Description-Driven Task-Oriented Dialog Modeling

Anonymous ACL submission

Abstract

Task-oriented dialogue (TOD) systems are required to identify key information from conversations for the completion of given tasks. Such information is conventionally specified in terms of intents and slots contained in task-specific ontology or schemata. Since these schemata are designed by system developers, the naming convention for slots and intents is not uniform across tasks, and may not convey their semantics effectively. This can lead to models memorizing arbitrary patterns in data, resulting in suboptimal performance and generalization. In this paper, we propose that schemata should be modified by replacing names or notations entirely with natural language descriptions. We show that a language description-driven system exhibits better understanding of task specifications, higher performance on state tracking, improved data efficiency, and effective zero-shot transfer to unseen tasks. Following this paradigm, we present a simple yet effective **D**escription-**D**riven **D**ialog **S**tate **T**racking (D3ST) model, which relies purely on schema descriptions and an "index-picking" mechanism. We demonstrate the superiority in quality, data efficiency and robustness of our approach as measured on the MultiWOZ (Budzianowski et al., 2018), SGD (Rastogi et al., 2020), and the recent SGD-X (Lee et al., 2021b) benchmarks.

1 Introduction

The design of a task-oriented dialogue (TOD) system conventionally starts with defining a rigid schema specifying types of information that are most critical to the completion of a given task, often in the form of a list of slots and intents relevant to the task. A model can then be trained to identify the specified slots and intents accurately from conversations for user language understanding.

The format of schema elements can in principle be defined in arbitrary ways, but they often appear as abbreviated notations like `train-leaveat` and `hotel-internet` to indicate the task domain and required information. The building procedure of many TOD models are driven by such abbreviated or loosely defined notations. For example, decoder-only or sequence-to-sequence (seq2seq) TOD models (Hosseini-Asl et al., 2020; Zhao et al., 2021) are usually trained with supervision to predict dialogue state sequences like `train-leaveat=3:00pm` and `hotel-internet=no`. This conventional way of defining and using schema, however, has several disadvantages. First, the element notations convey little semantic (and possibly ambiguous) meaning for the requirements of the slot (Du et al., 2021), potentially harming language understanding. Second, task-specific abstract schema notations make it easy for a model to overfit on observed tasks and fail to transfer to unseen ones, even if there is sufficient semantic similarity between the two. Finally, creating notations for each slot and intent also complicates the schema design process.

In this paper, we advocate presenting schema with more natural, human-readable and semantically richer natural language descriptions, rather than abbreviated or even arbitrary ones. For example, instead of "`hotel-internet`", it is more natural to describe this slot as "`whether the hotel has internet`". This would be easier for both the designer of the TOD system when specifying the task ontology, and we also argue that it plays an important role in improving model quality and data efficiency. To this end, we propose a simple yet effective **D**escription-**D**riven **D**ialog **S**tate **T**racking (D3ST) approach based on the seq2seq architecture. In this approach, schema descriptions are indexed and concatenated as prefixes to a seq2seq model, which then learns to predict active schema element indices and corresponding values. An index-picking mechanism reduces the chance of the model overfitting to specific schema descriptions, and we demonstrate not only its superior

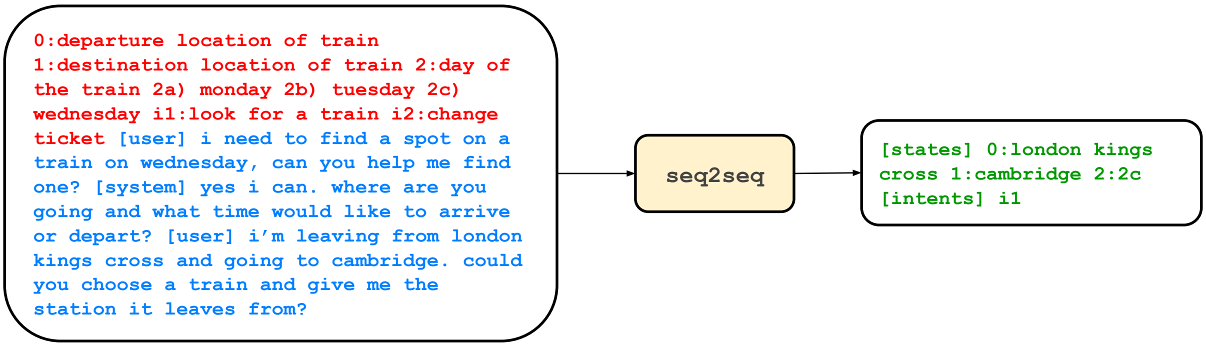

Figure 1: An example of D3ST. Red: Indexed schema description sequence as prefix; Blue: Conversation context; Green: State prediction sequence. See Section 3 for details. Best viewed in color.

