Learning to Transfer Prompts for Text Generation

Junyi Li[1,3,4], **Tianyi Tang**[1], **Jian-Yun Nie**[3], **Ji-Rong Wen**[1,2,4] and **Wayne Xin Zhao**[1,4*]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]School of Information, Renmin University of China
[3]DIRO, Université de Montréal
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods
{lijunyi,steven_tang,jrwen}@ruc.edu.cn
nie@iro.umontreal.ca batmanfly@gmail.com

Abstract

Pretrained language models (PLMs) have made remarkable progress in text generation tasks via fine-tuning. While, it is challenging to fine-tune PLMs in a data-scarce situation. Therefore, it is non-trivial to develop a general and lightweight model that can adapt to various text generation tasks based on PLMs. To fulfill this purpose, the recent prompt-based learning offers a potential solution. In this paper, we improve this technique and propose a novel prompt-based method (PTG) for text generation in a transferable setting. First, PTG learns a set of source prompts for various source generation tasks and then transfers these prompts as target prompts to perform target generation tasks. To consider both task- and instance-level information, we design an adaptive attention mechanism to derive the target prompts. For each data instance, PTG learns a specific target prompt by attending to highly relevant source prompts. In extensive experiments, PTG yields competitive or better results than fine-tuning methods. We release our source prompts as an open resource, where users can add or reuse them to improve new text generation tasks for future research. Code and data can be available at https://github.com/RUCAIBox/Transfer-Prompts-for-Text-Generation.

1 Introduction

In natural language processing (NLP), text generation is an important research topic that aims to automatically produce understandable text in human language from input data (Li et al., 2022). In recent decades, various approaches have been widely applied to a variety of text generation tasks (Li et al., 2019; Gehring et al., 2017; Li et al., 2021a), especially the emergence of pretrained language models (PLMs) (Li et al., 2021c). By involving large-scale parameters pretrained on massive general corpora, PLMs such as GPT-3 (Brown et al., 2020) have achieved substantial progress in text generation. Through the *fine-tuning* paradigm, PLMs can adapt to various text generation tasks by directly adjusting the model parameters with labelled datasets.

However, in real-world scenarios, we are inevitably confronted with tasks having only limited labelled data (*e.g.,* new domains). It is often difficult to fine-tune text generation models in a data-scarce situation (Chen et al., 2020; Li et al., 2021b). Although the input and output formats are different for various text generation tasks, these tasks essentially adopt similar learning and generation mechanism (*e.g.,* Seq2Seq (Sutskever et al., 2014)). Furthermore, the success of PLMs sheds light on the possibility of developing general or transferable text generation models. For example, Radford et al. (2019) framed generation tasks as language modeling by predicting the next token given previous tokens. Based on these studies, we aim to devise a general and lightweight text generation approach that can effectively adapt to various new tasks and datasets, based on PLMs.

To fulfill this purpose, the recently proposed *prompt-based learning* offers a potential technical solution (Liu et al., 2021b). In this paradigm, a text generation task can be solved with the help of a *prompt* containing task-specific information. For example, T5 (Raffel et al., 2020) framed summarization and question answering into a text-to-text format by utilizing prompts "`summarize:`" and "`answer the question:`". Based on learned or manually designed prompts, PLMs can be leveraged to perform existing or new generation tasks without being tuned (Brown et al., 2020; Li and Liang, 2021), which provides a unified approach to utilizing PLMs for various generation tasks. Furthermore, to quickly adapt PLMs to new NLU tasks, several works directly used a soft prompt learned from source NLU tasks to initialize the prompt for a target NLU task (Vu et al., 2021; Su et al., 2021).

*Corresponding author

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3506 - 3518

Inspired by these studies, we aim to apply prompt-based methods to data-scarce text generation tasks in a transferable setting.

Despite promising, there are still two major challenges for transferring prompts in text generation. Firstly, it has been found that prompts are highly task-specific (Gao et al., 2020), and it is difficult to effectively transfer or reuse existing prompts for new tasks. Second, for a single task, even a well-learned prompt may not be suitable for all the data instances from a large population (Scao and Rush, 2021), and hence it is non-trivial to design effective transferring strategy considering both task- and instance-level characteristics.

To address the above issues, we propose **PTG**: **P**rompt Transfer for **T**ext **G**eneration, a novel prompt-based transfer learning approach for text generation. PTG is built upon a transfer learning setting. Specifically, we learn *source prompts* from a number of representative source generation tasks and then transfer these prompts as *target prompts* to perform target generation tasks. The core idea is that these learned source prompts serve as representation bases (*i.e., value vectors* in self-attention mechanism). For each data instance from a new task, we learn a specific target prompt by attending to highly relevant source prompts. To support such an approach, we construct a multi-key memory network storing both source prompts and prompt clusters for key-value prompt finding, and then design an adaptive attention mechanism considering both task- and instance-level information to derive the target prompt. Instead of using a fixed prompt for a new task, our approach is able to effectively learn the most suitable prompt representation from source prompts for a specific data instance. Such an adaptive mechanism considers the specific instance-level features, making our approach more flexible to transfer to new text generation tasks.

To the best of our knowledge, we are the first to introduce the idea of prompting in transfer learning to address text generation tasks. For evaluation, we test PTG on 14 datasets from three sets of text generation tasks: i) *compression* to express salient information in concise text such as summarization; ii) *transduction* to transform text while preserving content precisely such as style transfer; and iii) *creation* to produce new content from input context such as story generation. In both fully-supervised and few-shot experiments, PTG yields competitive or better results than fine-tuning PLMs.

Besides performance benefits, more importantly, we release our source prompts to serve as an open-source prompt library. Researchers can train new task prompts added to our library and reuse these learned prompts to improve unseen text generation tasks. Our library can further act as an analysis tool, such as analyzing what factors influence prompts' transferability across generation tasks and interpreting the task similarity by measuring the corresponding prompt similarity.

