Dialogue State Tracking with a Language Model using Schema-Driven Prompting

Chia-Hsuan Lee
University of Washington
chiahlee@uw.edu

Hao Cheng
Microsoft Research
chehao@microsoft.com

Mari Ostendorf
University of Washington
ostendor@uw.edu

Abstract

Task-oriented conversational systems often use dialogue state tracking to represent the user's intentions, which involves filling in values of pre-defined slots. Many approaches have been proposed, often using task-specific architectures with special-purpose classifiers. Recently, good results have been obtained using more general architectures based on pre-trained language models. Here, we introduce a new variation of the language modeling approach that uses schema-driven prompting to provide task-aware history encoding that is used for both categorical and non-categorical slots. We further improve performance by augmenting the prompting with schema descriptions, a naturally occurring source of in-domain knowledge. Our purely generative system achieves state-of-the-art performance on MultiWOZ 2.2 and achieves competitive performance on two other benchmarks: MultiWOZ 2.1 and M2M. The data and code will be available at https://github.com/chiahsuan156/DST-as-Prompting.

1 Introduction

In task-oriented dialogues, systems communicate with users through natural language to accomplish a wide range of tasks, such as food ordering, tech support, restaurant/hotel/travel booking, etc. The backbone module of a typical system is dialogue state tracking (DST), where the user goal is inferred from the dialogue history (Henderson et al., 2014; Shah et al., 2018; Budzianowski et al., 2018). User goals are represented in terms of values of pre-defined slots associated with a schema determined by the information needed to execute task-specific queries to the backend. In other words, user goals are extracted progressively via slot filling based on the schema throughout the conversation. In this paper, we focus on multi-domain DST where the dialogue state is encoded as a list of triplets in the form of (*domain, slot, value*), e.g. (*"restaurant", "area", "centre"*).

There are two broad paradigms of DST models, *classification-based* and *generation-based* models, where the major difference is how the slot value is inferred. In classification-based models (Ye et al., 2021; Chen et al., 2020), the prediction of a slot value is restricted to a fixed set for each slot, and non-categorical slots are constrained to values observed in the training data. In contrast, generation-based models (Wu et al., 2019; Kim et al., 2020) decode slot values sequentially (token by token) based on the dialogue context, with the potential of recovering unseen values. Recently, generation-based DST built on large-scale pretrained neural language models (LM) achieve strong results without relying on domain-specific modules. Among them, the autoregressive model (Peng et al., 2020a; Hosseini-Asl et al., 2020) uses a uni-directional encoder whereas the sequence-to-sequence model (Lin et al., 2020a; Heck et al., 2020) represents the dialogue context using a bi-directional encoder.

In this study, we follow a generation-based DST approach using a pre-trained sequence-to-sequence model, but with the new strategy of adding task-specific prompts as input for sequence-to-sequence DST models, inspired by *prompt-based* fine-tuning (Radford et al., 2019; Brown et al., 2020a). Specifically, instead of generating domain and slot symbols in the decoder, we concatenate the dialogue context with domain and slot prompts as input to the encoder, where prompts are taken directly from the schema. We hypothesize that jointly encoding dialogue context and schema-specific textual information can further benefit a sequence-to-sequence DST model. This allows task-aware contextualization for more effectively guiding the decoder to generate slot values.

Although the domain and slot names typically have interpretable components, they often do not reflect standard written English, e.g. "*arriveby*" and "*ref*". Those custom meaning representations are

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4937–4949
November 7–11, 2021. ©2021 Association for Computational Linguistics

typically abbreviated and/or under-specified, which creates a barrier for effectively utilizing the pretrained LMs. To address this issue, we further incorporate natural language schema descriptions into prompting for DST, which include useful information to guide the decoder. For example, the description of "*ref*" is "*reference number of the hotel booking*"; the values of "*has_internet*" are "*yes*", "*no*", "*free*", and "*don't care*".

In short, this work advances generation-based DST in two ways. First, candidate schema labels are jointly encoded with the dialogue context, providing a task-aware contextualization for initializing the decoder. Second, natural language descriptions of schema categories associated with database documentation are incorporated in encoding as prompts to the language model, allowing uniform handling of categorical and non-categorical slots. When implemented using a strong pretrained text-to-text model, this simple approach achieves state-of-the-art (SOTA) results on MultiWOZ 2.2, and performance is on par with SOTA on MultiWOZ 2.1 and M2M. In addition, our analyses provide empirical results that contribute towards understanding how schema description augmentation can effectively constrain the model prediction.

2 Related Work

2.1 Multi-Domain Dialogue State Tracking

Task-oriented dialogue datasets (Shah et al., 2018; Henderson et al., 2014), have spurred the development of dialogue systems (Zhong et al., 2018; Chao and Lane, 2019). Recently, to further examine the generalization abilities, large scale cross-domain datasets have been proposed (Budzianowski et al., 2018; Zang et al., 2020; Eric et al., 2019; Rastogi et al., 2020b). *Classification-based* models (Ye et al., 2021; Chen et al., 2020) pick the candidate from the oracle list of possible slot values. The assumption of the full access of the schema makes them have limited generalization abilities. On the other hand, *generation-based* models (Wu et al., 2019; Kim et al., 2020; Lin et al., 2020a) directly generate slot values token by token, making it possible to handle unseen domains and values. Most of these models require task-specific modular designs.