performance as measured on benchmarks including MultiWOZ (Budzianowski et al., 2018; Zang et al., 2020; Han et al., 2021; Ye et al., 2021) and Schema-Guided Dialogue (SGD, (Rastogi et al., 2020)), but also strong zero- and few-shot transfer capability to unseen tasks.

There is prior work for leveraging language descriptions for better and more efficient dialogue. For example, the proposal of the SGD dataset (Rastogi et al., 2020) encourages adoption of language description for out-of-domain generalization, and (Lin et al., 2021b,a; Lee et al., 2021a; Mi et al., 2021) which takes advantage of descriptions or instructions as extra inputs to the model for improved model quality and sample efficiency. The differences and contributions from our work are summarized as follows:

1. We advocate creating schemata with detailed natural language descriptions for elements, doing away with abbreviated (or even arbitrary) schema element names. This paradigm not only simplifies schema design, but also improves model performance.

2. Based on the above, we propose an approach for dialogue state tracking via index selection, resulting in a state tracking model that requires a single forward pass for each turn to obtain the full dialogue state and leverages language descriptions in a simpler and more efficient manner than prior work.

3. We demonstrate superior performance on multiple benchmarks, as well as significant data efficiency improvement in zero-, few-shot, low-resource and cross-dataset settings.

4. We demonstrate its robustness to variations in language descriptions by evaluating on the SGD-X benchmark (Lee et al., 2021b), verifying that stronger language models lead to more robust task understanding.

2 Related Work

In recent years, there has been increasing interest in leveraging language prompts for data efficiency and quality improvement for dialogue modelling.

Inclusion of task descriptions: One line of research focuses on providing descriptions or instructions related to the dialogue tasks. Shah et al. (2019) utilized both slot descriptions and a small number of examples of slot values for learning slot representations for spoken language understanding. Similar to our work, Lin et al. (2021b); Lee et al. (2021a) provided slot descriptions as extra inputs to the model and have shown quality improvement as well as zero-shot transferability. Mi et al. (2021) extended the descriptions to a more detailed format by including task instructions, constraints and prompts altogether, demonstrating advantages of providing more sophisticated instructions to the model. However, unlike our approach, they predict slot values one-by-one in turn, which becomes increasingly inefficient as the number of slots increases, and is also prone to oversampling slot values since most slots are inactive at any stage during a dialogue. In contrast, our work predicts all states in a single pass, and is hence more efficient.

Prompting language models: Powerful language models like GPT (Radford et al., 2019; Brown et al., 2020) demonstrated impressive few-shot learning ability even without fine-tuning. It is therefore natural to consider leveraging these models for few-shot dialogue modeling. Madotto et al. (2020) applied GPT-2 by priming the model with examples for language understanding, state tracking, dialogue policy and language generation tasks respectively, and in Madotto et al. (2021) this approach has been extended to systematically evaluate on a set of diversified tasks using GPT-3 as backbone. Unlike these works in which the language models are frozen, we finetune the models on downstream tasks. Budzianowski and Vulić (2019); Baolin Peng (2020) on the other hand, applied GPT-2 for few-shot and transferable response generation with given actions, whereas our work focuses mainly on state tracking.

Describe task with questions: Another line of research casts state tracking as a question answering (QA) or machine reading (MR) problem (Gao et al., 2020; Namazifar et al., 2020; Li et al., 2021; Lin et al., 2021a), in which models are provided questions about each slot and their values are predicted as answers to these questions. The models are often finetuned on extractive QA or MR datasets, and by converting slot prediction into QA pairs the models are able to perform zero-shot state tracking on dialogue datasets. Their question generation procedure however, is more costly than using schema descriptions, which we adopt in our work.

3 Methodology

We make two design choices for our proposed approach: Use seq2seq model for state tracking, and use only descriptions of schema items to instruct the model.