2 Related Work

Prompt-based Language Models. Prompt-based learning is a way of leveraging PLMs by prepending task-specific instructions to the task input when feeding into PLMs. Early approaches mainly utilized hand-crafted prompts to adapt to different generation tasks (Brown et al., 2020; Raffel et al., 2020; Zou et al., 2021). However, manually designed prompts are not flexible and cannot be applied to more kinds of new tasks. Thus, recent works have focused on automating the learning of discrete prompts (Shin et al., 2020; Gao et al., 2020). However, learning prompts over discrete space is hard to optimize and likely to be sub-optimal. To address these problems, many works proposed to optimize continuous prompts (Liu et al., 2021c; Li and Liang, 2021), which are more flexible to many kinds of tasks. Among these studies, prefix-tuning (Li and Liang, 2021) prepended a sequence of vectors to the input for text generation tasks. By contrast, we utilize soft prompts to investigate transfer learning for text generation and demonstrate that generation tasks can often help each other via prompt transfer.

Transferability of Natural Language Processing. We are also closely related to existing works on transfer learning in NLP tasks (Jeong et al., 2020; Wiese et al., 2017; Liu et al., 2019). Prior studies have shown that cross-task transfer can address the data scarcity issue (Wiese et al., 2017), enhance the ability to complex reasoning and inference (Jeong et al., 2020), or learn effective word representations (Liu et al., 2019). Efforts to transfer prompts for addressing NLU tasks have also been developed (Vu et al., 2021; Su et al., 2021). As a representative work, Vu et al. (2021) used the learned prompt to directly initialize the prompt for a target task while not considering the specific input. Our work focuses on challenging text generation tasks by utilizing prompts to extract implicit task-related

knowledge and considering specific model inputs for the most helpful knowledge transfer.

3 Preliminary

3.1 Problem Formulation

Generally, the objective of text generation is to model the conditional probability $\Pr(y|x)$, where $x = \langle w_1, \ldots, w_n \rangle$ and $y = \langle z_1, \ldots, z_m \rangle$ denote the input text and output text respectively and consist of sequences of tokens from a vocabulary $\mathcal{V}$.

Prompting is a technique for injecting extra task information to PLMs as a condition during the generation of output text (Brown et al., 2020). Typically, prompting is conducted by prepending a series of tokens (discrete prompts) or continuous vectors (continuous prompts) to the input x. In our paper, we adopt continuous prompts. Specifically, given a series of n input tokens, $x = \langle w_1, \ldots, w_n \rangle$, we first utilize PLM to embed the tokens, forming a matrix $\boldsymbol{E}_x \in \mathbb{R}^{n \times e}$, where e is the dimension of the embedding space. Then, our continuous prompt p is represented as a parameter matrix $\boldsymbol{E}_p \in \mathbb{R}^{l \times e}$, where l is the number of prompt vectors. The prompt p is then prepended to the embedded input forming a single matrix $[\boldsymbol{E}_p; \boldsymbol{E}_x] \in \mathbb{R}^{(l+n) \times e}$ which is encoded by PLMs as an ordinary sequence, such that the model maximizes the likelihood of the ground-truth y, *i.e.,* $\Pr(y|[p; x])$.

3.2 Prompt-based Transfer Learning

In a general transfer learning framework, we define a set of source generation tasks $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$, where the t-th task $\mathcal{S}_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ contains n_t tuples of the input text $x_i^t \in \mathcal{X}$ and its corresponding output text $y_i^t \in \mathcal{Y}$. For a target generation task $\mathcal{T}$, the goal of transfer learning is to use the previously learned task-specific knowledge of the source tasks $\mathcal{S}$ to help improve the performance of a learned model f_θ (parameterized by θ) in the target task $\mathcal{T}$.

In this paper, we consider a new transfer learning setting based on prompting. Specifically, the parameters of the underlying PLM are frozen, and the text generation tasks have to be fulfilled by prepending prompts (continuous vectors) to input as described in Section 3.1. Formally, we will learn an independent *source prompt* p_t for each source generation task $\mathcal{S}_t$ based on a shared frozen PLM by maximizing the likelihood $\Pr(y_i^t|[p_t; x_i^t])$. Our core idea is to transfer these learned source prompts to a new (target) text generation task, such that the target generation task can be performed in zero or few shot settings.

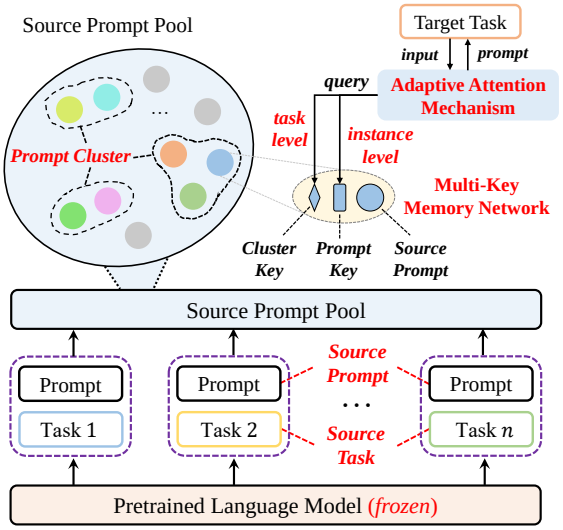

Figure 1: Overview of our proposed model PTG.

4 Approach

Our proposed method, Prompt Transfer for Text Generation (PTG), is depicted in Figure 1. Our approach first learns a number of source prompts for various representative source generation tasks, and then derive the prompt for the target generation task with a novel adaptive attention mechanism. Next we will describe each part in detail.