Recently, generation-based models that are built on large-scale autoregressive pretrained language models (Ham et al., 2020; Hosseini-Asl et al., 2020; Peng et al., 2020a) achieve promising state tracking results on MultiWOZ 2.0 and 2.1 when trained on additional supervision signals or dialogue corpus. Both Ham et al. (2020) and Hosseini-Asl et al. (2020) require dialogue acts as inputs. Both Hosseini-Asl et al. (2020) and Peng et al. (2020a) require DB search results as inputs. Peng et al. (2020a) also leverages other dialogue corpora to finetune the language model. Our work requires only the dialogue state labels and does not utilize any external dialogue datasets.

2.2 Language Models

Large-scale pretrained language models have obtained state-of-the-art performance on diverse generation and understanding tasks including bi-directional encoder style language models (Devlin et al., 2019; Liu et al., 2019), auto-regressive language models (Radford et al., 2019; Brown et al., 2020b) and more flexible sequence-to-sequence language models (Raffel et al., 2020). To adapt to dialogue tasks, variants of systems are finetuned on different dialogue corpora including chit-chat systems (Zhang et al., 2020; Adiwardana et al., 2020; Roller et al., 2020) and task-oriented dialogue systems (Mehri et al., 2019; Wu et al., 2020; Henderson et al., 2020; Peng et al., 2020b). We leave it as future work to leverage domain-adapted language models.

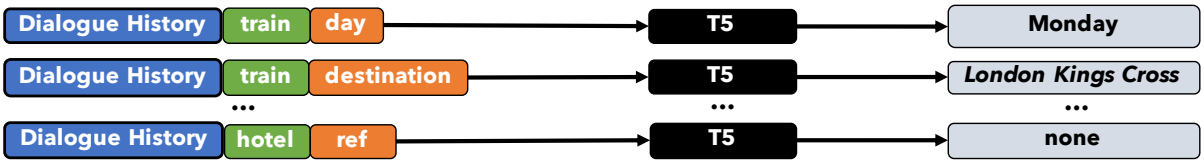
2.3 Prompting Language Models

Extending a language model's knowledge via prompts is an active line of research. Radford et al. (2019) obtain empirical success by using prompts to guide zero shot generation without finetuning on any prompts. Raffel et al. (2020) uses task-specific prompts in both finetuning and testing phase. Recent studies have also tried to automatically discover prompts rather than writing them by humans (Jiang et al., 2020). Our proposed prompting method is largely inspired by this body of work. Instead of prompt engineering/generation, we focus on using available natural language descriptions of schema categories associated with database documentation as task-specific promptings for DST.

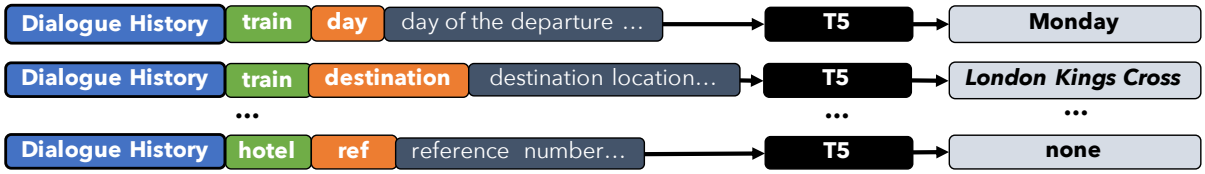
3 Prompting Language Model for Dialogue State Tracking

In this section, we first set up the notations that are used throughout paper, and then review the generative DST with the sequence-to-sequence framework. Based on that, we formally introduce our prompt-based DST model and the corresponding backbone

