Slot Dependency Modeling for Zero-Shot Cross-Domain Dialogue State Tracking

Qingyue Wang[1,2], Yanan Cao[1]*, Piji Li[3], Yanhe Fu[1,2], Zheng Lin[1] and **Li Guo[1]**
1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
2. School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
3. College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing, China
{wangqingyue,caoyanan,fuyanhe,linzheng,guoli}@iie.ac.cn
lipiji.pz@gmail.com

Abstract

Zero-shot learning for Dialogue State Tracking (DST) focuses on generalizing to an unseen domain without the expense of collecting in-domain data. However, previous zero-shot DST methods ignore the slot dependencies in a multi-domain dialogue, resulting in sub-optimal performances when adapting to unseen domains. In this paper, we utilize slot prompts combination, slot values demonstration, and slot constraint object to model the slot-slot dependency, slot-value dependency and slot-context dependency respectively. Specifically, each slot prompt consists of a slot-specific prompt and a slot-shared prompt to capture the shared knowledge across different domains. Experimental results show the effectiveness of our proposed method over existing state-of-art generation methods under zero-shot/few-shot settings.

1 Introduction

Task-oriented dialog systems help users to achieve specific goals using natural languages, such as movie booking and information support. Dialogue state tracking(DST), as a core component of task-oriented dialogue systems, tracks the user's requirements as dialogue states, which are typically in the form of a list of slot-value pairs. In practical applications, the multi-turn conversation usually refers to multiple domains. As shown in Figure 1, a user starts the conversation by asking a hotel and then requests a restaurant with a cheap price range, where *hotel* and *restaurant* are two different domains. At the third turn, the DST extracts multiple (slot, value) pairs like "(hotel-star, 4)" and "(restaurant-pricerange, cheap)" from the dialogue context.

In industrial applications, task-oriented dialogue systems are required to add new domains frequently based on users' needs, but collecting extensive data for every new domain is costly and inefficient.

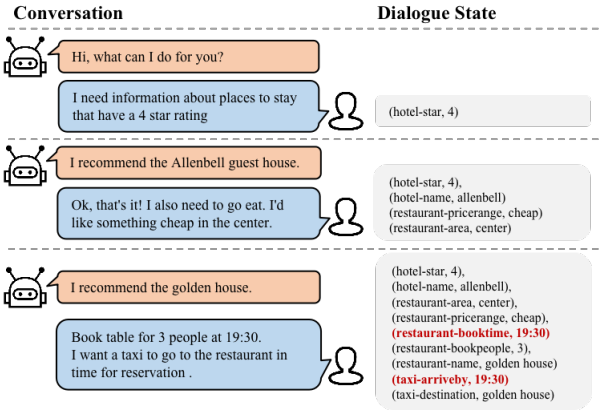


Figure 1: An multi-domain dialogue from MultiWOZ dataset (Budzianowski et al., 2018). Following the convention of this dataset, each slot is represented as a special token concatenated by domain and slot (e.g.,"restaurant-food").

Therefore, performing **zero-shot prediction** of dialogue states is becoming increasingly important since it does not require the expense of data acquisition.

The early works utilized the copy mechanism to handle new slot types in the unseen domain (Wu et al., 2019; Kumar et al., 2020). But the specialized models don't fully leverage the pre-trained language models(PLMs), which have shown impressive ability in transfer learning. Recently, a new paradigm named "prompt-based learning" utilizes language prompts to stimulate the knowledge of PLMs (Han et al., 2021). Compared to task-oriented fine-tuning, prompt-based learning is more similar to pre-training in terms of objectives, thereby adapting to downstream tasks faster even without any training samples. Inspired by it, some researchers add slot-specific prompt[1] into the sequence-to-sequence based model, achieving good performances in zero-shot DST(Lee et al., 2021; Su et al., 2021).

*Corresponding author.

[1]For example, the prompt of slot "restaurant-area" can be "what is the location of the restaurant?".

Proceedings of the 29th International Conference on Computational Linguistics, pages 510–520
October 12–17, 2022.

However, these approaches treat each slot independently, which ignore various slot dependencies during dialogue state tracking. We conceive that there exist several types of slot dependencies in multi-domain DST. For instance, the stars of a hotel and its price range often co-occur in a dialogue state. It could tell that the stars of a hotel might have a dependency on its price range. Take Figure 1 as another example, the user asks for a taxi to the restaurant, meaning that the taxi departure place can be inferred from the name of the hotel. According to the statistics, there are 36.53% slot-slot co-occurrence, 4.29% slot-value co-reference relations and many other types of slot dependencies in the training set of MultiWOZ 2.1 (Budzianowski et al., 2018; Feng et al., 2022). Intuitively, modeling these slot dependencies can help the DST model to handle complex dialogue scenes and infer the slot-value pairs in the zero-shot DST.