4.1 Learning Transferable Source Prompts

To extract task-related knowledge from source generation tasks, we learn a set of source prompts and store them in a *source prompt pool*. The motivations for introducing the prompt pool are twofold. First, we expect to identify the similarity between source generation tasks. Second, the pool stores task-specific prompts for every source task, which can be shared by all target tasks.

Constructing Source Prompt Pool. For each source generation task $\mathcal{S}_t$, we aim to learn a source prompt p_t given its training data $\{(x_i^t, y_i^t)\}_{i=1}^{n_t}$. Following the learning steps in Section 3.1, we learn an independent source prompt p_t for each source task $\mathcal{S}_t$ based on a shared frozen PLM, *i.e.,* BART. These source prompts are stored in a prompt pool $\mathcal{P} = \{p_1, \ldots, p_t, \ldots, p_T\}$, where T is the total number of source text generation tasks.

To construct the source prompt pool, a key point lies in the selection of source text generation tasks. According to the literature (Deng et al., 2021), text

generation tasks can be categorized as performing compression, transduction, or creation based on changes in conveyed information from input to output. Moreover, recent studies have shown that few but diverse source tasks/domains also lead to remarkable transfer learning performance (Friedman et al., 2021; Zhuang et al., 2021). Therefore, we select six text generation tasks (including 14 public datasets) within the three types of generation tasks for learning their corresponding source prompts.

Clustering Source Prompts. As described above, the source tasks are diverse in the prompt pool. It is challenging for PLMs to effectively transfer or reuse existing prompts for new tasks. Thus, to identify the similarity between source tasks (prompts), we construct a source prompt pool for more effective cross-task knowledge transfer. In particular, via spectral clustering algorithm (Ding et al., 2001), we group these source prompts into several prompt clusters. Under this algorithm, each prompt p_t is regarded as a node in a weighted undirected graph $\mathcal{G}$. The similarity degree (weight) between node (prompt) p_i and p_j is computed via the position-agnostic Euclidean distances (Su et al., 2021):

$$w_{i,j} = \frac{1}{1 + \frac{1}{l^2}\sum_{k_1=1}^{l}\sum_{k_2=1}^{l}||p_{i,k_1} - p_{j,k_2}||}, \tag{1}$$

where p_{i,k_1}, p_{j,k_2} denote the k_1-th and k_2-th vector of prompt p_i and p_j, respectively. We then adopt the min-max cut strategy (Ding et al., 2001) to partition the graph $\mathcal{G}$ into several subgraphs representing different prompt clusters $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$, where m is the total number of clusters. When transferring the source prompts, it will be better to identify the suitable prompt cluster and select the most relevant source prompt. By contrast, previous works considered each source prompt equally and ignore the differences between different tasks (Vu et al., 2021; Su et al., 2021).

Multi-Key Memory Network. With source prompts encoding task-related knowledge, the second motivation is to share them with every target generation task. To facilitate the prompt transfer from source tasks to target tasks, we build a multi-key memory network to store these clustered prompts. Specifically, for a source prompt p_t from the prompt cluster $\mathcal{C}_z$, *i.e.,* $p_t \in \mathcal{C}_z$, it is associated with a learnable cluster key $\boldsymbol{k}_z^c$ and a learnable prompt key $\boldsymbol{k}_t^p$, as follows:

$$\tilde{\mathcal{P}} = \{\mathcal{C}_z : \langle \boldsymbol{k}_z^c, \boldsymbol{k}_t^p, p_t \rangle\}_{z=1}^{m}, \tag{2}$$

where $\boldsymbol{k}_z^c, \boldsymbol{k}_t^p \in \mathbb{R}^d$, and d is the key embedding size. In our memory network, these learned source prompts serve as representation bases, *i.e.,* value vectors, which can be transferred to target generation tasks through key-value prompt finding.

4.2 Transferring Instance Adaptive Prompts

Previous works (Li and Liang, 2021; Vu et al., 2021) usually consider only the task information but ignore the specific input data when deriving prompts. However, for a single task, even a well-learned prompt may not be suitable for all the data instances (Scao and Rush, 2021), and thus it is non-trivial to design effective transferring strategy considering both task- and instance-level characteristics. In our model, we design an adaptive attention mechanism to incorporate the instance feature for constructing the target prompt.

Adaptive Attention Mechanism. Specifically, for an instance (x, y) of the target task $\mathcal{T}$, we use both task-level and instance-level queries to adaptively lookup and select the source prompts for transferring the previously learned task-related knowledge. The task-level query aims to select the overall information related to the specific target task, which is defined as a learnable task query vector $\boldsymbol{q}^{task} \in \mathbb{R}^d$. However, the source prompts in the pool are diverse but limited, thus the task-level prompt may not well adapt to all the data instances of the target generation task. Therefore, we design an instance-level query to learn the target prompt by attending to the highly relevant source prompts to help improve the model performance in specific instances. The instance-level query is computed as the input encoding $\boldsymbol{q}^{ins} \in \mathbb{R}^d$ through a frozen PLM such as BERT (Devlin et al., 2019):

$$\boldsymbol{q}^{ins} = \text{Average}(\text{BERT}(x)), \tag{3}$$

where we average the top-layer representations of every input tokens encoded by BERT.

For a source prompt $p_t \in \mathcal{C}_z$, we use $\boldsymbol{q}^{task}$ and $\boldsymbol{q}^{ins}$ to lookup its corresponding cluster key and source key respectively, following multi-head attention (Vaswani et al., 2017). Thus, the final matching score between the instance x and prompt p_t is calculated as:

$$s_t = \text{softmax}(\lambda \cdot \boldsymbol{q}^{task\top}\boldsymbol{k}_z^c + (1-\lambda) \cdot \boldsymbol{q}^{ins\top}\boldsymbol{k}_t^p), \tag{4}$$

where λ is a hyper-parameter. Finally, according to the weight score, the selected source prompt is computed as: $\tilde{p} = \sum_{t=1}^{T} s_t \cdot p_t$.

