

# Exploring Prompt-based Few-shot Learning for Grounded Dialog Generation

Chujie Zheng, Minlie Huang

The CoAI Group, DCST, Institute for Artificial Intelligence,

State Key Lab of Intelligent Technology and Systems,

Beijing National Research Center for Information Science and Technology,

Tsinghua University, Beijing 100084, China

chujiezhengchn@gmail.com, aihuang@tsinghua.edu.cn

## Abstract

Dialog grounding enables conversational models to make full use of external information to establish multiple desired qualities, such as knowledgeable, engaging and empathetic. However, naturally grounded dialog corpora are usually not directly available, which puts forward requirements for the few-shot learning ability of conversational models. Motivated by recent advances in pre-trained language models and prompt-based learning, in this paper we explore prompt-based few-shot learning for grounded dialog generation (GDG). We first formulate the prompt construction for GDG tasks, based on which we then conduct comprehensive empirical analysis on two common types of prompting methods: template-based prompting and soft-prompting. We demonstrate the potential of prompt-based methods in few-shot learning for GDG and provide directions of improvement for future work.

## 1 Introduction

Grounded dialog generation (GDG) aims at grounding the contents of model-generated dialogs on the given external information (Ghazvininejad et al., 2018; Huang et al., 2020). Previous works have enhanced conversational models to be knowledgeable, engaging and empathetic by leveraging various grounding sources, such as Wikipedia documents (Dinan et al., 2018), persona descriptions (Zhang et al., 2018) or emotional support strategies (Liu et al., 2021b).

However, one key bottleneck of GDG is the reliance on large-scale corpora where dialogs are seamlessly grounded on the external information. Since these data is usually not directly available, existing GDG datasets are mostly collected via crowdsourcing (Zhang et al., 2018; Moghe et al., 2018; Dinan et al., 2018; Gopalakrishnan et al., 2019; Liu et al., 2021b), which is costly and time-consuming. Under the above background, a natural question is raised: *can we build grounded dialog models*

*with only a small amount of data (that is, under the few-shot learning setting)?*

Motivated by recent advances in pre-trained language models (PLMs) (Zhao et al., 2020a; Zhang et al., 2020), we believe that PLMs, like the GPT models (Radford et al., 2018, 2019), play a critical role in few-shot learning for GDG. Very recently, a new paradigm named “prompt-based learning” (Brown et al., 2020; Liu et al., 2021a) focuses on equipping PLMs with constructed prompts to solve downstream tasks. This paradigm attracts increasing research interest due to the amazing results in few-shot or even zero-shot learning for various tasks such as classification (Schick and Schütze, 2021; Gao et al., 2021; Han et al., 2021) and text generation (Schick and Schütze, 2020; Li and Liang, 2021).

In this paper, we make a first step towards exploring prompt-based few-shot learning for grounded dialog generation. Specifically, we take the small- and medium-sized GPT-2 (Radford et al., 2019) as the backbone models and conduct experiments on three typical GDG tasks: Wizard-of-Wikipedia (Dinan et al., 2018), PersonaChat (Zhang et al., 2018), and ESConv (Liu et al., 2021b), where the dialogs are grounded on knowledge documents, persona descriptions, and emotional support strategies respectively. We formulate the prompt construction for GDG tasks (Section 2.3) and empirically compare two common types of prompting methods: *template-based prompting* (Section 4) and *soft-prompting* (Section 5), which are either crafted manually or adapted from previous works. We comprehensively analyze the influence of various factors on the few-shot learning performance, including task characteristics, model sizes, and inherent properties of different prompting methods. Our analysis highlights the potential of prompt-based methods in few-shot learning for GDG and also provides directions of improvement for future work.

Wikipedia (Lifeguard)	A lifeguard is a rescuer who <i>supervises the safety and rescue of swimmers, surfers, and other water sports participants</i> such as in a swimming pool, water park, beach, spa, river and lake... <i>In some areas, lifeguards are part of the emergency services system to incidents</i> and in some communities, lifeguards may function as the primary EMS provider.		Persona	I am an artist I have <i>four</i> children I recently got a cat I enjoy walking for exercise I love <i>watching Game of Thrones</i>	Seeker	I feel so frustrated. <i>(Question)</i>
	Speaker 1	Hi		Seeker	May I ask why you are feeling frustrated?	
Apprentice	Speaker 2	Hello ! How are you today ?	Supporter	My school was closed without any prior warning due to the pandemic. <i>(Self-disclosure)</i>		
	Speaker 1	I am good thank you , how are you.		Seeker	I understand you. I would also have been really frustrated if that happened to me.	
Wizard	Speaker 2	Great, thanks ! My children and I were just about to <i>watch Game of Thrones</i> .	Seeker	Yeah! I don't even know what is going to happen with our final. <i>(Reflection of Feelings)</i>		
	Speaker 1	Nice ! How old are your children?		Supporter	That is really upsetting and stressful. <i>(Providing Suggestions)</i>	
Apprentice	Speaker 2	I have <i>four</i> that range in age from 10 to 21. You?	Supporter	Have you thought about talking to your parents or a close friend about this?		
	Speaker 1	I've heard that <i>in some places, lifeguards also help with other sorts of emergencies</i> , like mountain rescues!		Seeker		

Figure 1: Examples of grounded dialogs from Wizard-of-Wikipedia (Dinan et al., 2018), PersonaChat (Zhang et al., 2018) and ESConv (Liu et al., 2021b) respectively. **Left:** the dialog between an apprentice and a wizard is grounded on the Wikipedia knowledge. **Middle:** the dialog between two speakers is grounded on the pre-defined persona profile. **Right:** the dialog between a help-seeker and a supporter is grounded on various emotional support strategies. The interlocutors whose roles are played by models in these tasks are marked in **bold**. The parts of grounding sources that the utterances are grounded on are marked in *red*.

Our contributions are summarized as follows:

- On three typical grounded dialog generation tasks, we conduct extensive experiments to evaluate the prompt-based few-shot learning performance. Specifically, we compare and analyze both template-based prompting and soft-prompting methods.
- We make a first step towards exploring prompt-based few-shot learning for grounded dialog generation. Our empirical analysis also provides potential directions for future work.