3.1 Model

We choose to use seq2seq for modeling for the following reasons: first, seq2seq is a general and versatile architecture that can easily handle different formats of language instructions; second, seq2seq has been shown to be an effective approach for DST (Zhao et al., 2021); and third, seq2seq as a generic model architecture can be easily initialized from a pretrained checkpoint publicly available.

For our implementation and experiments, We use the T5 (Raffel et al., 2020) model and the associated pretrained checkpoints of different sizes.

3.2 Description-Driven Modeling

As discussed in Section 1, we aim to adopt a pure description-driven paradigm for dialogue modeling. For this purpose, we propose a simple approach that makes full use of schema descriptions with an "index-picking" mechanism, which we call **D**escription-**D**riven **D**ialog **S**tate **T**racking (D3ST). An example of D3ST is provided in Figure 1.

Given a set of descriptions corresponding to slots and intents specified by a schema, let $\mathtt{d}_\mathtt{i}^{\mathtt{slot}}, \mathtt{i} = 1 \ldots \mathtt{I}$ and $\mathtt{d}_\mathtt{j}^{\mathtt{int}}, \mathtt{j} = 1 \ldots \mathtt{J}$ be the descriptions for slots and intents respectively, where $\mathtt{I}$ and $\mathtt{J}$ are the numbers of slots and intents. Let $\mathtt{u}_\mathtt{t}^{\mathtt{usr}}$ and $\mathtt{u}_\mathtt{t}^{\mathtt{sys}}$ be the utterances by the user and system at turn $\mathtt{t}$ respectively.

Input The input to the encoder consists of a concatenation of two parts: $\mathtt{descriptions} + \mathtt{context}$. The $\mathtt{descriptions}$ part contains all descriptions from the schema arranged in the following format:

$$\mathtt{0} : \mathtt{d}_\mathtt{0}^{\mathtt{slot}} \ldots \mathtt{I} : \mathtt{d}_\mathtt{S}^{\mathtt{slot}} \quad \mathtt{i0} : \mathtt{d}_\mathtt{0}^{\mathtt{int}} \ldots \mathtt{iJ} : \mathtt{d}_\mathtt{J}^{\mathtt{int}}$$

Note that $\mathtt{0} \ldots \mathtt{I}$ and $\mathtt{i0} \ldots \mathtt{iJ}$ are the indices we assign to each of the slot and intent descriptions respectively. Here, "i" is a literal character to differentiate intent indices from those for slots. The $\mathtt{context}$ part consists of conversation history in the format of

$$[\mathtt{usr}]\ \mathtt{u}_\mathtt{0}^{\mathtt{usr}}\ [\mathtt{sys}]\ \mathtt{u}_\mathtt{0}^{\mathtt{sys}} \ldots [\mathtt{usr}]\ \mathtt{u}_\mathtt{T}^{\mathtt{usr}}\ [\mathtt{sys}]\ \mathtt{u}_\mathtt{T}^{\mathtt{sys}}$$

listing all utterances up to the current turn T. To prevent the model from memorizing association between a specific index:description pair, we randomize the assignment of indices to descriptions for each example during training. Such a dynamic construction forces the model to consider descriptions rather than treating inputs as constant strings to make generalizable predictions.

Output The decoder generates a sequence of dialogue states in the format

$$[\mathtt{states}]\ \mathtt{a}_\mathtt{0}^{\mathtt{s}} : \mathtt{v}_\mathtt{0}^{\mathtt{s}} \ldots \mathtt{a}_\mathtt{M}^{\mathtt{s}} : \mathtt{v}_\mathtt{M}^{\mathtt{s}}\ [\mathtt{intents}]\ \mathtt{a}_\mathtt{0}^{\mathtt{i}} \ldots \mathtt{a}_\mathtt{N}^{\mathtt{i}}$$

where $\mathtt{a}_\mathtt{m}^{\mathtt{s}}$ is the index of the $\mathtt{m}^{th}$ *active* slot and there are $\mathtt{M}$ active slots in all, $\mathtt{v}_\mathtt{m}^{\mathtt{s}}$ is its corresponding value. $\mathtt{a}_\mathtt{n}^{\mathtt{i}}$ is the index of the $\mathtt{n}^{th}$ active intent and $\mathtt{N}$ is the number of active intents. This way the model learns to identify active schema elements with abstract indices, as we randomize the element order during training. Note that inactive elements are not generated.