Compared to other prompt-based transfer learning methods that used only a fixed prompt for a new task (Vu et al., 2021; Li and Liang, 2021), our adaptive attention mechanism is able to effectively learn the most suitable prompt representation from source prompts for a specific data instance. Such a mechanism makes our model more flexible to transfer to new text generation tasks.

Prompt-based Text Generation. Based on the above adaptive attention mechanism, we retrieve the prompt $\tilde{p}$ encoding the most useful and relevant knowledge to help the model perform the specific generation instances. As described in Section 3.1, we prepend the prompt $\tilde{p}$ to the input embedding of x, which then flows through a generative PLM such as BART (Lewis et al., 2020) for generating text. The generative PLM is optimized via maximum likelihood estimation (MLE) as:

$$\mathcal{L}_{\text{MLE}}(\theta) = \mathbb{E}_{(x,y)\sim(\mathcal{X},\mathcal{Y})} \log \Pr(y|[\tilde{p}; x]). \quad (5)$$

During the learning process of the target task, the retrieved prompt $\tilde{p}$ is adaptive to different instances and is frozen because it encodes the previously *learned* task-related knowledge.

4.3 Model Discussion

For prompt-based transfer learning in text generation, the key point lies in how to effectively transfer or reuse existing prompts (encoding task-specific knowledge) for new generation tasks considering both task- and instance-level characteristics.

To achieve this goal, we first learn a set of source prompts encoding task-specific knowledge from a number of representative source text generation tasks (Section 4.1). These source prompts serve as representation bases, *i.e.,* value vectors in the multi-key memory network. Moreover, we design an adaptive attention mechanism considering both task- and instance-level information for constructing the target prompt (Section 4.2). Each data instance from a new generation task can learn a specific prompt by attending to the most highly relevant source prompts.

Compared with typical transfer learning methods, our model utilizes a lightweight technique, *i.e.,* prompting, to learn task-specific knowledge from source tasks. Our pretrained source prompts can help PLMs perform more effective and useful knowledge transfer.

5 Experiments

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

5.1 Experimental Setup

Datasets. We select 14 public datasets divided into three types of text generation tasks: i) *compression* to express salient information in concise text including summarization (CNN/Daily Mail (See et al., 2017), XSum (Narayan et al., 2018), MSNews (Liu et al., 2021a), Multi-News (Fabbri et al., 2019), NEWSROOM (Grusky et al., 2018)) and question generation (SQuAD (Rajpurkar et al., 2016)); ii) *transduction* to transform text while preserving content precisely including style transfer (Wiki Neutrality (Pant et al., 2020)) and text paraphrase (Quora (Wang et al., 2017)); and iii) *creation* to produce new content from input context including dialog (PersonaChat (Zhang et al., 2018), TopicalChat (Gopalakrishnan et al., 2019), DailyDialog (Li et al., 2017), DSTC7-AVSD (Alamri et al., 2019), MultiWOZ (Budzianowski et al., 2018)) and story generation (WritingPrompts (Fan et al., 2018)). Dataset statistics are in Appendix A.

Baselines. We compare our proposed PTG to the following baselines:

• **GPT-2** (Radford et al., 2019), **BART** (Lewis et al., 2020), and **T5** (Raffel et al., 2020): These are three representative PLMs for text generation, where all pretrained parameters are fine-tuned on each target task dataset separately. We adopt the LARGE version of these PLMs.

• **PREFIXTUNING** (Li and Liang, 2021): It is the recent state-of-the-art prompt-based PLM for text generation by concatenating a sequence of vectors and the input, which keeps PLM parameters frozen but optimizes a set of continuous prefix vectors.

• **SPOT** (Vu et al., 2021): It also adopts a prompt-based transfer learning method which first trains a prompt on source tasks and then uses the resulting prompt to initialize the prompt for a target task.

• **MULTI-TASK MODELTUNING**: This strong multi-task baseline first fine-tunes **BART** on the same source tasks used for PTG and then fine-tunes on each target task dataset individually.

We conduct all methods in the same setting to obtain their results without special tricks such as label smoothing. Compared with other baselines, our model is extremely lightweight, *i.e.,* when solving target generation tasks, we freeze the transferred

Target Task	SUMMARIZATION (CNN/Daily Mail)			DIALOG (PersonaChat)			
#Metrics	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	Distinct-1	Distinct-2
GPT-2$_{\text{LARGE}}$	30.30	7.66	26.40	36.07	22.64	**1.57**	**8.54**
BART$_{\text{LARGE}}$	41.37	21.16	38.36	40.48	26.48	1.42	7.60
T5$_{\text{LARGE}}$	40.47	20.30	37.57	42.23	27.36	1.39	7.63
PREFIXTUNING	41.79	20.69	38.50	41.87	27.28	1.33	7.20
SPOT	39.38	17.24	36.71	39.74	26.52	1.33	7.81
MT MODELTUNING	41.43	21.17	38.40	40.47	26.49	1.45	7.83
PTG	**42.40**	**21.35**	**39.14**	**45.46**	**29.52**	1.46	8.34

Table 1: Cross-task transferability performance comparisons of different methods in fully-supervised setting. **Bold** and underline fonts denote the best and the second best methods (the same as below).