## 2 Related Work and Preliminaries

### 2.1 Grounded Dialog Generation (GDG)

In recent years, researchers are increasingly interested in grounding generated dialogs on external information (Ghazvininejad et al., 2018; Zhou et al., 2018b; Dinan et al., 2018; Zhou et al., 2018a; Wolf et al., 2019; Gopalakrishnan et al., 2019; Moon et al., 2019; Zhou et al., 2020; Zheng et al., 2020a; Liu et al., 2021b). (Zhou et al., 2018b) and (Dinan et al., 2018) both utilize Wikipedia documents as the background knowledge, where the former focuses on the movie domain while the latter covers more topics. (Zhang et al., 2018) show that endowing conversational agents with persona profiles can empower them to be more engaging during multi-turn interaction. In a broader sense, the grounding sources are not only in the form of unstructured texts, but can also be structured information or discrete concepts. For instance, (Zhou et al.,

2018a; Moon et al., 2019; Zhou et al., 2020) use knowledge graphs as the grounding sources, which can better leverages information about named entities with their relations. A few months ago, in a new task named emotional support conversation (Liu et al., 2021b), the dialogs are grounded on various emotional support strategies, such as self-disclosure (sharing similar experiences or emotions) and affirmation / reassurance (affirming the help-seeker’s strengths, motivation or capabilities), which enable dialog systems to be more empathetic and to provide more effective emotional support.

### 2.2 Pre-training and Prompt-based Learning

In the past few years, large-scale pre-trained language models (PLMs) have shown the dramatic utility in various NLP tasks (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020). In general, a PLM is first trained on massive textual data with unsupervised learning objectives, and then fine-tuned on downstream tasks with additional parameters and task-specific training goals. Such pre-training techniques have been similarly utilized to address low-resource knowledge-grounded dialog generation (Zhao et al., 2020a; Li et al., 2020).

Very recently, there emerges a new paradigm named “prompt-based learning” that presents impressive few-shot learning performance (Brown et al., 2020). In this paradigm, the downstream tasks are reformulated to be close to the pre-training objectives of PLMs with the help of textual or continuous prompts. For instance, Schick and

Schütze (2020) manually design natural language templates to enhance few-shot learning in text summarization and headline generation tasks. We refer readers to (Liu et al., 2021a) for a comprehensive survey.

Formally, when a PLM is directly fine-tuned on the downstream task, it learns the probability of the target output  $Y$  given the task input  $X$ :  $P(Y|X)$ . Instead, a prompting function  $f(\cdot)$  first converts the original input  $X$  into a specific form:  $\tilde{X} = f(X)$ , which may be combined with either natural language tokens or continuous vectors. The PLM is then trained or directly applied to predict  $Y$  given  $\tilde{X}$ :  $P(Y|\tilde{X})$ .

### 2.3 Prompt Construction for GDG Tasks

As pointed out in (Liu et al., 2021a), prompt-based learning makes it necessary to design (or find) suitable prompting functions to solve the downstream tasks. Prior to this step, we note that compared with the tasks where prompt-based methods are well applied (Schick and Schütze, 2020; Gao et al., 2021), GDG tasks have several unique characteristics, which may require special attention. First, the model should distinguish utterances from different interlocutors due to the dialog consistency of a same interlocutor (Zhang et al., 2018) and the potentially asymmetric roles between them (Dinan et al., 2018; Liu et al., 2021b). Second, the model should also treat various input constituents accordingly due to their remarkably different functions (such as the dialog history and the grounding source). To this end, we next formulate the prompt construction for GDG tasks, based on which we further explore prompt design in Section 4 and 5.

**Dialog History (DH) and System Response (SR)**  
Like conventional dialog generation tasks, one of the core goals of GDG is to generate a fluent SR that is relevant and coherent to the current DH.

**Grounding Source (GS)** In a GDG task, the model is additionally given with GS, which can be any forms of information such as texts, graphs / tables or discrete concepts. Hence, the other core goal of GDG is to meanwhile ground SR on GS:  $P(Y|X)$ , where  $X = (\text{DH}, \text{GS})$  and  $Y = \text{SR}$ . That is, SR should not only be conditioned on DH, but also contain consistent information with GS.

**Structure Indicator (SI)** As aforementioned, there are various constituents in the model input  $X = (\text{DH}, \text{GS})$ , such as different information sources (DH and GS) and the utterances from

different interlocutors in DH. To distinguish and treat accordingly these components, we usually require SI to indicate the structure of the input sequence. In previous GDG works, (Zhao et al., 2020b; Sun et al., 2021) prepended with special indicator tokens to distinguish different input constituents, while (Wolf et al., 2019; Zheng et al., 2020b, 2021) used the additional embedding layer to highlight their different attributes. In the GPT-3 chat demo<sup>1</sup>, the prefix “Human:” or “AI:” in each conversation line also serves as SI, which differentiates the user’s posts and the system’s responses. Note that SI is usually fused with DH and GS, thus we do not model it separately and explicitly.

**Task Instruction (TI)** TI plays a key role in prompt-based learning, which bridges the gap between the pre-training objectives of PLM and the downstream tasks. For instance, (Schick and Schütze, 2021) takes the format of cloze questions to perform text classification, and (Schick and Schütze, 2020) uses the textual prompt “TL;DR:” to instruct text summarization. Specifically, a devised TI defines a corresponding prompting function  $f_{\text{TI}} : X \rightarrow f_{\text{TI}}(X)$ . Combined with all the above construction components, the model is trained with the following objective:  $P(Y|\tilde{X}) = P(\text{SR}|f_{\text{TI}}(\text{DH}, \text{GS}))$ .

## 3 Experimental Setup

### 3.1 Data Preparation

Our experiments are conducted to explore the few-shot learning performance of prompt-based methods on GDG tasks. We selected three typical GDG tasks, whose examples are shown in Figure 1.