(a) Generation-based DST w/ Sequential Decoding



(b) Schema-Based Prompt DST w/ Independent Decoding



(c) Natural Language Augmented Prompt DST w/ Independent Decoding

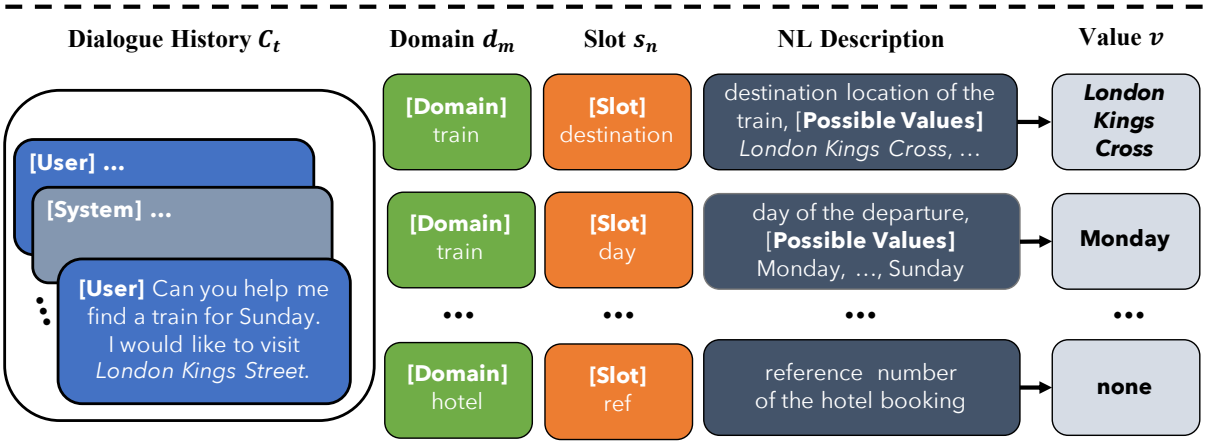


Figure 1: Overview of generative DST approaches for multi-domain scenario. The top three figures illustrate three different generative approaches considered in this paper and the bottom figure includes specific examples for dialogue history, domain names, slot names, natural language descriptions (types, set of valid values, etc.) for slots. Sub-figure (b)(c) demonstrate two prompt-based DST models proposed, where method in (c) includes additional natural language description of slots considered for tracking. Domain descriptions are omitted for brevity.

pretrained model.

Notation. For task-oriented dialogues considered in this paper, a dialogue consists of a sequence of utterances alternating between two parties, $U_1, A_1, ..., U_T, A_T$, where U and A represent the user utterance and the system response, respectively. In a turn t, the user provides a new utterance U_t and the system agent responds with utterance A_t. As shown in the bottom of Figure 1, at turn t, we denote the dialogue context as $C_t = \{U_1, A_1, \ldots, A_{t-1}, U_t\}$, which excludes the latest system response A_t. In this work, we assume a multi-domain scenario, in which case the schema contains M domains $\mathcal{D} = \{d_1, \ldots, d_M\}$ and N slots $\mathcal{S} = \{s_1, \ldots, s_N\}$ to track as examples illustrated in Figure 1. B_t, the dialogue state at turn t, is then defined as a mapping from a pair (d_m, s_n) into values v. Here, we define $B_t(d_m, s_n) = \phi$, if (d_m, s_n) is not in the current dialogue state. In the given example of Figure 1, the pair (domain=*hotel*, slot=*ref*) is not in the dialogue state, and the value "*none*" is assigned.

3.1 Generation-based DST with the Sequence-to-sequence Model

There are primarily two decoding strategies for generation-based DST in the literature for inferring the dialogue state at a particular turn – sequential (a)

and independent (b)(c) – both of which are explored in the paper as illustrated in Figure 1.

In the first case (top system (a) in Figure 1), the dialogue history C_t is taken as input to the encoder, and domain-slot-value triplets (d_m, s_n, v) are generated sequentially, where $B_t(d_m, s_n) \neq \phi$. This approach is adopted in many systems that leverage autoregressive LMs (Peng et al., 2020a; Hosseini-Asl et al., 2020). Despite being simple, this kind of sequential generation of multiple values is more likely to suffer from optimization issues with decoding long sequences resulting in lower performance. However, given its wide adoption in the literature, we still include this type of generative DST with the same backbone pretrained encoder-decoder Transformer model in our experiments. To partially address this issue, Lin et al. (2020b) propose a domain independent decoding where the decoder only have to generate a sequence of slot and value pairs within a specific given domain. Although their model leverages the same backbone model as ours, we empirically find that this form of strategy is still of limited effectiveness.

In the second case (middle two systems (b)(c) in Figure 1), the values for each domain-slot pair are generated independently, potentially in parallel. The domain and slot names (embedded as continuous representations) are either the initial hidden state of the decoder (Kim et al., 2020) or the first input of the decoder (Wu et al., 2019). Values are either generated for all possible domain-slot (d_m, s_n) pairs with a possible value of "*none*" and/or there is a separate gating mechanism for domain-slot combinations not currently active. Since we are interested in enriching the input with task-specific information, we focus on extending the independent decoding modeling for our prompt-based DST.

3.2 Prompt-based DST

In this section, we formally present the flow of our prompt-based DST with an encoder-decoder architecture. Here, we are interested in an encoder-decoder model with a bi-directional encoder (Raffel et al., 2020; Lewis et al., 2020), in contrast with the uni-directional encoder used in autoregressive LMs (Radford et al., 2019; Brown et al., 2020a).