Target Dataset	CNN/DAILY MAIL			PERSONACHAT			
#Metrics	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	Distinct-1	Distinct-2
GPT-2$_{\text{LARGE}}$	30.30	7.66	26.40	36.07	22.64	**1.57**	**8.54**
BART$_{\text{LARGE}}$	41.37	21.16	38.36	40.48	26.48	1.42	7.60
T5$_{\text{LARGE}}$	40.47	20.30	37.57	42.23	27.36	1.39	7.63
PREFIXTUNING	41.79	20.69	38.50	41.87	27.28	1.33	7.20
SPOT	39.85	18.21	36.33	40.39	26.34	1.32	7.60
MT MODELTUNING	41.71	21.41	38.67	42.53	27.83	1.39	7.86
PTG	**42.68**	**21.63**	**39.45**	**45.47**	**29.52**	1.43	8.34
Target Dataset	XSUM			DAILYDIALOG			
#Metrics	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	Distinct-1	Distinct-2
GPT-2$_{\text{LARGE}}$	28.28	9.17	22.29	29.14	18.01	**5.78**	21.52
BART$_{\text{LARGE}}$	43.93	20.78	35.94	32.62	21.77	5.16	25.08
T5$_{\text{LARGE}}$	41.01	17.84	32.60	31.54	20.08	5.70	29.25
PREFIXTUNING	42.87	19.98	34.82	34.00	21.63	4.31	19.95
SPOT	41.43	17.56	31.33	30.22	20.11	4.91	25.56
MT MODELTUNING	43.75	20.70	35.66	34.41	23.08	5.46	27.23
PTG	**44.21**	**20.99**	**36.00**	**42.72**	**28.75**	5.36	**29.48**

Table 2: Cross-dataset transferability performance comparisons of different methods in fully-supervised setting.

target prompt and parameters of the backbone PLM but only tune the multi-head attention parameters in adaptive attention mechanism (Eq. 4).

In particular, we adopt BART-LARGE to learn a set of source prompts. The length of prompt is set to 200 and the learning rate is set to 1×10^{-3}. For the target generation task, we utilize BART-LARGE as the generation backbone and frozen BERT-LARGE to obtain the instance-level query $\boldsymbol{q}^{ins}$. The dimension d is set to 1024, which is the same as the embedding size e of the BERT/BART-LARGE. The multi-head attention in adaptive attention mechanism has 16 heads. During fine-tuning, the learning rate of BART is set to 3×10^{-5} and the learning rate of cluster key $\boldsymbol{k}^c$, prompt key $\boldsymbol{k}^p$, task key $\boldsymbol{q}^{task}$ and multi-head attention is set to 1×10^{-3}. The value of λ is set to 0.5 based on the performance in validation set. The training details of baselines can be found in Appendix B.

Evaluation Metrics. For performance comparison, we adopt three automatic evaluation metrics widely used by previous works, *i.e.,* BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and Distinct (Li et al., 2016). Specifically, BLEU-n measures the ratios of the co-occurrences of n-grams between the generated and real text; ROUGE-n measures the text quality by counting the overlapping n-grams between the generated and real text; and Distinct-n measures the degree of diversity by calculating the number of distinct n-grams in generated text.

5.2 Fully-Supervised Setting

Table 1 and Table 2 present the fully-supervised results of cross-task and cross-dataset transferability, respectively, for our model and baselines. In fully-supervised setting, we use all training instances of the target task to train our model.

For the *cross-task* experiment, we consider two pairs of source and target tasks transfer: 1) the tar-

Target Task	SUMMARIZATION (R-1/R-2/R-L)				DIALOG (B-1/B-2/D-1/D-2)			
#Instances	50	100	200	500	50	100	200	500
GPT-2$_{\text{LARGE}}$	19.8/ 3.1/17.5	20.6/ 3.9/18.4	26.1/ 6.1/23.3	29.2/ 7.3/26.0	27.7/10.4/2.2/10.9	27.6/10.4/2.2/11.4	29.3/11.1/2.1/11.5	31.4/12.0/1.9/10.7
BART$_{\text{LARGE}}$	37.5/16.9/34.4	38.8/17.9/35.6	39.3/18.4/36.1	39.9/19.0/36.7	22.7/ 9.0/1.3/ 5.4	30.0/11.9/1.3/ 5.2	32.4/12.8/1.3/ 5.7	31.7/12.6/1.3/ 5.6
T5$_{\text{LARGE}}$	39.1/18.3/36.2	39.9/18.5/36.8	40.0/18.7/37.0	39.6/19.2/36.7	41.7/15.5/0.9/ 6.6	42.1/15.7/0.8/ 5.4	43.1/16.3/0.7/ 4.6	45.1/17.4/0.8/ 4.4
PREFIXT	32.2/12.4/28.5	32.3/12.5/28.5	34.0/13.7/30.9	37.5/16.3/34.7	39.6/23.9/0.6/ 3.4	39.7/24.0/0.5/ 3.1	36.4/22.4/0.8/ 3.7	25.7/16.1/1.1/ 4.1
SPOT	31.3/11.8/27.5	31.9/11.8/27.5	33.6/12.6/29.3	36.5/16.0/33.6	38.3/22.1/0.5/ 3.0	38.2/22.0/0.5/ 3.0	39.0/23.2/0.8/ 4.1	41.1/23.5/1.0/ 4.5
MODELT	36.2/15.6/32.8	37.8/16.6/34.4	38.6/17.3/35.2	39.3/17.9/35.8	24.9/ 9.9/1.5/ 6.6	24.8/ 9.8/1.6/ 6.6	27.8/11.0/1.6/ 7.1	28.9/11.4/1.7/ 7.8
PTG	37.8/16.7/34.5	39.0/17.5/35.6	39.3/17.7/36.2	**40.1**/19.1/**36.8**	37.3/22.6/1.1/ 6.2	39.9/21.2/1.1/ 5.3	37.7/**23.6**/1.1/ 4.9	37.7/**24.2**/1.4/ 6.3

Table 3: Cross-task transferability performance comparisons of different methods in few-shot setting. B-n, R-n, D-n, and MODELT are short for BLEU, ROUGE, Distinct and MULTI-TASK MODELTUNING (the same as below).