**Wizard-of-Wikipedia (WoW)** (Dinan et al., 2018)  
WoW is a knowledge-grounded dialog task, where the model plays the role of a knowledgeable wizard, who grounds the utterances on Wikipedia documents to converse and provide information. In WoW, each conversation is tied with a given topic along with its Wikipedia abstract, which we use as GS of this conversation<sup>2</sup>. The average length of grounding sources is 265.5 (with GPT-2 tokenization).

**PersonaChat (PC)** (Zhang et al., 2018)  
PC is

<sup>1</sup><https://beta.openai.com/examples/default-chat>

<sup>2</sup>In the original WoW dataset, the wizard’s utterances may sometimes shift to other relevant topics, which account for about 30% of the utterances that explicitly select knowledge pieces. For GPU memory limitations and for simplicity, we only use the knowledge of the initial topic as GS.

a persona-grounded dialog task, where the model is assigned with a pre-defined profile consisting of several textual persona descriptions. For each conversation, we concatenate all the persona descriptions as GS, whose average length is 34.4. Note that different from WoW where utterances are encouraged to be constructed using knowledge, the persona profiles in PC merely serve as additional background information of the conversations. Hence, in PC there is a large proportion of utterances that do not refer to any persona description, which is also reflected in Figure 1.

**ESConv** (Liu et al., 2021b) ESConv is a support strategy-grounded dialog task, where the model uses various strategies to provide emotional support to the help-seekers. Different from WoW and PC where GS is in the form of unstructured texts, ESConv takes discrete concepts (support strategies) as GS, which have more abstract and complex meanings.

Before sampling few-shot training sets for each task, we noted that the three datasets vary obviously in the dialog lengths or the numbers of (DH, SR) pairs in dialogs. Table 1 shows related statistics. One dialog in ESConv contains over three times as many (DH, SR) pairs as WoW (13.1 vs. 4.0) while PC nearly two times (7.4 vs. 4.0). In order to balance the sizes of training data on different tasks, we set a minimum unit of sampled dialogs for each task. Specifically, one unit of WoW contains 30 dialogs while PC and ESConv are 15 and 10 respectively. As a result, our subsequently sampled training sets on three tasks contain 121.5, 110.3, 134.5 pairs of (DH, SR) per unit on average, as shown in Table 1. Under the few-shot learning setting, for each task we sampled 1, 2, 4, 8 and 16 units as the training sets. We meanwhile sampled another 16 units as the test sets, saying 480, 240, 160 dialogs for WoW<sup>3</sup>, PC, ESConv respectively.

### 3.2 Model Selection

We used GPT-2 (Radford et al., 2019) as our backbone model for two major reasons. (a) The dialog generation task is more close to GPT-2’s pre-training objective of language modeling than other objectives such as text-infilling (T5, Raffel et al. 2020) or denoising (BART, Lewis et al. 2020). Our preliminary experiments also showed better performance of GPT-2 than other PLMs. (b) Existing

<sup>3</sup>The original WoW dataset contains one in-domain test set and the other out-of-domain one, from which we sampled 8 units separately (240 dialogs each, 480 dialogs totally).

Datasets	Avg. # Pairs / Dialog	# Dialogs / Unit	Avg. # Pairs / Unit
WoW	4.0	30	121.5
PC	7.4	15	110.3
ESConv	13.1	10	134.5

Table 1: Statistics of the three datasets and the sampled training sets. One “pair” denotes one tuple of (DH, SR). The “unit” of each dataset denotes the corresponding minimum unit of sampled dialogs.

pre-trained models usually include Wikipedia in their pre-training corpora (Dong et al., 2019; Raffel et al., 2020). It may lead to data leakage for WoW and cannot truly reflect the effect of prompting, which however is exactly the purpose of our experiments. In contrast, the pre-training corpora of GPT-2 removed all Wikipedia documents, making it more suitable for our experiments. Finally, we adopted GPT-2 models of two sizes: 117M and 345M (768 / 1024 hidden sizes, 12 / 16 attention heads, 12 / 24 layers, respectively).

### 3.3 Implementation Details

For each designed prompt and for each size of GPT-2, we used the AdamW (Loshchilov and Hutter, 2018) optimizer with batch size 4 and learning rate 1e-4. Each model was trained for 2 epochs<sup>4</sup> with the linear learning rate scheduler and the warmup steps 20. During inference, we adopted beam search decoding with beam size 4. Due to the instability of few-shot learning, for each training data size, we randomly sampled 5 different sets, and for each set we used 2 random seeds for model training. Hence, each of the later reported results was averaged using 10 model outputs.

### 3.4 Evaluation Metrics

We used two kinds of automatic metrics to evaluate the model performance.

**Reference-F1** We computed the unigram F1 between SR and the reference response. Such word-overlap based metric can reflect how well the model adapts to the downstream task.

**Groundedness** Recall that in a GDG task, SR is generated conditioned on both DH and GS. It is natural and necessary to evaluate the extent to which SR is grounded on GS. For WoW and PC, we used **GS-F1** as the groundedness score, which

<sup>4</sup>We noted that the two epochs of training could already make the model converge well, while increasing the training epochs did not lead to obviously different performance.

<p>Below is the conversation about the topic <code>{topic}</code> between a user and a <b>knowledgeable</b> system. The system's utterances are grounded on the background knowledge: <code>{knowledge}</code></p> <p>User: <b>POST1</b> System: <b>RESPONSE1</b> User: <b>POST2</b> System: <b>RESPONSE2</b></p>	<p>...(omitted)... The system's utterances are grounded on the background knowledge: <code>{knowledge}</code></p> <p>User: <b>POST1</b> System: <b>RESPONSE1</b> User: <b>POST2</b></p> <p>Grounded on the background knowledge, what does the system probably say in the next response? System: <b>RESPONSE2</b></p>	<p>...(omitted)... User: <b>POST1</b> System: <b>RESPONSE1</b> User: <b>POST2</b></p> <p>The system's utterances are grounded on the background knowledge: <code>{knowledge}</code> Grounded on the background knowledge, what does the system probably say in the next response? System: <b>RESPONSE2</b></p>
---	---	--

Figure 2: Templates manually designed for WoW (those for PC are similar and are shown in Appendix, Figure 9). **Left:** directly continue the conversation. **Middle:** use a query to predict the system’s next response in the form of QA. **Right:** move GS right after DH (and before the query). **Blue:** task instruction (**TI**, containing a scenario description or / and a query). **Orange:** grounding source (**GS**). **Green:** structure indicator (**SI**). **Black:** dialog history (**DH**). **Red:** system response (**SR**). All the non-red texts are fed into the models and the red texts are the target outputs.