Handling categorical slots Some slots are categorical, that is, they have pre-defined candidate values for the model to choose from. For example "`whether the hotel provides free wifi or not`" could have the categorical values "`yes`" and "`no`". To improve categorical slot prediction accuracy, we enumerate possible values together with their slot descriptions. That is, assuming the $\mathtt{i}^{th}$ slot is categorical and has $\mathtt{k}$ values $\mathtt{v_a} \ldots \mathtt{v_k}$, its corresponding input format is

$$\mathtt{i} : \mathtt{d_i^{slot}}\ \mathtt{ia)}\ \mathtt{v_a} \ldots \mathtt{ik)}\ \mathtt{v_k}$$

in which $\mathtt{ia)} \ldots \mathtt{ik)}$ are indices assigned to each of the values.[1] Assuming this slot is active with its third value ($\mathtt{v_c}$) being mentioned, then the corresponding prediction has the format $\mathtt{i} : \mathtt{ic)}$.

3.3 Properties

From the formulation described in Section 3.2, we expect our proposed approach to have the following properties. First, the model relies fully on the understanding of schema descriptions for the identification of active slots and intents. Second, the model learns to pick indices corresponding to the active slots, intents or categorical values, instead of generating these schema elements. This "index-picking" mechanism, based on schema description understanding, reduces the chance of the model memorizing training schemata and makes it easier for the model to zero-shot transfer to unseen tasks. Finally, unlike previous work which also takes advantage of schema descriptions (for example Lin et al., 2021b; Lee et al., 2021a) but generates values for each slot in turn (even if a slot is inactive), our approach enables predicting multiple active (and only active) slot-value pairs together with intents with a single decoding pass, making the inference procedure more efficient.

We also note that the sequence of schema descriptions prepended to the conversation context plays a similar role as instructions for specific tasks (Wei et al., 2021; Mishra et al., 2021). Providing more detailed human-readable descriptions enables the language model understand task requirements better, and leads to improved few-shot performance, as will be seen in experimental results.

4 Experiments

We design our experiments to answer the following questions:

1. What is the quality of the D3ST model, when all training data is available?
2. How does the description type for schema definition, including human-readable natural descriptions, abbreviated or even random notations, affect model quality?
3. How data-efficient is D3ST in the low-resource or zero-shot regimes, and how do different description types affect efficiency?
4. How robust is the model to different wordings of the human-readable descriptions?

4.1 Setups

Datasets We conduct experiments on the MultiWOZ 2.1-2.4 (Budzianowski et al., 2018; Zang et al., 2020; Han et al., 2021; Ye et al., 2021) and SGD (Rastogi et al., 2020) datasets. The MultiWOZ dataset is known to contain annotation errors in multiple places and previous work adopted different data pre-processing procedures, so we follow the recommended procedure[2] of using the TRADE (Wu et al., 2019) script to pre-process MultiWOZ 2.1, but do not apply any pre-processing to 2.2-2.4 for reproducibility and fair comparison with existing results. We use Joint-Goal-Accuracy (JGA) as evaluation metric, which measures the percentage of turns for which all states are correctly predicted by the model.

Training setup We use the open-source T5 code base[3] and the associated T5 1.1 checkpoints.[4] We consider models of the size base (250M parameters), large (800M) and xxl (11B) initialized from the corresponding pretrained checkpoints, and ran each experiment on 64 TPU v3 chips (Jouppi et al., 2017). For fine-tuning, we use batch size 32 and use constant learning rate of $1e-4$ across all experiments.

We use the slot and intent descriptions included in the original MultiWOZ and SGD datasets as inputs ($\mathtt{d_i^{slot}}$ and $\mathtt{d_i^{int}}$ described in Section 3.2) to the model. For MultiWOZ, we include schema descriptions across all domains as model prefix and set the input length limit to 2048. To avoid ambiguity between descriptions from different domains, we also add domain names as part of the descriptions. For example for the `hotel-parking` slot, the description is "`hotel-parking facility at`

[1] One may also adopt $\mathtt{a)} \ldots \mathtt{k)}$ as value indices or even completely discard indexing for categorical values, however we found this shared indexing across categorical slots can sometimes cause selection ambiguity when some values (like "`true`" or "`false`") are shared by multiple categorical slots. We therefore apply slot-specific indices $\mathtt{ia)} \ldots \mathtt{ik)}$ to constrain index-picking within the $\mathtt{i}^{th}$ slot value range.