Target Data	CNN/DAILY MAIL (R-1/R-2/R-L)				PERSONACHAT (B-1/B-2/D-1/D-2)			
#Instances	50	100	200	500	50	100	200	500
GPT-2$_{\text{LARGE}}$	19.8/ 3.1/17.5	20.6/ 3.9/18.4	26.1/ 6.1/23.3	29.2/ 7.3/26.0	27.7/10.4/2.2/10.9	27.6/10.4/2.2/11.4	29.3/11.1/2.1/11.5	31.4/12.0/1.9/10.7
BART$_{\text{LARGE}}$	37.5/16.9/34.4	38.8/17.9/35.6	39.3/18.1/36.1	39.9/19.0/36.7	22.7/ 9.0/1.3/ 5.4	30.0/11.9/1.3/ 5.2	32.4/12.8/1.3/ 5.7	31.7/12.6/1.3/ 5.6
T5$_{\text{LARGE}}$	39.1/18.3/36.2	37.9/18.5/36.8	39.0/18.7/36.0	39.6/19.2/36.7	31.7/15.5/0.9/ 6.6	32.1/15.7/0.8/ 5.4	33.1/16.3/0.7/ 4.6	35.1/17.4/0.8/ 4.4
PREFIXT	32.2/12.4/28.5	32.3/12.5/28.5	34.0/13.7/30.9	37.5/16.3/34.7	39.6/23.9/0.6/ 3.4	39.7/24.0/0.5/ 3.1	36.4/22.4/0.8/ 3.7	25.7/16.1/1.1/ 4.1
SPOT	31.9/11.5/26.8	31.9/11.4/26.8	33.0/12.8/29.3	36.6/15.5/33.2	37.6/22.0/0.5/ 3.1	37.6/22.2/0.5/ 3.2	35.0/20.2/0.7/ 3.2	21.2/15.6/1.0/ 3.8
MODELT	37.7/17.0/34.5	38.8/17.9/35.6	39.3/18.2/36.0	40.5/19.0/36.1	32.0/13.1/2.4/12.4	34.2/13.9/2.2/11.9	35.9/14.7/2.1/11.7	35.5/14.7/2.0/10.8
PTG	37.9/16.5/34.5	38.7/17.5/35.8	**39.5**/18.3/**36.2**	39.9/18.7/36.6	34.6/21.5/1.1/ 4.5	36.9/19.3/1.0/ 5.5	**38.6/24.1**/1.0/ 4.4	**36.7/23.0**/1.2/ 5.5
Target Data	XSUM (R-1/R-2/R-L)				DAILYDIALOG (B-1/B-2/D-1/D-2)			
#Instances	50	100	200	500	50	100	200	500
GPT-2$_{\text{LARGE}}$	12.2/ 1.5/ 9.8	11.3/ 1.1/ 9.1	11.1/ 1.1/ 8.9	12.9/ 1.7/10.2	18.5/ 7.0/5.9/23.3	19.3/ 7.3/5.6/22.8	20.9/ 7.9/5.4/22.0	22.0/ 8.3/5.5/ 2.9
BART$_{\text{LARGE}}$	33.2/10.3/25.2	32.8/11.0/26.6	34.5/11.6/25.5	36.4/13.2/28.2	22.0/ 8.5/3.5/15.6	22.2/ 8.5/3.3/14.5	24.8/ 9.6/3.4/14.9	24.3/ 9.4/3.8/11.4
T5$_{\text{LARGE}}$	23.2/ 5.0/16.6	23.4/ 5.3/17.1	26.0/ 7.1/19.5	30.8/10.3/24.2	30.6/14.8/2.5/14.9	41.0/15.0/2.4/14.1	30.9/15.1/2.8/15.4	30.6/15.1/3.2/17.7
PREFIXT	25.0/ 8.3/17.9	25.0/ 8.2/17.9	25.1/ 8.2/18.1	27.5/ 9.8/19.7	38.1/22.6/2.8/14.1	38.4/22.8/2.5/12.0	35.2/21.0/2.5/11.6	21.8/13.5/3.8/16.1
SPOT	23.4/ 6.6/16.6	23.4/ 6.5/16.6	23.5/ 6.8/17.0	25.5/ 7.5/18.6	35.5/20.6/2.5/13.2	35.7/20.8/2.3/12.8	33.6/18.9/2.2/11.9	25.0/13.2/3.7/16.1
MODELT	35.6/13.1/27.8	35.7/13.3/28.0	36.0/13.6/28.4	36.1/13.8/28.5	28.2/11.1/5.4/24.3	30.4/11.8/5.2/23.9	29.8/11.8/4.9/23.0	29.5/11.7/4.7/22.3
PTG	33.6/10.9/25.4	33.8/11.2/25.9	34.7/12.0/26.8	**36.8**/13.6/27.7	31.8/19.4/2.5/11.6	30.9/18.9/2.8/12.8	31.5/19.3/2.9/13.9	**31.0/19.0**/3.1/14.9

Table 4: Cross-dataset transferability performance comparisons of different methods in few-shot setting.

get task is summarization (CNN/Daily Mail), and the source tasks are the mixture of other five tasks; and 2) the target task is dialog (PersonChat), and the source tasks are other five tasks. For the *cross-dataset* experiment, we consider datasets within summarization and dialog. For summarization, the target dataset is CNN/Daily Mail or XSum, and the source datasets are the mixture of other four summarization datasets. For dialog, the target dataset is PersonaChat or DailyDialog, and the source datasets are other four dialog datasets.

First, by transferring prompts from source tasks to the target task, PTG outperforms GPT-2, BART, T5 and PREFIXTUNING. The results suggest that prompt transfer in PTG provides an effective means of improving the performance of typical fine-tuning and prompt methods since our method utilizes the knowledge learned from source tasks.