<p>Below is the conversation between a user and an <b>empathetic</b> system. The system uses various support skills to provide emotional support to the user.</p> <p>User: <b>POST1</b> System reflects and describes user's feelings: <b>RESPONSE1</b> User: <b>POST2</b> System shares similar experiences or emotions: <b>RESPONSE2</b></p>	<p>...(omitted)... User: <b>POST1</b> System: <b>RESPONSE1</b> User: <b>POST2</b></p> <p>To share similar experiences or emotions, what does the system probably say in the next response? System: <b>RESPONSE2</b></p>
--	---

Figure 3: Templates manually designed for ESConv. **Left:** directly continue the conversation (to maintain the consistency of the input format, the explanations of previously adopted strategies are also added in the corresponding turns). **Right:** use a query to predict the system’s next response in the form of QA.

is computed as the maximum unigram F1 between SR and the sentences in GS (after removing the stop words). For ESConv, we first trained a BERT (Devlin et al., 2019) classifier on ESConv to identify the support strategy displayed in SR. Then we computed **match ratio**, the ratio of cases where the identified strategies exactly matched the designated ones<sup>5</sup>.

## 4 Template-based Prompting

We first focus on the *template-based prompting* methods, where the prompting function  $f_{\text{TI}}(\cdot)$  directly inserts natural language tokens in the original input sequence. In Section 4.1, we elaborate on several manually crafted templates for the three GDG tasks, and then we empirically compare and analyze the performance of these methods in Section 4.2 and 4.3.

### 4.1 Prompt Design

The core of template-based prompting is to devise proper textual templates, by which DH and GS can

be combined naturally. Inspired by previous works, we manually designed three generic templates for the selected GDG tasks, as shown in Figure 2 and 3. It is worth noting that we do not claim that our intuitively designed templates are optimal. We also found it intractable to directly apply previously proposed methods of template-searching (Shin et al., 2020; Gao et al., 2021) on GDG tasks. Future work can explore effective ways of automatic template generation for GDG tasks.

**Continuation** Inspired by GPT-3’s demo examples (Brown et al., 2020), we first design the template where the model directly continues the current conversation. Similarly, we also use a textual *scenario description* to guide the model to complete tasks, as the **blue** texts shown in the left parts of Figure 2 and 3. Specifically, a *scenario description* describes the scenario setting corresponding to the later input texts, and pre-defines the attributes of the elements inside. For instance, in WoW (Figure 2), the scenario description defines a *conversation* scenario between a *user* and a *system*, that the system is *knowledgeable*, and the fact that *the system grounds its utterances on the background knowledge*. Then, we use “User:” and “System:” as SI

<sup>5</sup>ESConv defines 7 support strategies along with an “others” one, which does not explicitly refer to any specific strategy. Hence, we removed the cases where the designated strategies are “others” when computing match ratio.

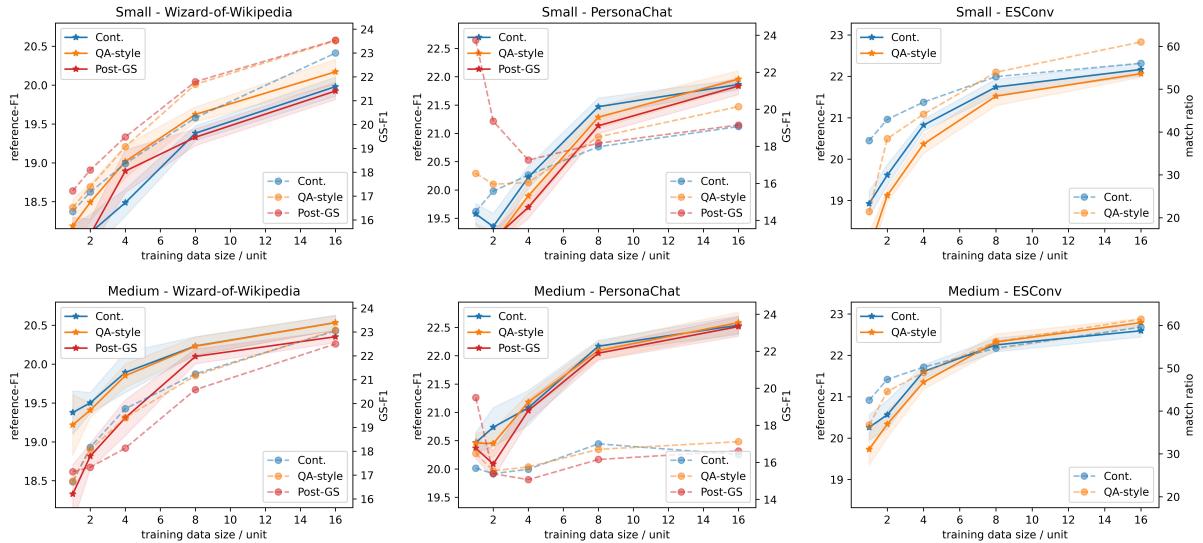


Figure 4: Evaluation results of template-based prompting methods. Solid lines marked with stars denote reference-F1, while dashed lines marked with dots denote groundedness scores (GS-F1 or match ratio). The meanings of these metrics are explained in Section 3.4.

to distinguish the two interlocutors<sup>6</sup>. Note that GS in ESConv is discrete and abstract concepts (emotional support strategies), which is different from WoW and PC where GS is unstructured texts and can be directly appended after the scenario description. To address this discrepancy, we instead infill textual explanations of adopted strategies in the system SIs, which interpret the strategies that the system’s utterances are grounded on. Finally, given the scenario description, the infilled GS, the DH decorated with SI, and the system SI placed at the end, the model would continue the conversation by completing the next system response.