The input of the prompt-based DST is made up of a dialogue context C_t and a task-specific prompt. Here, we use two types of task-specific prompts, the domain-related prompt $X(d_m)$, and slot-related prompt $X(s_n)$, both of which are derived based on the given schema. We leave the discussion of two specific realizations of task-specific prompts to the later part of this section. Specifically, all sub-sequences are concatenated with special segment tokens, i.e., "`[user]` U_1 `[system]` A_1 ... `[system]` A_{t-1} `[user]` U_t `[domain]` $X(d_m)$ `[slot]` $X(s_n)$", as input to the encoder, where `[user]`, `[system]`, `[domain]`, `[slot]` are special segment tokens for indicating the start of a specific user utterance, system utterance, domain-related prompt, and slot-related prompt, respectively.

Given this prompt-augmented input, the bi-directional encoder then outputs

$$H_t = \text{Encoder}(C_t, X(d_m), X(s_n)), \quad (1)$$

where $H_t \in \mathbb{R}^{L \times k}$ is the hidden states of the encoder, L is the input sequence length, and k is the encoder hidden size. Then, the decoder attends to the encoder hidden states and decodes the corresponding slot value $B_t(d_m, s_n)$:

$$B_t(d_m, s_n) = \text{Decoder}(H_t). \quad (2)$$

The overall learning objective of this generation processing is maximizing the log-likelihood of $B_t(d_m, s_n)$ given C_t, $X(d_m)$ and $X(s_n)$, that is

$$\sum_{(m,n)} \log P(B_t(d_m, s_n) | C_t, X(d_m), X(s_n)). \quad (3)$$

During inference, a greedy decoding procedure is directly applied, i.e., only the most likely token in the given model vocabulary is predicted at each decoding step.

Schema-Based Prompt. The first realization of task-specific prompt considered in this paper is based on the domain and slot names as defined in the task-dependent schema. As shown in (b) of Figure 1, given the domain name *train* and the slot name *day*, the specific prompt is in the form of "`[domain]` *train* `[slot]` *day*". Different from (Lin et al., 2020a; Wu et al., 2019) where the task-specific information is used in the decoder side, our symbol-based prompt as additional input to the bi-directional encoder can potentially achieve task-aware contextualizations. Observing that users often revise/repair their earlier requests in dialogues, we posit that the resulting encoded representations can be more effectively used by the decoder for generating corresponding slot values.

Natural Language Augmented Prompt. One main drawback of symbol-based prompt is that

Dataset	MWOZ 2.2	MWOZ 2.1	M2M
# Domains	8	8	2
# Dialogues	10438	10438	3008
# Total Turns	143004	143048	27120
Avg. Turns per Dial.	13.70	13.70	9.01
Avg. Toks per Turn	13.23	13.18	8.28
# Cat. Slots	21	0	0
# Non-Cat. Slots	40	37	12
Domain Desc.	Y	N	N
Slot Desc.	Y	Y	N
Value Set	Y	N	N

Table 1: Experiment data summary. The numbers are computed on all splits of the datasets. `MWOZ` stands for MultiWOZ. `Cat. Slots` and `Non-Cat. Slots` stand for categorical slots and non-categorical slots, respectively. The rows `Domain Desc.` and `Slot Desc.` indicate whether the corresponding dataset has natural language description for domains and slots, respectively. The row `Value Set` incates whether the corresponding dataset provides possible value set for categorical slots.

those domain/slot names contain limited information that can be utilized by pretrained LMs. In other words, those symbols from the custom schema are typically under-specified and unlikely to appear in corpus for LM pretraining. Fortunately, documentation is commonly available for real-world databases, and it is a rich resource for domain knowledge that allows dialogue systems to better understand the meanings of the abbreviated domain and slot names. The documentation includes but is not limited to domain/slot descriptions and the list of possible values for categorical slots. In this work, we experiment with a simple approach that augments the input by incorporating the domain description after the domain name and the slot description (with the sequence of values, if any) following the slot name, as illustrated in the system (c) in Figure 1.

3.3 Backbone Sequence-to-sequence Model

Our prompt-based DST model is initialized with weights from a pretrained LM in an encoder-decoder fashion. In this paper, we use the Text-to-Text Transformer (T5) (Raffel et al., 2020) as our backbone model. T5 is an encoder-decoder Transformer with relative position encodings (Shaw et al., 2018). We refer interested readers to the original paper for more details.

4 Experiments

4.1 Datasets

Table 1 summarizes the statistics of the datasets used in our experiments.

MultiWOZ (Budzianowski et al., 2018) is a multi-domain task-oriented dialogue dataset that contains over 10K dialogues across 8 domains. It is a collection of human-human written conversations and has been one of the most popular benchmarks in the DST literature. Since its initial release, many erroneous annotations and user utterances have been identified and fixed in subsequent versions, i.e., MultiWOZ 2.1 (Eric et al., 2019) and MultiWOZ 2.2 (Zang et al., 2020). In addition, MultiWOZ 2.1 provides 2-3 descriptions for every slot in the dataset. We randomly sample one of them and use the same descriptions for every experiment. The original dataset does not have domain descriptions and possible values so these are omitted in the corresponding experiments. MultiWOZ 2.2 further provides descriptions of domain and slot as well as possible values for categorical slots.