Second, PTG performs better than the prompt-based transfer method, SPOT. While transferring prompts, SPOT considers each source task equally and ignored the specific instance information. And SPOT only learns a common prompt for source tasks to directly initialize the target prompt. By constrast, PTG clusters diverse source prompts and uses an adaptive attention mechanism considering both task- and instance-level characteristics.

Finally, PTG produces competitive performance or even exceeds the strong MULTI-TASK MODELTUNING. Different from most NLU tasks sharing some common knowledge to understand the semantics and syntax of surface words, text generation tasks need to generate diverse text based on different input data, thus having large task boundaries. Thus, in cross-task transfer, simply tuning PLMs on a mixture of tasks without considering the task similarity leads to a performance decrease. While, our prompt-based transfer learning approach can still achieve the best performance, showing that PTG improves stability across tasks and datasets.

5.3 Few-Shot Setting

In few-shot setting, we only sample a handful of training instances of the target task to train our model. Specificlly, we subsample the target task dataset to obtain small training datasets of size {50, 100, 200, 500}. For each size, we sample 5 different datasets and average over 2 training random

Target Task	SUMMARIZATION (CNN/Daily Mail)			DIALOG (PersonaChat)			
Model	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	Distinct-1	Distinct-2
PTG w/o Prompt Pool	41.46	20.40	38.40	39.70	24.45	0.77	4.00
PTG w/o Prompt Cluster	42.10	21.15	38.86	44.63	29.20	1.34	7.78
PTG w/o Multi-Key Memory	42.12	21.14	38.85	44.67	29.34	1.42	8.23
PTG w/o Instance-level Query	42.16	21.22	38.93	44.74	29.28	1.36	7.80
PTG	42.40	21.35	39.14	45.46	29.52	1.46	8.43

Table 5: Ablation analysis on cross-task transferability experiments.

seeds. Thus, we average over 10 models for each few-shot setting. In few-shot setting, we adopt the same cross-task and cross-dataset experiments with the fully-supervised setting. Table 3 and 4 shows the few-shot results of our model and baselines.

We can clearly see that PTG achieves competitive (underline fonts) or better performance (**bold** fonts) than the strong baseline (*i.e.,* MULTI-TASK MODELTUNING) in most low-data regimes, but the gap narrows as the training dataset size increases. In addition, our model outperforms most of vanilla PLMs in most cases. The reason behind this might be that large PLMs can easily suffer from overfitting during few-shot learning due to their massive parameters (Gao et al., 2020). While, in our framework, we adopt a lightweight technique, *i.e.,* prompting, to learn source prompts, which can provide the previously learned knowledge in source tasks to PLMs and serve as a better starting point when solving the target tasks.

5.4 Effectiveness of Core Designs

We further conduct ablation studies to demonstrate the effectiveness of the core designs of PTG.

Source Prompt Pool. To confirm the importance of the prompt pool, we design a counterpart of our method with only training a sharing prompt for all source tasks. From Table 5 (row 1), we can see that PTG significantly outperforms its counterpart with a single prompt, suggesting that the prompt pool encodes task-specific knowledge well.

Source Prompt Cluster. We remove the step of grouping source prompts into different clusters and directly lookup source prompts based on queries (see in Table 5 row 2). The decrease in performance demonstrates that when tasks are diverse, clustering task prompts can identify the similarity between source tasks, thus promoting effective knowledge transfer.

Multi-Key Memory Network. We remove the learnable key vector associated with prompts and directly transfer the mean of the source prompts to the target task. From Table 5 (row 4), we can see this results in a significant drop, demonstrating the importance of introducing learnable keys to dynamically select prompts through query-key matching.

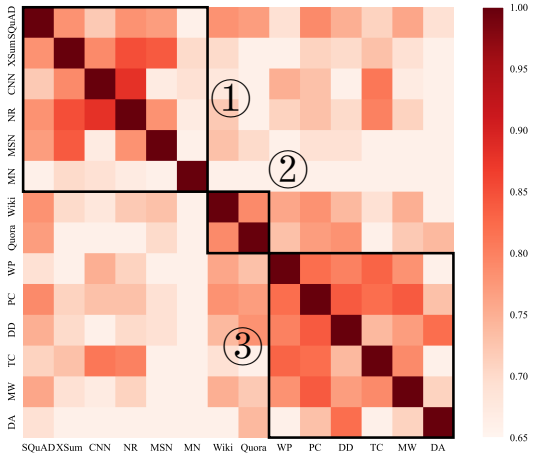

Figure 2: Similarity analysis of 14 datasets within our six generation tasks.

Instance-level Query. The instance-level query is used in adaptive attention mechanism. When we remove it (Table 5 row 3), we only use the task-level query to select source prompts. The declined performance demonstrates that incorporating the instance-level features can indeed help to transfer the most helpful knowledge to the specific instances in target tasks.

5.5 Task Similarity Analysis

Figure 2 shows a clustered heatmap of cosine similarities between the source prompts of the 14 public datasets within our six text generation tasks using the position-agnostic Euclidean distances defined by Eq. 1. We can clearly observe that our learned 14 source prompts are roughly grouped into three clusters. Similar tasks and datasets are grouped together into clusters in this heatmap, and these

clusters capture many intuitive task relationships. Specifically, these three clusters mainly focus on *compression*, *transduction*, and *creation* tasks respectively. For example, story generation (WritingPrompts) and dialog (PersonaChat) are grouped together into the third cluster. This observation further verifies our conclusion that text generation tasks can help each other within our approach by learning task-specific prompts and then transferring them to the target task. The results also suggest that our method can serve as an effective means of predicting task transferability.