Motivated by above analysis, we consider that there are three kinds of slot dependencies, i.e. slot-slot dependency, slot-value dependency and slot-context dependency. This paper proposes a prompt-based approach to model above slot dependencies for zero-shot DST. For the slot-slot dependency, we combine slot prompts as the specialized prompt and decode corresponding slot values, making the model consider semantic information across slots. Specifically, each slot prompt consists of a slot-specific prompt and a slot-shared prompt, which respectively stimulates language understanding and captures the shared knowledge between slots by sharing parameters. For the slot-value dependency, we use value demonstration, i.e., filling partial slot values into slot prompts, to explore possible dependency between slots and values. For the slot-context dependency, we use the masked language model and predict masked tokens inside the context with the constraint of slot values, further enhancing the relationships between slots and dialogue context. The experimental results show that our proposed model achieves a significantly higher joint goal accuracy compared to previous zero-shot DST approaches.

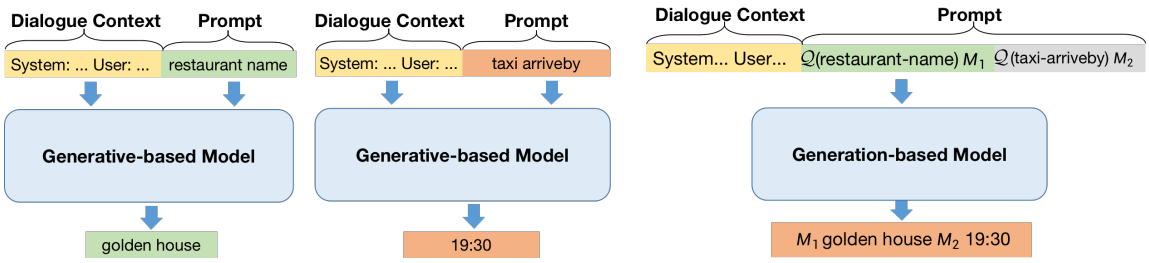
In summary, our main contributions include:

- We propose a prompt-based method for zero-shot cross-domain DST, which leverages slot prompts combination, slot value demonstration and slot constraint object to explore the slot dependency among domains and slots.
- Experimental results show that our approach can transfer into unseen domains effectively and achieve the new state-of-the-art performances on the MultiWOZ 2.1 and SGD dataset under zero/few-shot settings.

2 Related Work

Multi-Domain Dialogue State Tracking Traditional statistical dialogue state tracking models combine semantics extracted by spoken language understanding models to predict the current dialogue state (Williams and Young, 2007; Thomson and Young, 2010) or jointly learn language understanding in an end-to-end way. Recently, many DST models that are built on deep neural networks have achieved promising state tracking results(Dai et al., 2021; Rastogi et al., 2019). Among them, some recent works attempted to model slot relationships by predefined schema graphs (Chen et al., 2020) or attention mechanism (Feng et al., 2021). But they heavily rely on a huge number of annotated data and human efforts without the generalizability to new domains (Feng et al., 2022), which is not suitable for industrial applications. To solve the above problem, some researchers leverage machine reading question answering data to facilitate the low-resource DST (Gao et al., 2020; Lin et al., 2021a), also called cross-task transfer. However, cross-task transfer needs a large-scale corpus and it is hard to learn the semantic consistency with the task-oriented dialogue. In this paper, we focus on the zero-shot cross-domain DST (Wu et al., 2019; Kumar et al., 2020; Lin et al., 2021b),where these models are first trained on several domains and are transferred into unknown domains.

Prompt-based Learning Various recent PLMs like GPT (Radford and Narasimhan, 2018), BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) provide a new approach to utilize large-scale unlabeled data for NLP tasks. However, there is a big gap between pre-training objectives and fine-tuning objectives. Recently, prompt tuning attracts many researchers to design prompt templates and then fine-tune PLMs to downstream tasks, which obtains successful results (Han et al., 2021; Zheng and Huang, 2021). For the DST task, Lee et al. (2021) proposed a slot-specific prompt to augment the multi-domain prompt-based DST model. However, these traditional prompt-based DST approaches handle slots independently while we focus on modeling the dependencies among slots in this paper.



(a) (b)

Figure 2: Traditional prompt-based DST(a) vs Overview of ours(b).

3 Preliminaries

In this section, we give general notations for multi-domain DST task and details about the traditional prompt-based DST, which are the basis of proposed approach in the next section.

3.1 Notations

In task-oriented dialogue systems, a dialogue with T turns can be represented as $\{(A_1, U_1), (A_2, U_2) \ldots, (A_T, U_T)\}$, where A represents the system response and U represents a user utterance. At turn t, we denote the dialogue context as $C_t = \{(A_1, U_1), (A_2, U_2), \ldots, (A_t, U_t)\}$, which includes t turns from system and user. For multi-domain DST, the dialogue state at turn t is represented as a set of (slot, value) pairs, denoted as $B_t = \{(s_j, v_j)|1 \leq j \leq J\}$, where s_j is the slot name given by schema and v_j is its slot value. J is the total number of slots in all domains. If there is no information in the dialogue given about the slot s_j, v_j is set to "none". The goal of DST is to predict the dialogue state B_t given a dialogue context C_t.