Machines Talking To Machines (M2M) (Shah et al., 2018) is a framework that combines simulation and online crowdsourcing. Templates of each dialogue are first generated and then online workers rewrite the conversations to make them human-readable while preserving the meaning. It provides 3,000 dialogues spanning 2 domains. The restaurant domain is denoted as `Sim-R` and the movie domain is denoted as `Sim-M`. Since there are no descriptions provided in the corpus, we take existing descriptions from other corpora that have the same slots. Specifically, descriptions for the restaurant domain are taken from MultiWOZ 2.2, whereas descriptions for the movie domain are taken from SGD (Rastogi et al., 2020b). All slots in M2M are covered. Since all slots are non-categorical, the descriptions do not include the possible values.

Evaluation Metric. The standard joint goal accuracy (JGA) is used as the evaluation metric. It treats a prediction as correct only if for every domain all slots exactly match the ground-truth values. For MultiWOZ 2.1 and 2.2, we use the official evaluation script from the DSTC8 challenge (Rastogi et al., 2020a).[1] For M2M, we adopt the above evaluation scripts with simple modifications.

[1] https://github.com/google-research/google-research/tree/master/schema_guided_dst#evaluation-on-multiwoz-21

Models	Pretrained-Model/ # Para.	JGA
TRADE	N	48.6
DS-DST	BERT-base / (110M)	51.7
Seq2Seq-DU	BERT-base / (110M)	54.4
Sequential	T5-small / (60M)	48.9
Sequential	T5-base / (220M)	51.2
Independent	T5-small / (60M)	55.2
w. desc	T5-small / (60M)	56.3
Independent	T5-base / (220M)	56.7
***w.* desc**	T5-base / (220M)	**57.6**

Table 2: Results on MultiWOZ 2.2. All numbers are reported in joint goal accuracy (JGA)(%). *w.* desc means the model is trained with the description. # Para. stands for the number of model parameters.

Models	Pretrained-Model / # Para.	JGA
TRADE	N	45.60
MinTL	T5-small / (60M)	50.95
MinTL	BART-large / (406M)	53.62
SST	N	55.23
TripPy	BERT-base / (110M)	55.29
Simple-TOD[2]	GPT2 / (117M)	55.72
*SOLOIST	GPT-2 / (117M)	56.85
***TripPy + SCORE**	ROBERTA-large / (355M)	**60.48**
Independent	T5-small / (60M)	55.37
w. desc	T5-small / (60M)	56.12
Independent	T5-base / (220M)	56.39
***w.* desc**	T5-base / (220M)	**56.66**

Table 3: Results on MultiWOZ 2.1. All numbers are reported in joint goal accuracy (JGA)(%). *w.* desc means the model is trained with the description. * means extra dialogue data is used to finetune the language model. # Para. stands for the number of model parameters.

4.2 MultiWOZ 2.2: Fully Annotated Natural Language Augmented Prompt

We present the evaluation results on MultiWOZ 2.2 in Table 2. The following baseline models are considered: TRADE (Wu et al., 2019), DS-DST (Zhang et al., 2019) and Seq2Seq-DU (Feng et al., 2020). Similar to ours, the decoding strategy of TRADE is independent. However, the sum of domain and slot embeddings are the first input of the decoder, which makes their dialogue history representation not task-aware contextualized. The sequential decoding strategy is worse than the independent decoding strategy by over 5% with both T5-small and T5-base. Even with T5-small (almost half the model size of BERT-base which is used in most previous benchmark models), our system achieves the SOTA performance using the independent decoding. As expected, T5-base systems outperform T5-small systems. With the augmentation of descriptions, we improve the overall JGA by over 1% in both T5-small and T5-base.

4.3 MultiWOZ 2.1: Partially Annotated Natural Language Augmented Prompt

Different from MultiWOZ 2.2 studied in the previous section, MultiWOZ 2.1 only contains natural language descriptions for slots but not domains. In addition, there is no possible slot value information.

The evaluation results on MultiWOZ 2.1 are shown in Table 3, where we compare with TRADE (Wu et al., 2019), MinTL (Lin et al., 2020a), SST (Chen et al., 2020), TripPy (Heck et al., 2020), Simple-TOD (Hosseini-Asl et al., 2020), SOLOIST (Peng et al., 2020a) and TripPy+SCORE (Yu et al., 2020). Note that both SOLOIST and TripPY+SCORE use external dialogue datasets to finetune their models.

As expected, we observe that T5-base models perform consistently better than T5-small models. Moreover, using descriptions consistently improves the performance of both models. All our models outperform baselines that do not use extra dialogue data. It is worth noting that comparing with MinTL (T5-small), our model is better by over 4% even without descriptions. Further, our T5-small system is even better than MinTL built on BART-LARGE (Lewis et al., 2020) which has substantially more parameters. Similar to ours, MinTL leverages a sequence-to-sequence LM. One difference is that their domain information is fed only to the decoder while our approaches enables task-aware contextualization by prompting the LMs with domain and slot information on the encoder side. Another difference is that they jointly learn DST together with dialogue response generation, which provides more supervision signals. Therefore, the better performance of our systems implies that schema-driven prompting is effective.