6 Conclusion

This paper presented a prompt-based transfer learning approach for text generation. We learn a set of prompts from a number of representative source generation tasks and then transfer these prompts as target prompts to perform the target generation tasks. In our model, we design an adaptive attention mechanism considering both task- and instance-level information to construct the target prompts. Experiments in fully-supervised and few-shot settings demonstrate the effectiveness of our prompt-based transfer learning model. In future work, we will consider incorporating more kinds of text generation tasks.

7 Ethical Concerns

Text generation techniques has been applied to a wide range of meaningful applications for society, such as game narrative generation, news report generation, and weather report generation. However, this technique may be potentially utilized for harmful applications. Our work improves the quality of generated text compared with traditional methods. Thus, the high-quality text generated by our work makes it difficult to distinguish synthetic text from human-written text, such as fake news and stories. Here we are primarily concerned with two potential ethical issues: the possibility of deliberate misuse of our methodology and the issue of bias.

First, it is somewhat challenging to anticipate the harmful usages of our method since they often involve repurposing our model in a totally different setting or for an unexpected purpose than we planned. To alleviate this issue, we can ask for the assistance of classic security risk assessment frameworks such as detecting threats. Second, biases in training data may cause our model to generate stereotyped or prejudiced texts. This is a worry since the model bias has the potential to hurt some persons in relevant groups in unforeseen ways. To avoid prejudice, it may be useful to develop a common vocabulary that connects the normative, technological, and empirical difficulties of bias reduction for our model.

Acknowledgement

This work was partially supported by Beijing Natural Science Foundation under Grant No. 4222027, National Natural Science Foundation of China under Grant No. 61872369, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098, and the Outstanding Innovative Talents Cultivation Funded Programs 2021 of Renmin University of China. Xin Zhao is the corresponding author.

References

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Stefan Lee, Peter Anderson, Irfan Essa, Devi Parikh, Dhruv Batra, Anoop Cherian, Tim K. Marks, and Chiori Hori. 2019. Audio-visual scene-aware dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

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

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 183–190. Association for Computational Linguistics.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7580–7605. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. 2001. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2 December 2001, San Jose, California, USA*, pages 107–114. IEEE Computer Society.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1074–1084. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Dan Friedman, Ben Dodge, and Danqi Chen. 2021. Single-dataset experts for multi-dataset question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6128–6137. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 708–719. Association for Computational Linguistics.

Minbyul Jeong, Mujeen Sung, Gangwoo Kim, Donghyeon Kim, Wonjin Yoon, Jaehyo Yoo, and Jaewoo Kang. 2020. Transferability of natural language inference to biomedical question answering. In *CLEF*.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. 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 *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The 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. 2021a. 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.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2022. A survey of pretrained language models based text generation. *CoRR*, abs/2201.05273.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021b. Few-shot knowledge graph-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1558–1568. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021c. Pretrained language model for text generation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4492–4499. ijcai.org.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 1969–1979. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 986–995. Asian Federation of Natural Language Processing.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021a. GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 408–420. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *NAACL-HLT*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Kartikey Pant, Tanvi Dadu, and Radhika Mamidi. 2020. Towards detection of subjective bias using contextualized word embeddings. In *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 75–76. ACM / IW3C2.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2627–2636. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. 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 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

YuSheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2021. On transferability of prompt tuning for natural language understanding. *CoRR*, abs/2111.06719.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information*

Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *CoRR*, abs/2110.07904.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150. ijcai.org.

Georg Wiese, Dirk Weissenborn, and Mariana L. Neves. 2017. Neural domain adaptation for biomedical question answering. In *CoNLL*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76.

Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. Controllable generation from pre-trained language models via inverse prompting. *arXiv preprint arXiv:2103.10685*.

Appendix

We provide some experiment-related information as supplementary materials. The appendix is organized into three sections:

- Statistics of each dataset are presented in Appendix A;
- Training settings of baselines and our model PTG are presented in Appendix B.

A Statistics of Datasets

The detailed information of the dataset for each task is listed in Table 6, including summarization (CNN/Daily Mail, XSum, MSNews, Multi-News and NEWSROOM), question generation (SQuAD), style transfer (Wiki Neutrality), text paraphrase (Quora), dialog (PersonaChat, TopicalChat, DailyDialog, DSTC7-AVSD and MultiWOZ) and story generation (WritingPrompts). These datasets are utilized under MIT license.

B Configuration of Models

The learning rate of other baselines is set to 3×10^{-5}, which is the same as our backbone BART. The other settings of baselines and our model are set the same for fair comparison. And we do not utilize special tricks such as label smoothing, warm-up learning rate and length penalty. We apply the Adam optimizer and set $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1 \times 10^{-6}$. We set the accumulated batch size of each model to 96 using accumulated gradients. Furthermore, we use 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 a no repeat ngram size of 3. We train our models using NVIDIA A100 GPUs and PyTorch 1.9.0 upon Ubuntu 20.04.2 LTS.

Dataset	#Train	#Valid	#Test	#Input	#Output
CNN/Daily Mail	287113	13368	11490	790.2	58.4
Xsum	204017	11327	11333	358.8	21.1
MSNews	136082	7496	7562	311.6	24.8
Multi-News	44972	5622	5622	2291.9	263.1
NEWSROOM	995040	108837	108862	658.5	26.7
SQuAD	75722	10570	11877	148.3	11.6
Wiki Neutrality	145197	18149	18150	29.1	27.3
Quora	119410	14927	14926	9.8	9.9
PersonaChat	122499	14602	14056	122.1	11.9
TopicalChat	179750	11142	11221	216.6	20.3
DailyDialog	76052	7069	6740	68.4	13.9
DSTC7-AVSD	145521	33953	11780	90.7	9.5
MultiWOZ	105115	13748	13744	110.7	13.2
WritingPrompts	67765	3952	3784	25.7	232.3

Table 6: 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.