3.2 Traditional Prompt-based DST

In this part, we introduce traditional prompt-based DST model (Lin et al., 2021b) with a sequence-to-sequence framework, which is shown in Figure 2(a). A generative model (e.g T5) concatenates dialogue history C_t and a slot-specific prompt $\mathcal{T}_j$ as input and decodes corresponding slot value v_j.

$$v_j = Seq2seq(C_t, \mathcal{T}_j) \tag{1}$$

where $\mathcal{T}_j$ is the prompt for slot s_j. The learning objective of the generation process is minimizing the negative log-likelihood of v_j given context C_t and prompt $\mathcal{T}_j$:

$$L = -\sum_{t}^{T}\sum_{j}^{J} \log p(v_j|C_t, \mathcal{T}_j) \tag{2}$$

The example in Figure 2(a) takes slot name as the slot-specific prompt. For the input with different slots like "restaurant name" and "taxi arriveby", the model generates slot value independently, i.e. "golden house" and "19:30".

4 Methodology

As we mentioned before, we argue that traditional prompt-based DST approaches ignore significant slot dependencies in a dialogue. In this paper, we propose a prompt-based approach to model the dependency of the slot-slot, the slot-value, and the slot-context. The architecture of our model is shown in Figure 3.

4.1 Slot-slot Dependency Modeling

The traditional prompt-based DST utilizes the slot-specific prompt independently. Differently, we compose multiple slot prompts as the final prompt to model the slot-slot dependency. A generation-based model concatenates composed prompt $\mathcal{T}$ and dialogue context C_t as input, and decodes a sequence of values:

$$\begin{aligned} \mathcal{T} &= \text{“}\mathcal{Q}(s_1), M_1, ..., \mathcal{Q}(s_J), M_J\text{”} \\ \mathcal{V} &= Seq2seq(C_t, \mathcal{T}) \end{aligned} \tag{3}$$

where $\mathcal{Q}(s_i)$ refers the slot prompt of slot s_i. Here, $\mathcal{V} =$ "$M_1, v_1, \ldots, M_J, v_J$" and M_* are mask tokens. Each mask token is inserted after the slot prompt and it is also the start token for a slot value. We insert mask tokens inside the prompt because we hope the model can focus on the specific slot prompt when generating its corresponding value.

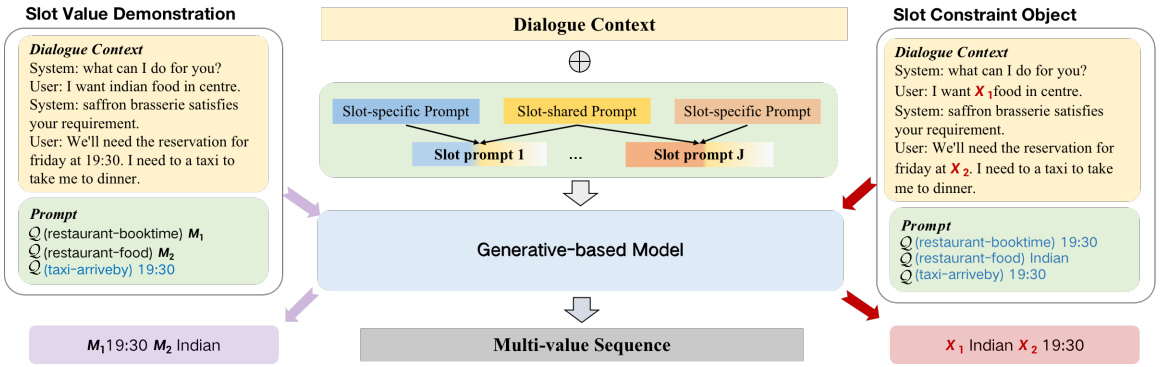


Figure 3: The architecture of our proposed model. It concatenates dialogue context (in yellow) and several slot prompts (in green) to model the slot-slot dependency. The model explores slot-value dependency and slot-context dependency utilizing slot value demonstration (left-shown example) and slot constraint object (right-shown example). Noted that M_* and X_* are mask tokens and $\mathcal{Q}(slot)$ represents a slot prompt.

The objective of the generative model is to minimize the sequence of values given context C_t and prompt $\mathcal{T}$:

$$L = -\sum_{t}^{T} \log p(\mathcal{V}|C_t, \mathcal{T}) \tag{4}$$

Figure 2(b) shows the overview of our method, in which the model concatenates a dialogue context and two slot prompts, "restaurant-name" and "taxi-arriveby", and then generates the sequence "M_1 golden house M_2 19:30".

Slot Prompt Design For each slot, we utilize two types of prompts, a slot-specific prompt and a slot-shared prompt to construct the slot prompt. The slot-specific prompt stimulates the language understanding from PLMs and the slot-shared prompt capture the universal knowledge across slots.

Formally, we define a slot-specific prompt as $\{P_1^k, P_2^k, \ldots, P_I^k\}$ for the slot s_k, and a slot-shared prompt as $\{P_1', P_2', \ldots, P_Q'\}$ for all slots. The I and Q are the number of slot-specific tokens and pseudo tokens respectively. For a slot s_k, its slot prompt $\mathcal{Q}(s_k)$ is written as:

$$\{P_1^k, \ldots, P_I^k, P_1', \ldots, P_Q'\} \tag{5}$$

For instance, the slot prompt of "restaurant-name" can be "restaurant name $[P_1'], [P_2']$"[2]. The slot prompt embedding of slot s_k is represented as follows:

$$PE(s_k) = \{e_1^k, \ldots, e_I^k, h_1', \ldots, h_Q'\} \tag{6}$$

[2]We try different value of Q and the optimal value 2 is selected using the validation set

where e_* are original word embeddings. h_*' are trainable embedding tensors, which are encoded by a full-connected network and share parameters across slots.