Lastly, compared with MultiWOZ 2.2, the performance gain brought by augmenting natural language descriptions is less pronounced which is likely caused by the reduced information available in MultiWOZ 2.1 descriptions.

4.4 M2M: Borrowed Natural Language Augmented Prompt

Table 4 shows the evaluation results on M2M. In this case, all natural language descriptions are directly borrowed from dialogue datasets that are an-

Models	Sim-M	Sim-R	Sim-M+R
(Rastogi et al., 2017)	96.8	94.4	–
(Rastogi et al., 2018)	50.4	87.1	73.8
(Chao and Lane, 2019)	80.1	89.6	–
(Heck et al., 2020)	**83.5**	90.0	–
Independent	83.3	89.6	**88.0**
w. desc	81.0	**90.6**	86.4

Table 4: Results on M2M. All numbers are reported in joint goal accuracy(JGA)(%). (Rastogi et al., 2017) should be considered as a kind of oracle upper bound performance because the target slot value is guaranteed to be in the candidate list and consider by the model.

notated in a different manner. We achieve the SOTA performance on Sim-R and Sim-M+R while being comparable on Sim-M. The improvements of descriptions are only evident on the restaurant domain. The lack of improvement from slot descriptions for the movie domain may be because the slot descriptions do not add much beyond the slot name (compared to "category" for the restaurant domain) or that it has slots that generalize better across domains (e.g. date, time, number of people).

5 Analysis

5.1 Breakdown Evaluation for MultiWOZ

In Table 5, we follow the categorization provided in (Zang et al., 2020) and show the breakdown evaluation of categorical and non-categorical slots on MultiWOZ 2.2. As we can see, the breakdown accuracy scores for both categorical and non-categorical slots are pretty consistent with the overall JGA. For both T5-small and T5-base models, models with sequential decoding perform worse than the corresponding models with independent decoding for both categorical and non-categorical slots. In particular, the independent decoding models achieve more pronounced improvement in categorical slots indicating that the task-specific prompt is very helpful for guiding the decoder to predict valid values. When comparing models using natural language description with those not, we observe performance gains for both types of slots for T-base but only non-categorical slots for T5-small. It is likely that the smaller size of T5 has limited representation capability to effectively utilize the additional textual description information regarding types and possible values.

Models	JGA	CAT	NON-CAT
Sequential (T5-small)	48.9	61.3	69.0
Sequential (T5-base)	51.2	62.9	70.9
Independent (T5-small)	55.2	71.4	75.2
w. desc	56.3	71.1	76.2
w. only slot desc	55.2	70.4	75.8
w. only domain desc	54.3	70.1	75.4
w. only slot + domain desc	55.9	71.2	76
Independent (T5-base)	56.7	71.6	76.3
w. desc	57.6	72.4	76.8

Table 5: Slot type breakdown results on the test set of MultiWOZ 2.2. All numbers are reported in joint goal accuracy(JGA) (%). CAT and NON-CAT correspond to categorical slots JGA and non-categorical slots JGA, respectively. *w.* desc indicates that the model is trained with the full description.

5.2 Ablation Study on Schema Descriptions

To understand what parts of the schema descriptions are most important, we experiment with three kinds of description combinations on MultiWOZ 2.2 using the T5-small configuration: **(i)** excludes the list of possible values for categorical slots **(ii)** excludes slot descriptions **(iii)** excludes domain descriptions. For **(i)**, there is an 0.4% point drop in JGA, validating that value sets can successfully constrain the model output, as we illustrate in Table 6. For **(ii)**, there is a 0.8% point drop in JGA. And for **(iii)**, there is a 0.1% point drop in JGA. This shows that slot descriptions are the most important part of the schema prompts and domain descriptions are relatively less effective. This is probably due to the fact that there are 61 slots in MultiWOZ 2.2 but only 8 domains. Also, the domain names are all self-contained single words.

5.3 The Effectiveness of Natural Language Augmented Prompt

In order to understand the benefit of natural language augmented prompt, we focus on analyzing the examples where the description augmented model correctly tracks the dialogue state while the unaugmented one fails. Based on our analysis of T5-base model on MultiWOZ 2.2, the most common errors are either misses of gold slots or over-predictions of irrelevant slots (82.8% of all errors). The remaining error cases are correct slot predictions with wrong slot values (17.2%).