**QA-style** Existing template-based prompting methods usually adopt the form of QA (such as cloze test) (Petroni et al., 2019; Schick and Schütze, 2021). Inspired by this, we design the second template as a QA-style one. Instead of directly continuing the given question, the model is first queried *what does the system probably say next* given the scenario description, GS and DH. Then the model answers the query with its predicted system response, as shown in the middle part of Figure 2 and the right part of Figure 3.

**Post-GS** On top of the QA-style template, we additionally attempt to move GS after DH and before the query in WoW and PC, as shown in the right part of Figure 2. This is due to that the order

of multiple constituents may potentially impact the effect of dialog grounding, as revealed in previous works (Wolf et al., 2019; Golovanov et al., 2019). Intuitively, making GS and SR closer may improve the groundedness.

## 4.2 Results

Evaluation results are shown in Figure 4. We analyze the results from following perspectives.

**Task Characteristics** In most cases, Continuation performs best in terms of reference-F1, indicating that this template makes dialog generation more close to the pre-training objective of language modeling. One exception is on the WoW task. With the small-sized GPT-2, QA-style obviously outperforms other templates, while it is slightly inferior to Continuation with the medium-sized model. It may result from that the text styles of DH and GS in WoW are more formal, making it also suitable for the QA-style template.

**Model Sizes** The larger-sized model has a stronger ability of task adaption with more training data, which leads to higher reference-F1 scores and making the gaps between different templates smaller. For instance, the performances of Continuation and QA-style across three tasks get closer when the training data is more than 4 units.

**Groundedness** Notably, Post-GS improves GS-F1 on WoW with the small-sized GPT-2 and on PC with both sizes. It suggests that the postposition of GS indeed enables SR to focus more on GS. We

<sup>6</sup>In our preliminary experiments, we found that the keywords “User” / “System” led to better performance than other candidates, such as “Human” / “AI” adopted in GPT-3 demos.

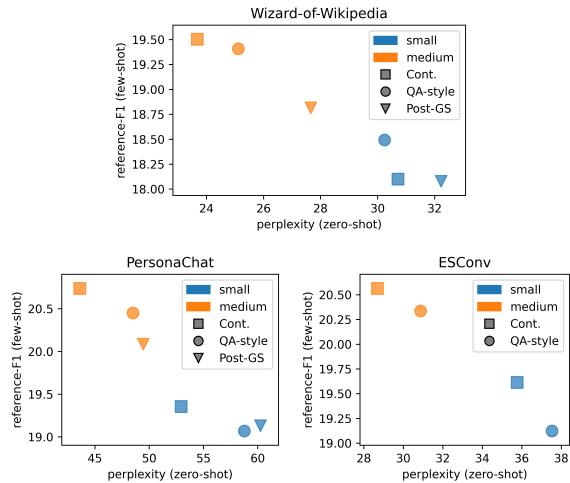


Figure 5: Few-shot learning performance with 2 units training data vs. zero-shot perplexity of GPT-2 models.

also note that on PC, the GS-F1 scores of Post-GS reach the peak with the fewest training data, which, however, drop rapidly when the training data size is enlarged. For the former phenomenon, it may be due to the text styles of DH and GS in PC are similar, which enables models to easily imitate the contents of GS. For the latter phenomenon, we think the reason is that many utterances in PC do not refer to the given persona, which leads to that the gains of groundedness from Post-GS gradually disappear during the models fitting the training data.

#### 4.3 Relation with Zero-shot Performance

While human-crafted templates can address the few-shot learning for GDG tasks to some extent, it is difficult and time-consuming to manually discover the optimal templates (Liu et al., 2021a). If the potential of templates can be pre-judged before extensive experiments, it will provide useful reference for automatic template searching. Due to that prompt-based learning aims at making the forms of downstream tasks closer to the pre-training objectives, we then naturally hypothesize: if a textual template can be better modeled by GPT-2, it may lead to better few-shot learning performance.

To verify our hypothesis, we computed the perplexity of GPT-2 models on the test sets of the three tasks. Without loss of generality, we took the reference-F1 scores under 2 units’ training data to reflect the few-shot learning performance. Results are shown in Figure 5. It can be seen that the reference-F1 scores are positively correlated with the performance of language modeling, which

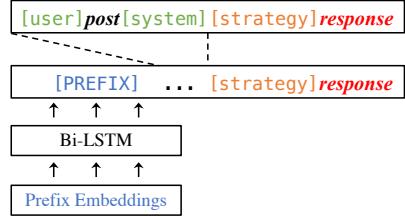


Figure 6: Schematic of the soft-prompting methods on ESConv. **Upper (top row):** insert special tokens (new parameters) into the original input sequence. **Lower:** additionally prepend with a continuous prefix (Li and Liang, 2021), which is obtained through using a Bi-LSTM to encode the initial prefix embeddings (Liu et al., 2021c).

holds both between different templates and between models of different sizes. It suggests that templates for few-shot learning can be searched or optimized based on the zero-shot performance of language modeling, which is referable for future exploration.

### 5 Soft-prompting

While template prompts are interpretable to human, the fundamental purpose of prompting is still to enhance PLMs to effectively solve downstream tasks. Instead of manually devising “hard” templates as prompts, previous works have found it feasible to perform prompting directly in the embedding space of models, namely *soft-prompting* (Li and Liang, 2021; Lester et al., 2021). In this case, the prompting function  $f_{TI}(\cdot)$  uses learnable vectors as continuous TI, which is inserted in the initial embeddings (Lester et al., 2021) or the hidden representations (Li and Liang, 2021) of the input  $X = (\text{DH}, \text{GS})$ . Next, we first provides several designs of soft prompts with reference to previous works (Section 5.1) and then experimentally analyze their performance (Section 5.2).