4.2 Slot-Value Dependency Modeling

Except for the dependency among slots, we find that many slot values are also highly correlated, i.e. demonstrated by co-reference and exclusion. For example, in a dialogue, the value of "taxi-departure" might be inferred from "hotel-name" but must be different from the value of "taxi-destination". We suppose that considering other slot values helps the model to capture the slot-value dependency and understand the dialogue context better.

Specifically, we introduce some ground-truth slot values into the prompt $\mathcal{T}$, called **slot value demonstration**. Since there are multiple mask tokens in prompt, we replace each mask token with its slot value at ratio β (a hyper-parameter). The left example in Figure 3 takes an input with three slot prompts and one of them is supplied with the slot value (in blue). Accordingly, the model only needs output two slot values, "19:30" for "restaurant-booktime" and "Indian" for "restaurant-food".

4.3 Slot-Context Dependency Modeling

To model the dependency between slots and dialogue context, we introduce a **slot constraint object** with a masked language model. Specifically, we first utilize ground-truth slot values to fill mask tokens, obtaining a new prompt $\tilde{\mathcal{T}}$. After that, we use other symbols X_* to mask v_* inside the context. The slot constraint objective is to predict the masked values sequence given context $\tilde{C}_t$ and

prompt $\tilde{\mathcal{T}}$:

$$\tilde{\mathcal{T}} = \text{“}\mathcal{Q}(s_1), v_1, \ldots, \mathcal{Q}(s_J), v_J\text{”}$$
$$L_{sc} = -\sum_{t}^{T} \log p(\mathcal{W}|\tilde{C}_t, \tilde{\mathcal{T}}) \quad (7)$$

where $\mathcal{W}$ = “$X_1, w_1, X_2, w_2, \ldots, w_Z$” and w_i refers the masked value inside context. For the slot that its value is unable to match strings in dialogue context (e.g “none”), we skip the mask operation to it. Therefore, the number of masked values Z might not be equal to the number of slot prompts, and w_i might not actually be v_i.

Take the right-shown example in Figure 3 for illustration. Although there are three slots in the prompt, only two mask symbols (X_1 and X_2) are used in the context. The reason is that the value of “taxi-arriveby” is inferred from “restaurant-booktime”, causing the same location for their values in context.

4.4 Training and Inference

During training, we have the following loss function:

$$L_{train} = -\sum_{t}^{T} \log p(\mathcal{V}|C_t, \mathcal{T}) + \lambda L_{sc} \quad (8)$$

where $\mathcal{T}$ is a specific prompt using slot value demonstration. λ is a hyper-parameter and controls the weight of slot constraint object. During Inference, considering that the number of unseen domains and slots might be huge, we concatenate single slot prompt and dialogue context as input to predict slot value, just like traditional prompt-based DST (shown in Section 3.2).

5 Experiments

5.1 Datasets

We evaluate the proposed model on the most popular multi-domain task-oriented dialogue benchmarks, MultiWOZ (Budzianowski et al., 2018; Eric et al., 2020) and Schema-Guided-Dialogue(SGD) (Raffel et al., 2020). Both datasets provide turn-level annotations of dialogue states and descriptions of domain and slot. The MultiWOZ dataset contains over 10K dialogue across 8 domains. We follow the previous pre-processing and evaluation setup (Lin et al., 2021b), where the restaurant, train, attraction, hotel, and taxi domains are used for training and testing. Appendix A gives more statistics of MultiWOZ datasets. The SGD dataset has over 16K dialogues in the training set, spanning 26 services belonging to 16 domains. The test set has 18 domains, and 5 domains of them are not presented in the training set.

5.2 Baselines

We compare the performance of our model with the following existing models. **TRADE** (Wu et al., 2019) leverages context-enhanced slot gate and copy mechanism to track slot values mentioned in dialogue history . **SUMBT** (Lee et al., 2019) proposes a non-parametric method to score each candidate slot-value pair in a predefined ontology . **MA-DST** (Kumar et al., 2020) designs multiple layers of cross-attention to capture relationships at different levels of dialogue granularity. **DSTQA** (Zhou and Small, 2019) models the DST task as a question answering problem and uses a dynamically-evolving knowledge graph to learn the relationships between domains . **T5DST** (Lin et al., 2021b) is a strong prompt baseline that first uses slot descriptions as a prompt in zero-shot cross-domain DST. **TransferQA** (Lin et al., 2021a) is a cross-task zero-shot DST method where the model is pre-trained on question answering data first and then is applied to unseen domains. **SGD-baseline** (Rastogi et al., 2020) uses schema descriptions and applies a BERT-based DST model to predict the dialogue state of unseen domains.

