\$Input** Story:” and prompt₂: “Given the title **\$Input**, please write the following story:”.
- *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. 8) from our proposed context-tuning.

From Table 4, we can see that variants replacing continuous prompts with manual prompts are worse than the model with continuous prompts. The performance of manual prompts is sensitive to different instructions and does not always lead to gains. This verifies the effectiveness of utilizing continuous prompts rather than discrete ones for text generation tasks. The variants replacing the BERT-based prompt generator with RoBERTa are worse than the full model. We further observe a slight performance drop when our method removes the semantic mapping and inverse prompting. This implies that the proposed semantic mapping and continuous inverse prompting approaches can enforce the informativeness relevance of output text.

4.4 Model Sensitivity

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

In contextualized prompt learning, the number

#Prompt	B-1	B-2	D-1	D-4
$k = 50$	27.82	11.00	2.28	63.77
$k = 100$	28.55	11.12	2.17	63.69
$k = 150$	29.31	11.45	2.19	64.09
$k = 200$	28.48	11.22	1.97	61.09

Table 6: Performance tuning on WRITINGPROMPTS dataset. We do not utilize continuous inverse prompting methods here.

of prompt 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 will examine how it affects the final performance of our context-tuning. Given the statistics in Table 2, we vary the number of prompt vectors in the set $\{50, 100, 150, 200\}$. We separately train our model with different numbers of prompt vectors and do not utilize continuous inverse prompting methods for convenience. As shown in Table 6, the performance of our model gradually improves as the number of prompt vectors increases up to a threshold, and then a performance drop occurs. More importantly, our model achieves the best performance with 150 prompt vectors over baselines.

4.5 Human Evaluation

Besides automatic evaluation, we further conduct a human evaluation for testing the effectiveness of our approach. We randomly select 500 input texts from the test set of the WRITINGPROMPTS dataset. We collect the stories generated by GPT-2, BART, T5, and context-tuning, then shuffle them for human evaluation. Following Zou et al. (2021), we invite ten human judges to assign scores to a generated text concerning four factors of quality, namely *informativeness* (how much it provides valuable and meaningful information), *relevance* (how relevant it is according to the input contexts), *coherence* (how coherent both intra and inter sentences are) and *fluency* (how likely a human produces the generated text).

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 design a Turing test where a human judge is asked to distinguish whether a human produces the given text. The detailed evaluation guidelines and examples are listed in Figure 2, Figure 3, Figure 4, and Fig-

ure 5 in the Appendix.

We present the human evaluation results in Table 5. It can be seen that our model is better than the 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 improves the informativeness of the generated text. Moreover, the proposed continuous inverse prompting method enhances the relevance of the generated text to the input.

4.6 Qualitative Analysis

In this part, we present an intuitive analysis of why our model works well.

Table 7 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 is 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 richer semantic information. A primary 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 generating unmeaningful and irrelevant content, such as “I love you” and “language”.

Furthermore, to probe whether our contextualized prompts contain input-relevant knowledge, we find close actual words for a 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 *closer words to contextualized prompts* of Table 7. As we can see, most recalled keywords are included in the story generated by our context-tuning. It shows that our contextualized prompts can better capture input-relevant knowledge. For example, the keywords “forest”, “woods”, and “animal” are closely related to the aspect of *nature*.

Title	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.
Closer words to contextualized prompts: <u>forest trees marching winds food historically story movie pet pikachu animation whilst critical asking write</u>	
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 7: The generated examples of the given title from the WRITINGPROMPTS dataset. Marks in closer words to contextualized prompts and texts generated by context-tuning refer to four chosen aspects, *i.e.*, nature, documentary, Pokémon, and the story generation task.

5 Conclusion

This paper presents a novel continuous prompting approach, *i.e.*, context-tuning, to fine-tuning PLMs for natural language generation. The core idea is to inject input-related context information into continuous prompts, called contextualized prompts, to enhance the informativeness of generation. The contextualized prompts can 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 method 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. Extensive experiments on four generation tasks have demonstrated the effectiveness of our model in fine-tuning PLMs for text generation tasks.

In future work, we will consider integrating more types of context information (*e.g.*, sentiment) to derive more expressive prompts and investigate how our model could be applied to other tasks.

Acknowledgement

This work was partially supported by Beijing Natural Science Foundation under Grant No. 4222027, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098 and Beijing Academy of Artificial Intelligence (BAAI). Xin Zhao is the corresponding author.

References

- Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. [BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. [GSum: A general framework for guided neural abstractive summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. [Making pre-trained language models better few-shot learners](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*,

- pages 3816–3830, Online. Association for Computational Linguistics.
- Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. [Long text generation by modeling sentence-level and discourse-level coherence](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6379–6393, Online. Association for Computational Linguistics.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. [Toward controlled generation of text](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.
- Xinyu Hua and Lu Wang. 2020. [PAIR: Planning and iterative refinement in pre-trained transformers for long text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. [Controlling output length in neural encoder-decoders](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Junyi Li, Tianyi Tang, Gaole He, Jinhao Jiang, Xiaoxuan Hu, Puzhao Xie, Zhipeng Chen, Zhuohao Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2021. [TextBox: A unified, modularized, and extensible framework for text generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 30–39, Online. Association for Computational Linguistics.
- Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2022. [A survey of pretrained language models based text generation](#). *arXiv preprint arXiv:2201.05273*.
- Junyi Li, Wayne Xin Zhao, Ji-Rong Wen, and Yang Song. 2019. [Generating long and informative reviews with aspect-aware coarse-to-fine decoding](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1969–1979, Florence, Italy. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *arXiv preprint arXiv:2107.13586*.
- 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*.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. [A corpus and cloze evaluation for deeper understanding of commonsense stories](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*, 1(8):9.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. [Investigating pretrained language models for graph-to-text generation](#). In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021a. [Exploiting cloze-questions for few-shot text classification and natural language inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021b. [Few-shot text generation with natural language instructions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. [AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. [K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
- Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. [Controllable generation from pre-trained language models via inverse prompting](#). In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21*, page 2450–2460. Association for Computing Machinery.

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 2: 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 3: Evaluation examples for *fluency* 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 5: Evaluation examples for *coherence* factor.