#### 5.1 Prompt Design

In addition to the trend of prompt-based learning, previous researchers have tried to introduce additional parameters to enhance the learning of GDG tasks. For instance, (Zhao et al., 2020b; Liu et al., 2021b) prepend input sequences with *special tokens* as SI or GS, which is essentially a simple yet basic soft-prompting method and is also adopted in recent works of prompt-based learning (Liu et al., 2021c; Lester et al., 2021). Based on this way, we elaborate on several variant soft-prompting methods, as shown in Figure 6.

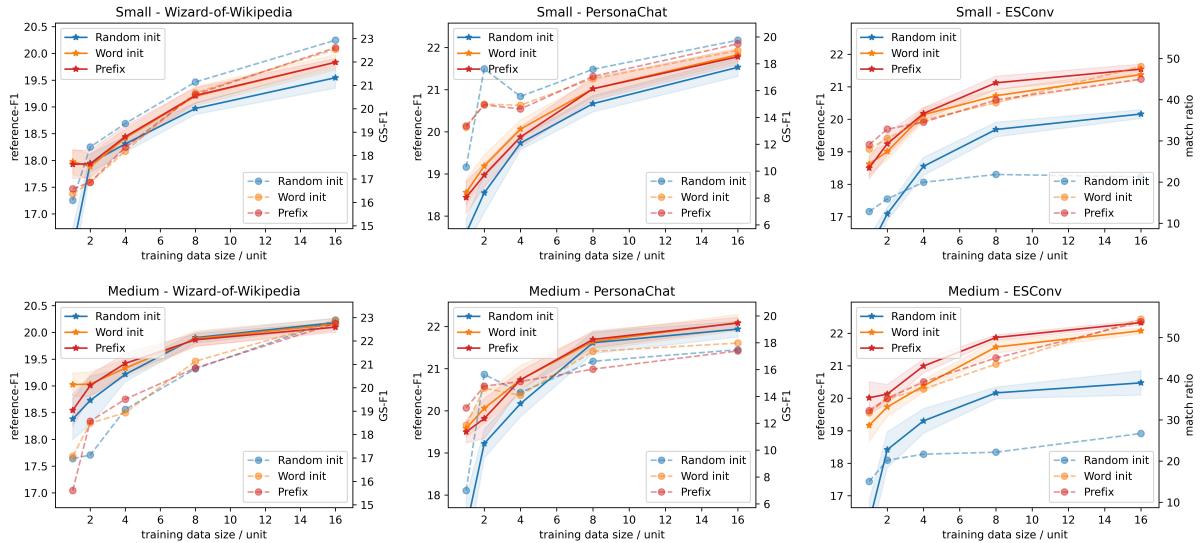


Figure 7: Evaluation results of soft-prompting methods.

**Special Token** In a typical setting, the special tokens are new parameters added in the embedding layer, which are used to distinguish different constituents and speakers. The example of ESConv is shown in the upper part (the top row) in Figure 6, where GS (support strategies) is also represented with corresponding special tokens, as done in (Liu et al., 2021b). We list the used special tokens and the input formats in Appendix, Table 2.

However, what is especially tricky in soft-prompting methods is the initialization of new parameters, which can sensitively affect the model performance in particular under the low-data setting (Li and Liang, 2021). We thus compare two different initialization methods: **Random** or based on pre-trained **Word** embeddings. For the latter one, the texts inside the special tokens (e.g., “knowledge” in [knowledge]) are first tokenized with the GPT-2 tokenizer. Then, the averaged embeddings of the obtained tokens (directly from the embedding layers of the pre-trained GPT-2 models) are used to initialize the embeddings of the corresponding special tokens.

**Prefix** Inspired by Prefix-Tuning (Li and Liang, 2021), on top of the Word Initialization method, we additionally prepend with a continuous prefix, which functionally serves as a continuous TI. In order to alleviate the training instability caused by these additional prefix parameters, we refer to the idea of P-Tuning (Liu et al., 2021c). Specifically, the initial embeddings<sup>7</sup> of the continuous prefix are

encoded with a Bi-LSTM in prior to being concatenated with the embeddings of the input sequence, as shown in Figure 6.

## 5.2 Results

Evaluation results are shown in Figure 7, where the prefix length defaults to 20. We analyze the results from following perspectives.

**Task Characteristics** On all the three tasks, initializing the embeddings of special tokens with additional semantic information always brings obvious improvement. It is worth noting that while several previous works of GDG also utilized special tokens to improve model performance (Zhao et al., 2020b; Liu et al., 2021b), they did not pay enough attention to the initialization of these additional parameters, which would naturally hinder the few-shot learning performance.

Specifically, on ESConv, prepending with continuous prefix leads to remarkable performance improvement, while no obvious difference on WoW and even slightly worse on PC. (a) As to ESConv, the reason may be that the task itself (providing emotional support) and its GS (support strategies) are more abstract. In this case, task-specific information incorporated in initialization (using the average word embeddings) and implicit task instruction in continuous prefix can both bring more specific prompts and thus lead to better prompting effects. (b) In contrast, on PC, due to the simplest form of GS and its similar text style to DH, appending prefix fails to provide additional useful information. (c) On WoW, we conjecture that it

<sup>7</sup>The initial embeddings of prefix are randomly initialized, which is consistent with the original P-Tuning.