5.3 Evaluation

Following previous works (Lin et al., 2021b), we use Joint Goal Accuracy(JGA) and Average Goal Accuracy (AGA) to evaluate our models and baselines. Joint goal accuracy is the percentage of turns for which all the slots are correctly identified. Average goal accuracy is the average accuracy of the *active* slots in each turn. A slot becomes *active* if its value is mentioned in the current turn and is not inherited from previous turns. We compute JAG per domain in MultiWOZ datasets and use the official evaluation script in SGD dataset.

In zero-shot settings, all models are trained on four domains in the MultiWOZ dataset then zero-shot on the held-out domain. In the SGD dataset, there are 5 domains in the testing set but are not in the training set, so all models are trained with the whole training set and tested on these 5 unseen domains. For few-shot experiments in MultiWOZ dataset, all models are first trained on 4 source domains and then fine-tuned with 1%, 5%, and

Model	Pretrained-Model	Joint Goal Accuracy					
		Attraction	Hotel	Restaurant	Taxi	Train	Average
TRADE (Wu et al., 2019)	N	20.06	14.20	12.59	59.21	22.39	25.69
MA-DST (Kumar et al., 2020)	N	22.46	16.28	13.56	59.27	22.76	26.87
SUMBT (Lee et al., 2019)	Bert-base	22.60	19.08	16.50	59.50	22.50	28.18
T5DST † (Lin et al., 2021b)	T5-small	31.92	20.72	20.09	64.12	28.83	33.56
Ours †	T5-small	33.92	19.85	20.75	66.25	36.96	35.55
T5DST †* (Lin et al., 2021b)	T5-base	35.51	22.48	25.04	65.93	34.82	36.25
TransferQA †(Lin et al., 2021a)	T5-large	31.25	22.72	26.28	61.87	36.72	35.77
Ours †	T5-base	**37.83**	**26.50**	**27.05**	**69.23**	**40.27**	**40.18**

Table 1: Zero-shot results on MultiWOZ 2.1. All numbers are reported in joint goal accuracy(%). The averaged zero shot joint goal accuracy among five domains is reported. All results of baselines are from the original public papers, except for T5DST* where we rerun their code with T5-base. † means the model is a prompt-based method. For fair comparison, all prompt-based methods use the slot-description provided from schema as slot-specific prompt.

10% of target domain data. The zero-shot/few-shot settings are consistent with the previous works on zero-shot cross-domain DST (Wu et al., 2019; Lin et al., 2021a,b).

5.4 Implementation

We implement our approach based on T5-small (60M parameters) and T5-base (220M parameters) (Raffel et al., 2020). We train the model with a batch size of 128 for T5-small and a batch size of 256 for T5-base. Both of them are trained using AdamW optimizer (Loshchilov and Hutter, 2019). The peak learning rate is set to 1e-4 for T5 and 2e-4 for other learned modules. To balance the efficiency and performance of the model, we adopt a random sampling strategy in slot prompts combination, i.e. setting a hyper-parameter α as the max number of slot prompts. Given a dialogue context, the model randomly selects 1 up to α slots to construct the prompt. And we apply multiple iterations for training so that almost all slots can be sampled. In all experiments, the α is set to 3, β is set to 0.5 and the weight λ in loss function is 0.3. We use greedy decoding for all models.

6 Main Results

6.1 Zero-Shot Cross-Domain Results

Table 1 gives the results of our model and baselines under the zero-shot setting. Compared to previous works, our model using T5-base achieves significantly higher JGA (3.93% on average) and even exceeds the cross-task method using T5-large (TransferQA). Among these baselines, the methods (T5DST and TransferQA) using T5 model have much better performances than those without pre-trained models (TRADE and MA-DST). We analyze that T5 is pre-trained on a large unlabeled corpus, which can provide a promising language understanding for unseen slots. Notably, our method using T5-small outperforms prior prompt-based DST (T5DST) on almost domains, except for hotel domain. Appendix B shows that our method exceeds T5DST on six hotel slots but falls behind on four slots. The reason is that these four slots are completely independent from source domains, making their prediction mainly depend on the ability of language understanding. Our proposed model with small trainable parameters tends to build the slot dependencies, which might hurt partial language understanding in PLMs. However, our model with T5-base brings obvious improvements on all domains, including hotel domain. It also verifies that the stronger ability of language understanding the pre-trained model has, the easier to benefit from slot dependency modeling our method is.

Table 3 summarizes the zero-shot results on SGD dataset. Compared with SGD-baseline, the zero-shot performance of our model is consistently higher in five unseen domains. Compared to transferQA with T5-large and large labeled training data (QA dataset), our model with T5-small is still competitive in zero-shot settings. Particularly, our model gains a great improvement on "bus" and "train" domains. We analysis that these two domains are closely related to some seen domains, e.g "flight" and "travel" domains, which easily benefit from the slot dependency modeling.