We provide representative examples for which the description augmented system correctly tracks the dialogue states but not the unaugmented one in Table 6. In the first example, the phrases in

Database *Train*	**Slot Descriptions ‖ Possible Values**
arriveby	**arrival time of the train**
destination	destination of the train ‖ Birmingham New Street, **London Kings Cross**, ..., Stevenage
Dialogue History	... *[SYS]* The earliest being 19:09 and arriving by 20:54. Would that work for you? *[USR]* Yes, I think the **20:54 arrival time** should work.
no desc.	*(train, day, friday) (train, departure, leicester) (train, destination, cambridge) (train, leaveat, 19:00)*
desc.	***(train, arriveby, 20:54)*** *(train, day, friday) (train, departure, leicester) (train, destination, cambridge) (train, leaveat, 19:00)*
Dialogue History	*[USER]* I need to find a train going to Leicester that arrives by **4:45 PM**. Do you know of one?
no desc.	*(train, arriveby, 04:45) (train, destination, leicester)*
desc.	***(train, arriveby, 16:45)*** *(train, destination, leicester)*
Dialogue History	*[USER]* Can you help me find a train for Sunday. I would like to visit **London Kings Street**.
no desc.	*(train, destination, London Kings Street) (train, day, Sunday)*
desc.	***(train, destination, London Kings Cross)*** *(train, day, Sunday)*

Table 6: Examples for `train` domain dialogues where the description-augmented ("desc.") model make the correct state predictions but the unaugmented models ("no desc.") fails. The correctly predicted triplets are in bold.

	53.33%: Annotation Errors
Dialogue History	...*[SYSTEM]* Out of the 21 restaurant choices, one is the **Yippee Noodle Bar which is moderately priced in the centre of town**. Would you like to make a reservation? *[USER]* **That sounds great**, what is the postcode?
Gold	*()*
desc. Prediction	*(restaurant, area, centre) (restaurant, pricerange, moderate) (restaurant, name, yippee noodle bar*
	20.00%: Unable to Capture System Information
Dialogue History	... *[SYSTEM]* There is TR6679. **It leaves at 19:35 and arrives at 19:52**. Is that good for you? *[USER]* **Sounds good**. May I have the travel time and ticket price, please?
Gold	*(train, arriveby, 19:52) (train, leaveat, 19:35)*
desc. Prediction	*()*
	16.66%: Unable to Mention Slot Provided by User
Dialogue History	... *[USER]* Do you happen to know if there is a nightclub in the centre? *[SYSTEM]* Yes, we have FIVE nightclubs in the centre of town. Is there a particular one you're looking for? *[USER]* **I don't care which one you recommend**, but can you tell me the entrance fee and address?
Gold	*(attraction, area, centre) (attraction, type, nightclub)* **(attraction, name, dontcare)**
desc. Prediction	*(attraction, area, centre) (attraction, type, nightclub)*
	10.00%: Incorrect Value Reference
Dialogue History	*[USER]* Hi can you help me find a very nice Italian restaurant near the centre of cambridge? *[SYSTEM]* Please specify your price range. *[USER]* **It does not matter**.
Gold	*(restaurant, area, centre) (restaurant, food, italian)* ***(restaurant, pricerange, dontcare)***
desc. Prediction	*(restaurant, area, centre) (restaurant, food, italian)* ***(restaurant, pricerange, expensive)***

Table 7: The most common error types of our best model(t5-base *w/ desc.*) and corresponding examples.

the dialogue history are partially matched to the slot description of *arriveby* making it easier for the description-augmented system to detect the mention of the correct slot. For the second example, the type information in the description implicitly guides the model to focus on time-related information leading the correct output of the normalized time expression, 16:45. In contrast, the model without descriptions only generates the partial answer 4:45, ignoring PM. Lastly, "London Kings Street" is a typographical error in this case. By utilizing the provided possible values included in the slot descriptions, the model is able to generate the correct slot value without spelling error, demonstrating that the natural language augmented prompt can successfully constrain the model output and potentially provides robustness to the dialogue state tracking system.

5.4 Error Analysis of Natural Language Augmented Prompt-based DST

Here, we further carry out error analyses into the natural language augmented prompt-based T5-base

model on MultiWOZ 2.2. As shown in Table 7, we randomly sample 50 turns and categorize them into different types. In summary, there are four types of errors: **(i)** The most common error type is annotation error in which the model prediction is actually correct, which is similar to the findings of (Zhou and Small, 2019). **(ii)** 20% of the errors come from model failing to capture information provided by the system.[3] **(iii)** 16.66% of the errors are caused by the model misses of at least one gold slot. **(iv)** 10% of the errors are correct slot predictions with the wrong corresponding values. In general, most errors are likely caused by the lack of explicit modeling of user-system interactions.

6 Conclusion

In this work, we propose a simple but effective task-oriented dialogue system based on large-scale pretrained LM. We show that, by reformulating the dialogue state tracking task as prompting knowledge from LM, our model can benefit from the knowledge-rich sequence to sequence T5 model. Based on our experiments, the proposed natural language augmented prompt-based DST model achieve SOTA on MultiWOZ 2.2 and comparable performance on MultiWOZ 2.1 and M2M to recent SOTA models. Moreover, our analyses provide evidence that the natural language prompt is effectively utilized to constrain the model prediction.