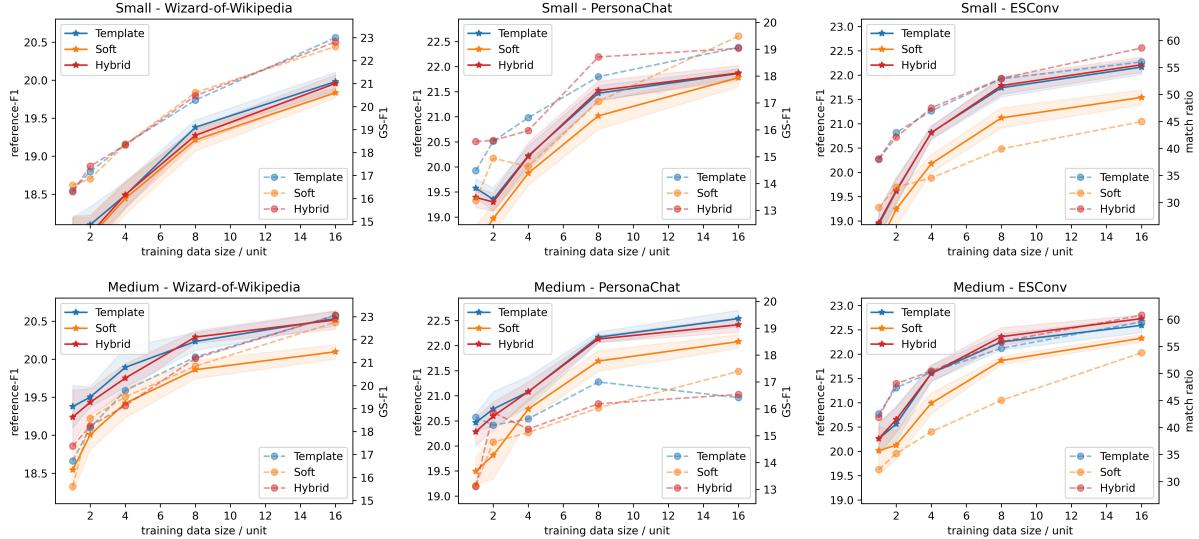


Figure 8: Comparison between the template-based method (Continuation), the soft-prompting method (Prefix), and the hybrid one combining both methods.

may result from the extremely large lengths of GS in WoW (265.5 tokens on average, Section 3.1). In this case, a short prefix may have only a limited impact.

**Model Size** Again, with the medium-sized GPT-2’s stronger ability of fitting more data, the differences between Word Initialization and Prefix get smaller when the training data is more than 4 units. Nevertheless, these two methods are still obviously superior to Random Initialization.

**Prefix Length** We additionally study the influence of prefix length on model performance. The results are shown in Appendix, Figure 10. We find that with the small-sized model and on all the three tasks, varying the prefix length from 20 to 100 does not result in obvious differences. With the medium-sized model and with the extremely little training data (1 unit), WoW benefits more from the larger prefix length (100), which confirms our aforementioned conjecture in (c). In contrast, the smaller length (20) still performs best on both PC and ESCConv.

## 6 Further Discussion

After investigating both template-based prompting and soft-prompting methods, we naturally wonder whether they can be combined to achieve better few-shot learning performance. To this end, we additionally experiment with a hybrid prompting method: on top of the Continuation template, we prepend with a continuous prefix with length 20. Note that we do not claim such combination is

optimal among all the possible candidate combinations, and the combination is selected based on our preceding experimental results in Section 4 and 5.

Evaluation results are shown in Figure 8. With models of both sizes and on three tasks, the template-based method always outperforms the soft-prompting method in terms of both reference-F1 and groundedness scores. It highlights the advantages of the intuitive design of templates, and also indicates a large space for further research on optimizing continuous prompts. However, we can see that combining both methods do not bring more improvement, which even performs slightly worse than the pure template-based method (with the medium-size GPT-2, on WoW and PC). It suggests that the natural language templates already contain abundant task-specific information so that appending continuous prompts does not bring additional help.

**Directions of Future Improvement** Though the template-based prompting methods outperform the soft-prompting ones, the former still needs to search for better templates in an automated way, instead of merely relying on intuition-based manual crafting. To this end, we found in Section 4.3 that the zero-shot performance of language modeling under can serve as a reference for template searching and optimization.

For the soft-prompting methods, we think that their critical bottleneck is the lack of sufficient training data, which makes them difficult to perform few-shot learning for GDG tasks. One potential so-

lution is to apply additional pre-training on the parameters of these soft prompts, as done in (Gu et al., 2021). However, the question that follows is how to design pre-training objectives that are both applicable to unlabeled or unparallel dialog corpora and suitable for the downstream GDG tasks. As we see above, the various task characteristics all influence the prompt-based few-shot learning performance, including the forms of GS, the gaps between DH and GS, and the groundedness of SR, which suggests the need to design specific approaches for different tasks.

Another way of improvement is to combine multiple prompting methods together, namely “multi-prompt learning” (Liu et al., 2021a). While existing works of multi-prompt learning mainly focus on the classification tasks (Jiang et al., 2020; Schick and Schütze, 2021), little attention has been paid to text generation ones, let alone our interested GDG tasks. We believe that by integrating prompting methods with different advantages (for instance, the Continuation and Post-GS templates, template-based prompting methods and pre-trained soft-prompting methods, etc.), the few-shot learning performance can be comprehensively improved. We leave these studies as future work.

## 7 Conclusion

In this paper, we make a first step towards exploring prompt-based few-shot learning for grounded dialog generation (GDG). We conduct extensive experiments to evaluate, compare and analyze both template-based prompting and soft-prompting methods on different GDG tasks. Our findings are summarized as follows. (a) The few-shot learning performance of template-based prompting is positively correlated with the zero-shot performance of language modeling. (b) The gains brought by soft prompts depend on initialization methods and task characteristics. Specifically, the performance of soft-prompting usually benefits from the task-specific information introduced in initialization and the gap in form or style between the grounding sources and the dialogue history. (c) The template-based prompts generally outperform soft ones, while combining both of them does not lead to obvious improvement, which may result from that manually crafted templates contain much richer task-related information than soft prompts.

Correspondingly, there are also three potential directions for future work. (a) Automatic template

searching and optimization based on zero-shot performance of language modeling. (b) Pre-training soft prompts with properly designed objectives that are suitable for downstream GDG tasks. (c) Multi-prompt learning by integrating approaches with different advantages.

## Acknowledgements

This work was supported by the National Science Foundation for Distinguished Young Scholars (with No. 62125604) and the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005.