6.2 Few-Shot Cross-Domain Results

We further conduct experiments in few-shot cross-domain settings on MultiWOZ 2.0, as in (Wu et al., 2019; Lin et al., 2021a,b). The models are first trained on 4 domains and then fine-tuned with 1%,

Model	Attraction			Hotel			Restaurant			Taxi			Train		
	1%	5%	10%	1%	5%	10%	1%	5%	10%	1%	5%	10%	1%	5%	10%
TRADE	35.88	57.55	63.12	19.73	37.45	41.42	42.42	55.70	60.94	63.81	66.58	70.19	59.83	69.27	71.11
DSTQA	N/A	**70.47**	71.60	N/A	50.18	53.68	N/A	58.95	64.51	N/A	70.90	74.19	N/A	70.35	74.50
T5DST	58.77	65.72	69.54	43.07	50.71	54.86	57.63	61.86	63.47	70.12	73.67	74.70	70.82	74.18	77.57
Our Approach	**60.03**	69.69	**71.61**	**45.76**	**52.53**	**56.71**	**60.56**	**64.24**	**67.31**	**76.23**	**78.32**	**79.61**	**70.93**	**75.50**	**77.89**

Table 2: Few-shot experiments in MultiWOZ 2.0. The experiments are conducted on MultiWOZ 2.0 for comparing with previous works. N/A represents the results are not reported in the original paper.

Domain	SGD-baseline	TransferQA	Seq2seq-DU	Ours
Bus	9.7/50.9	15.9/63.6	16.8/N	43.9/86.3
Messaging	10.2/20.0	13.3/37.9	4.9/N	36.6/61.4
Payment	11.5/34.8	24.7/60.7	7.2/N	16.5/62.0
Trains	13.6/63.5	17.4/64.9	16.8/N	46.7/86.9
Alarm	57.7/1.8	58.3/81.7	55.6/N	58.3/87.5
Average	20.5/34.2	25.9/61.8	20.3/N	**40.4/76.8**

Table 3: Zero-Shot results on SGD dataset. All results are reported in JGA(%)/AVG(%). Seq2seq-DU(Feng et al., 2021) is seq2seq baseline without any pre-trained model. N represents the results are not reported in the original paper.

Models	Pretrained-Model	Zero-shot	JGA
TRADE (Wu et al., 2019)	N	✔	45.60
STARC (Gao et al., 2020)	Bert-base	✔	49.48
SGD-baseline (Rastogi et al., 2020)	Bert-base	✔	43.40
T5DST (Lin et al., 2021b)	T5-small	✔	51.91
T5DST (Lin et al., 2021b)	T5-base	✔	53.15
Ours	T5-small	✔	52.83
Ours	T5-base	✔	**54.75**
MinTL(Lin et al., 2020)	BART	✘	50.95
SOM-DST(Kim et al., 2020)	Bert-base	✘	53.68
Tripy (Heck et al., 2020)	Bert-base	✘	55.29
Simple-TOD (Hosseini-Asl et al., 2020)	GPT-2	✘	55.72

Table 4: Full data results on MultiWOZ 2.1 dataset.

5%, and 10% of target domain data. In Table 3, the experiment result shows that DSTQA is a competitive baseline. However, our approach outperforms previous transfer-learning methods in almost all domains, except for the situation with 5% *Attraction* domain data fine-tuning. We suppose that the DSTQA introduces an extra schema graph to model explicit relationships across slots. The significant improvements on most domains indicate that our model still keeps a robust learning ability with a minute quantity of dialogue fine-tuning.

6.3 Full Data Results

We also evaluate our model on full dataset to understand the full-shot performance, and the results are shown in Table 4. Compared with prior models with zero-shot capability, our model improves the joint goal accuracy by 1.6% in MultiWOZ 2.1 dataset. Particularly, our model exceeds traditional prompt-based methods, T5DST, illustrating that modeling slot dependency is helpful even in a full-data scene. We notice that many training strategies can be applied into the full-data experiment, such as additional supervision (Chen et al., 2020) and pre-process strategies (Heck et al., 2020), that may improve the performances. In this paper, we focus on modeling slot dependency for zero-shot DST not achieving state-of-art on full-data.

7 Discussion

7.1 Ablation Study

In Table 5, we study the effect of different modules for the proposed model in the zero-shot setting. Firstly, we set the hyper-parameter α as 1 to check the effect of several slot prompts. There has 2% drop of performance on hotel and taxi domain. Secondly, we only use the slot-specific prompt to investigate the effect of the slot-shared prompt. One can observe that the performance deteriorates considerably, which is similar to the results of removing composing slot prompts. It indicates that composing specialized slot prompts can enhance the pre-trained model's ability on predicting unseen domains and slots. Thirdly, we explore the effect of slot value demonstration by setting the ratio β as 0. The performance of the model decreases markedly, especially for taxi domain. We conclude that value demonstration in prompt can effectively explore the slot-value dependency, such as co-reference and exclusion. These relationships mainly occur in some slots related to time or location, causing a huge influence on taxi domain. Furthermore, the model without slot constraint object performs declining results on different domains, which illustrates that learning the slot-context dependency is also important for zero-shot learning.