Acknowledgements

This research was supported in part by a grant from Allstate. We would like to thank the reviewers for their constructive feedback.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

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.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Yue Feng, Yang Wang, and Hang Li. 2020. A sequence-to-sequence approach to dialogue state tracking. *arXiv preprint arXiv:2011.09553*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the*

[3]There is label inconsistency in the MultiWoZ as pointed out by (Zhou and Small, 2019). If the user confirms the booking or gives a positive response, then the dialogue states in the previous system utterance should be grounded. However, this rule is not always followed in the dataset construction. So to some extent, this type of error is inevitable.

Special Interest Group on Discourse and Dialogue, pages 35–44.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. Convert: Efficient and accurate conversational representations from transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2161–2174.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 263–272.

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

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.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.

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.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020a. MinTL: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020b. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020a. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020b. Few-shot natural language generation for task-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 172–182.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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*, 21:1–67.

Abhinav Rastogi, Raghav Gupta, and Dilek Hakkani-Tur. 2018. Multi-task learning for joint language understanding and dialogue state tracking. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 376–384.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020a. Schema-guided dialogue state tracking task at dstc8. *arXiv preprint arXiv:2002.01359*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020b. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Chien-Sheng Wu, Steven CH Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. *arXiv preprint arXiv:2101.09374*.

Tao Yu, Rui Zhang, Alex Polozov, Christopher Meek, and Ahmed Hassan Awadallah. 2020. Score: Pre-training for context representation in conversational semantic parsing. In *International Conference on Learning Representations*.

Xiaoxue Zang, Abhinav Rastogi, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B 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.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive dialogue state tracker. *arXiv preprint arXiv:1805.09655*.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.

Supplementary Material

A Implementation Details

The backbone models we use for finetuning are T5-small(60M parameters) and T5-base(220M parameters). We use the pretrained checkpoint from *transformers* library[4]. For T5-small, we train the model with a batch size 4, a learning rate of 5e-5 for 3 epochs. For T4-base, we train the model with a batch size of 64, a learning rate of 5e−4 for 2 epochs. Both models are trained using Adam(Loshchilov and Hutter, 2018). We don't use any text or label normalization scripts like (Wu et al., 2019; Hosseini-Asl et al., 2020).

For MultiWOZ 2.1 and 2.2, following many previous works(Wu et al., 2019), since *police* and *hospital* domains only appear in the training set, we exclude them in all our experiments.

B Descriptions

We show the descriptions of M2M and MultiWOZ 2.1in Table8 and Table9

[4] https://huggingface.co/t5-small ,https://huggingface.co/t5-base

Table 8: Domain and slot descriptions of M2M used in our experiments. The descriptions of the movie domain is taken from (Rastogi et al., 2020a) and the descriptions of the restaurant domain is taken from (Zang et al., 2020).

Sim-M			
Domain	**Domain Description**	**Slot**	**Slot Description**
Movie	A go-to provider for finding movies, searching for show times and booking tickets	theatre_name	the name of the theatre where the movie is playing
		movie	name of the movie
		date	date of the show booking
		time	time of the show booking
		num_people	number of people to purchase tickets for

Sim-R			
Domain	**Domain Description**	**Slot**	**Slot Description**
Restaurant	find places to dine and whet your appetite	price_range	price budget for the restaurant
		location	the location or area of the restaurant
		restaurant_name	the name of the restaurant
		category	the cuisine of the restaurant you are looking for
		num_people	how many people for the restaurant reservation
		date	date of the restaurant booking
		time	time of the restaurant booking

Table 9: The randomly sampled descriptions of MultiWOZ 2.1 used in all our experiments.

MultiWOZ 2.1		
Domain	**Slot**	**Slot Description**
taxi	leaveat	what time you want the taxi to leave your departure location by
taxi	destination	destination of taxi
taxi	departure	what place do you want to meet the taxi
taxi	arriveby	when you want the taxi to drop you off at your destination
restaurant	book people	number of people booking the restaurant
restaurant	book day	what day of the week to book the table at the restaurant
restaurant	book time	time of the restaurant booking
restaurant	food	food type for the restaurant
restaurant	pricerange	price budget for the restaurant
restaurant	name	name of the restaurant
restaurant	area	preferred location of restaurant
train	destination	destination of the train
train	day	what day you want to take the train
train	departure	departure location of the train
train	arriveby	what time you want the train to arrive at your destination station by
train	book people	number of people booking for train
train	leaveat	when you want to arrive at your destination by train
hotel	pricerange	preferred cost of the hotel
hotel	type	type of hotel building
hotel	parking	parking facility at the hotel
hotel	book stay	length of stay at the hotel
hotel	book day	day of the hotel booking
hotel	book people	how many people are staying at the hotel
hotel	area	rough location of the hotel
hotel	stars	rating of the hotel out of five stars
hotel	internet	whether the hotel has internet
hotel	name	which hotel are you looking for
attraction	type	type of attraction or point of interest
attraction	area	area or place of the attraction
attraction	name	name of the attraction