## References

- 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*.
- 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.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 13063–13075.
- 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.
- Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and

- Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. 2019. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058, Florence, Italy. Association for Computational Linguistics.
- 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.
- Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.
- Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- 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.
- Linxiao Li, Can Xu, Wei Wu, Yufan Zhao, Xueliang Zhao, and Chongyang Tao. 2020. Zero-resource knowledge-grounded dialogue generation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*.
- 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. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021b. Towards emotional support dialog systems. 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 3469–3483, Online. Association for Computational Linguistics.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.
- Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- 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. *Journal of Machine Learning Research*, 21(140):1–67.
- Timo Schick and Hinrich Schütze. 2020. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*.
- Timo Schick and Hinrich Schütze. 2021. 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.
- 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.
- Hao Sun, Zhenru Lin, Chujie Zheng, Siyang Liu, and Minlie Huang. 2021. PsyQA: A Chinese dataset for generating long counseling text for mental health support. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1489–1503, Online. Association for Computational Linguistics.
- Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.
- 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 (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. 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.
- Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020a. Low-resource knowledge-grounded dialogue generation. In *International Conference on Learning Representations*.
- Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020b. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.
- Chujie Zheng, Yunbo Cao, Dixin Jiang, and Minlie Huang. 2020a. Difference-aware knowledge selection for knowledge-grounded conversation generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 115–125, Online. Association for Computational Linguistics.
- Chujie Zheng, Yong Liu, Wei Chen, Yongcai Leng, and Minlie Huang. 2021. CoMAE: A multi-factor hierarchical framework for empathetic response generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 813–824, Online. Association for Computational Linguistics.
- Yinhe Zheng, Rongsheng Zhang, Minlie Huang, and Xiaoxi Mao. 2020b. A pre-training based personalized dialogue generation model with persona-sparse data. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018a. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization.
- Hao Zhou, Chujie Zheng, Kaili Huang, Minlie Huang, and Xiaoyan Zhu. 2020. KdConv: A Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7098–7108, Online. Association for Computational Linguistics.
- Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018b. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

Below is the conversation between a user and an <b>engaging</b> system. The system's utterances are grounded on the given persona: <code>{persona}</code>	... (omitted)... The system's utterances are grounded on the given persona: <code>{persona}</code>  User: POST1 System: RESPONSE1 User: POST2 System: RESPONSE2	... (omitted)...  User: POST1 System: RESPONSE1 User: POST2  Grounded on the given persona, what does the system probably say in the next response? System: RESPONSE2
		The system's utterances are grounded on the given persona: <code>{persona}</code> Grounded on the given persona, what does the system probably say in the next response? System: RESPONSE2

Figure 9: Templates for PersonaChat.

Datasets	Wizard-of-Wikipedia	PersonaChat	ESConv
SI Tokens	[knowledge] [context] [user] [system]	[persona] [context] [user] [system]	[user] [system]
GS Tokens	N/A	N/A	[Question] [Restatement or Paraphrasing] [Reflection of feelings] [Self-disclosure] [Affirmation and Reassurance] [Providing Suggestions] [Information] [Others]
Input Formats	[knowledge] <b>knowledge</b> [context] [user] <b>post</b> [system] <b>response</b>	[persona] <b>persona</b> [context] [user] <b>post</b> [system] <b>response</b>	[user] <b>post</b> [system] <b>strategy</b> <b>response</b>

Table 2: Special tokens used in soft-prompting methods.

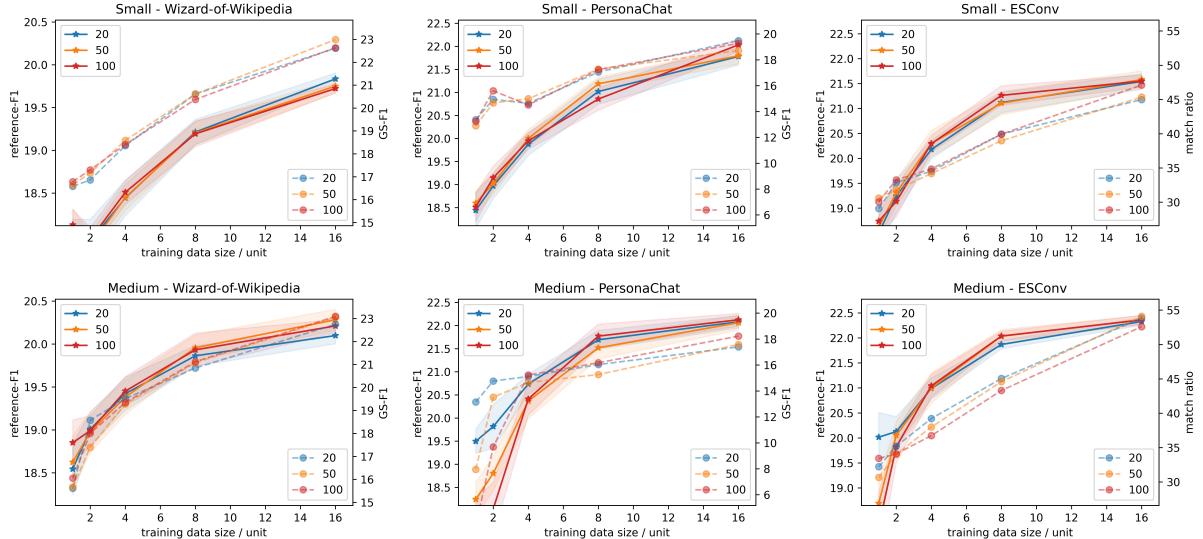


Figure 10: Results of varying prefix length in soft-prompting methods.

## A Textual Explanations for Support Strategies in ESConv

The following explanations are summarized based on the original definitions in (Liu et al., 2021b).

**Question** Ask questions.

**Restatement or Paraphrasing** Rephrase user's statements.

**Reflection of feelings** Reflect and describe user's feelings.

**Self-disclosure** Share similar experiences or emotions.

**Affirmation and Reassurance** Affirm user's

strengths, motivation or capabilities.

**Providing Suggestions** Provide useful suggestions.

**Information** Provide specific information.