7.2 Analysis of Parameters

We further investigate the impacts of hyper-parameter settings on the performance of the proposed model on MultiWOZ2.1 in zero-shot set-

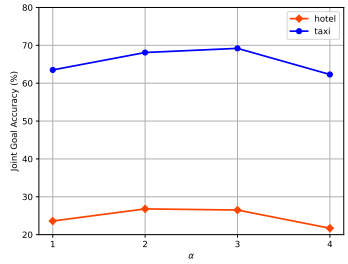


Figure 4: Effects of the max number of slot prompts α.

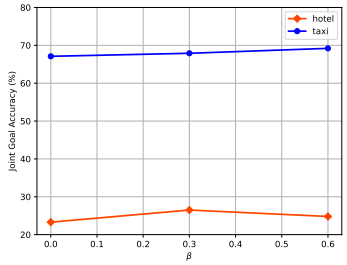


Figure 5: Effects of ratio of β in value demonstrations.

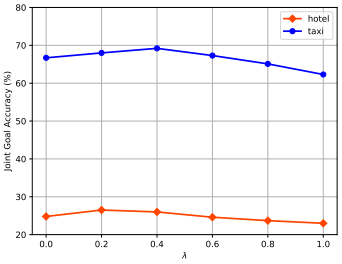


Figure 6: Effects of the weight of slot constraint object.

Model	Joint Goal Accuracy	
	Hotel	Taxi
Our approach	**26.5**	**69.2**
w/o Slot Prompt Combination	24.9(-1.6)	67.1(-2.1)
w/o Slot-shared Prompt	25.3(-1.2)	67.9(-1.3)
w/o Slot Value Demonstration	25.8(-0.7)	67.4(-1.8)
w/o Slot Constraint Object	25.9(-0.6)	67.9(-1.3)

Table 5: Ablation studies on the MultiWOZ 2.1 in zero-shot setting on target domain *hotel* and *taxi*.

tings. We validate the effects of three factors: the max number α of slot prompts, the ratio of value demonstrations β, and the weight λ in the loss function. Figure 4, 5, 6 show the results of proposed model with varying parameters in zero-shot setting on domain hotel and taxi. We observe that the optimal parameters are not completely consistent across different domains. In Figure 4 and 5, the model achieves a better performance with more slot prompts and a bigger ratio of value demonstrations on taxi domain. We conjecture that taxi domain only has four slots ("taxi-departure" , "taxi-destination", "taxi-arriveby" , and "taxi-leaveat") and all of them are related to source domains, such as the co-reference between "hotel-name" and "taxi-destination" . That means that exploring the slot-value dependency has a bigger influence on taxi domain than hotel domain. Figure 6 show the effect of using different weight λ in the loss function. When the weight of the slot constraint object is too low, the model doesn't own enough strong constraint for slot-context dependency; when it is too high, the model tends to over-predict "masked tokens" not track dialogue state. Finally, we find that our model achieves a balance with 0.2∼0.4.

7.3 Case Study

In Figure 7, we make a qualitative analysis of the results of T5DST and our method on the MultiWOZ dataset under zero-shot settings. From the results, we find that both models accurately predict the "Ballare" of "taxi-destination" and "17:30" of "taxi-arriveby". These two slot values are easy to predict because they don't depend on any other domains and slots. Besides, our method generates the "lovell lodge" for "taxi-departure", while the T5DST model outputs a wrong value, i.e., "none". We analyze that the T5DST leverages slot-specific prompt and generates slot value independently, which can not infer the relations between slots and values. Our approach leverages slot prompts combination and slot value demonstrations, making it possible to model the slot-slot and slot-value dependencies.

Dialogue Context	System: hi. what can i do for you? User: Can you give me information on an attraction called ballare ? System:The Ballare is located in Heidelberg Gardens User: Thanks. I'm also looking for somewhere to stay in the north. System: Would you want to try the lovell lodge. User: Let s do that .Can you help me to book a taxi from the hotel to the Ballare. I want to leave by 17:30 .
Golden Dialogue State	(taxi-leaveat,17:30) (taxi-destination,ballare) (taxi-departure,lovell lodge)
T5DST	(taxi-leaveat,17:30) (taxi-destination,ballare) (taxi-departure,none)
Ours	(taxi-leaveat,17:30) (taxi-destination,ballare) (taxi-departure,lovell lodge)

Figure 7: The zero-shot evaluation results for T5DST vs. Ours. We mark the key information in blue and the wrong prediction in red.

8 Conclusion and Future Work

In this paper, we attempt to model three slot dependencies for zero-shot cross-domain DST, i.e. slot-slot dependency, slot-value dependency, and slot-context dependency. Experimental results on popular datasets show that the proposed approach performs much better than baselines in zero-shot/few-

shot settings. In the future, we would like to explore more ways to model slot dependency effectively.

Acknowledgements

This work is supported by the Youth Innovation Promotion Association of the Chinese Academy of Sciences (No. 2018192), the NSFC (No. 61902394). We would like to thank the anonymous reviewers for their valuable comments.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *ArXiv*, abs/1810.00278.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *AAAI*.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. 2021. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking. *ArXiv*, abs/2106.00291.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Anuj Kumar Goyal, Peter Ku, Sanchit Agarwal, and Shuyang Gao. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *LREC*.

Yue Feng, Aldo Lipani, Fanghua Ye, Qiang Zhang, and Emine Yilmaz. 2022. Dynamic schema graph fusion network for multi-domain dialogue state tracking. In *ACL*.

Yue Feng, Yang Wang, and Hang Li. 2021. A sequence-to-sequence approach to dialogue state tracking. In *ACL*.

Shuyang Gao, Sanchit Agarwal, Tagyoung Chung, Di Jin, and Dilek Z. Hakkani-Tür. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. *ArXiv*, abs/2004.05827.

Xu Han, Weilin Zhao, NNN, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *ArXiv*, abs/2105.11259.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gavsi'c. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. *ArXiv*, abs/2005.02877.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *ArXiv*, abs/2005.00796.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *ACL*.

Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Z. Hakkani-Tür. 2020. Ma-dst: Multi-attention based scalable dialog state tracking. In *AAAI*.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. *ArXiv*, abs/2109.07506.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *ACL*.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Paul A. Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021a. Zero-shot dialogue state tracking via cross-task transfer. In *EMNLP*.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul A. Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021b. Leveraging slot descriptions for zero-shot cross-domain dialogue statetracking. In *NAACL*.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. *ArXiv*, abs/2009.12005.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam M. 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. *ArXiv*, abs/1910.10683.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *ArXiv*, abs/1909.05855.

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Lambert Mathias. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In *NAACL*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *ArXiv*, abs/2109.14739.

Blaise Thomson and S. Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Comput. Speech Lang.*, 24:562–588.

J. Williams and S. Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Comput. Speech Lang.*, 21:393–422.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, R. Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *ACL*.

Chujie Zheng and Minlie Huang. 2021. Exploring prompt-based few-shot learning for grounded dialog generation. *ArXiv*, abs/2109.06513.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *ArXiv*, abs/1911.06192.

A Dataset Statistics

The MultiWOZ dataset is a fully-labeled collection of human-human written conversations spanning multiple domains and topics. Some statistics of MultiWOZ 2.1 are reported in Table 6. We further draw a schema graph to illustrate the slot dependency, which is shown in Figure 8.

Domain	**Slot**	*Train*	*Valid*	*Test*
Attraction	area, name, type	2717	401	395
Hotel	area, internet, name, parking, price range, stars, type, book day, book people, book stay	3381	416	394
Restaurant	area, food, name, price range, book day, book people, book time	3813	438	437
Taxi	arrive by, departure, destination, leave at	1654	207	195
Train	arrive by, day, departure, destination, leave at, book people	3103	484	494
Total		8438	1000	1000

Table 6: The dataset statistics of MultiWOZ dataset.

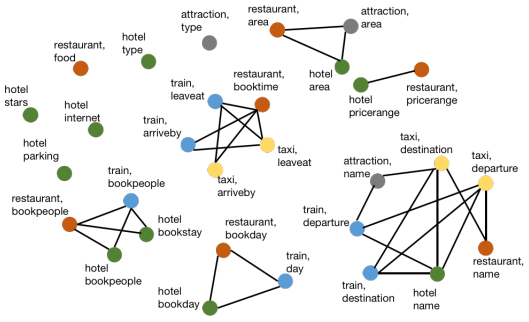


Figure 8: The schema graph on MultiWOZ dataset. Each nodes represents a slot and the nodes in same color belong to a domain. There is an edge between two nodes if some of their candidate values are same.

B Performance on Per-Slot

Figure 9 shows the difference in performance between T5DST and ours in the hotel domain when using T5-small. From the results, our method exceeds T5DST on six slots while falling behind on four slots, i.e "stars", "internet", "type" and "parking". In Figure 10, we list the performance of per-slot on taxi domain when using T5-small. There are four slots in taxi and all of them are related to

source domains. Our method can effectively handle these slots due to the modeling of slot dependency.

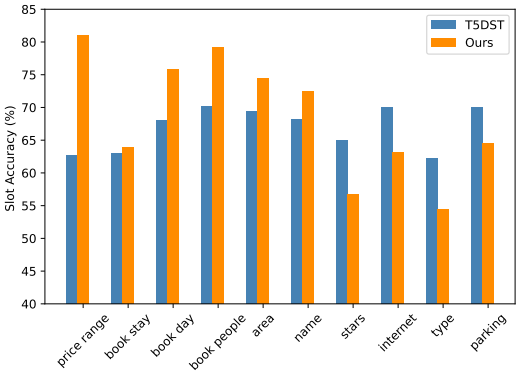


Figure 9: Slot Accuracy in hotel of MultiWOZ 2.1.

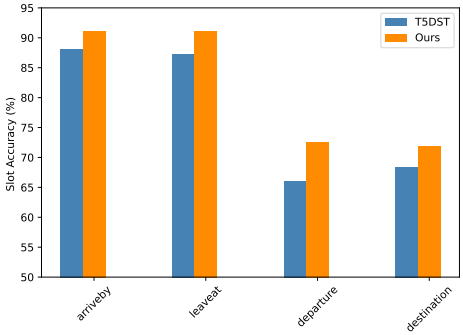


Figure 10: Slot Accuracy in taxi of MultiWOZ 2.1.