[2] https://github.com/budzianowski/multiwoz#dialog-state-tracking

[3] https://github.com/google-research/text-to-text-transfer-transformer

[4] https://github.com/google-research/text-to-text-transfer-transformer/blob/main/released_checkpoints.md

the hotel". For SGD, we include descriptions from domains relevant to each turn as suggested by the standard evaluation, and the input length limit is set to 1024. The output length is 512 in all cases.

4.2 Main Results

Our first experiment examines the model quality when the entire training datasets are used for fine-tuning. For MultiWOZ, we compare results with existing methods: TRADE (Wu et al., 2019), SUMBT (Lee et al., 2019), DS-DST (Zhang et al., 2020), Seq2Seq-DU (Feng et al., 2021), SOM-DST (Kim et al., 2020), Transformer-DST (Zeng and Nie, 2021), TripPy (Heck et al., 2020), SAVN (Wang et al., 2020), SimpleTOD (Hosseini-Asl et al., 2020), Seq2seq (Zhao et al., 2021), and DST-as-Prompting (DaP, Lee et al. (2021a)). For DaP, we consider two variations of the approach, namely sequential prediction (seq) and independent prediction (ind), described in their paper.

For SGD, we compare with the SGD baseline (Rastogi et al., 2020), SGP-DST (Ruan et al., 2020), paDST (Ma et al., 2020), DaP, as well as Team14[5] from the DSTC8 challenge (Kim et al., 2019) .

The results are given in Table 1, which show that D3ST is close to, or at the state-of-the-art across all benchmarks, illustrating the effectiveness of the proposed approach. We also see that increasing the model size significantly improves the quality.

Note however that not all results are directly comparable, and we discuss some notable incongruities. The best result on SGD is given by paDST, which uses both a data augmentation procedure by back-translating between English and Chinese, as well as special handcrafted rules for model predictions. In contrast, our models only train on the default SGD dataset, and do not apply any handcrafted rules whatsoever. While paDST has significantly higher JGA compared to D3ST base, our xxl model is only marginally worse. On the other hand, DaP also relies on slot descriptions and is finetuned from a T5 base model, making it directly comparable to our D3ST base model and we observe better performance on SGD and MultiWOZ. One additional advantage of D3ST is that it predicts all slots at once in a single inference pass. In contrast, the independent (ind) decoding variant of DaP does inference once for every slot, similar to most other baselines, and is thus far less efficient.

[5]We are not aware of any publicly available implementation for the methodology used by Team14.

Model	2.1	2.2	2.3	2.4
TRADE	45.6	45.4	49.2	55.1
SUMBT	49.2	49.7	52.9	61.9
DS-DST	51.2	51.7	-	-
Seq2Seq-DU	-	54.4	-	-
Transformer-DST	55.35	-	-	-
SOM-DST	51.2	-	55.5	66.8
TripPy	55.3	-	**63.0**	59.6
SAVN	54.5	-	58.0	60.1
SimpleTOD★	50.3/55.7	-	51.3	-
Seq2seq✦	52.8	57.6	59.3	67.1
DaP (seq)	-	51.2	-	-
DaP (ind)	56.7	57.6	-	-
D3ST (base)	54.2	56.1	59.1	72.1
D3ST (large)	54.5	54.2	58.6	70.8
D3ST (xxl)	**57.8**	**58.7**	60.8	**75.9**

(a) JGA on MultiWOZ 2.1-2.4.

Model	JGA	Intent	Req slot
SGD baseline	25.4	90.6	96.5
DaP (ind)	71.8	90.2	97.8
SGP-DST	72.2	91.8	99.0
Team14▲	77.3	96.9	**99.5**
paDST■	**86.5**	94.8	98.5
D3ST (base)	72.9	97.2	98.9
D3ST (large)	80.0	97.1	99.1
D3ST (xxl)	86.4	**98.8**	99.4

(b) JGA, active intent accuracy and requested slot F1 on SGD.

Table 1: Results on MultiWOZ and SGD datasets with full training data. ★: SimpleTOD results are retrieved from the 2.3 website `https://github.com/lexmen318/MultiWOZ-coref`, in which two numbers are reported for 2.1 (one produced by the 2.3 author, the other by the original SimpleTOD paper). ✦: No data pre-processing applied for MultiWOZ 2.1. ▲: No publication for the methodology or open-source codes available. ■: Data augmentation and special rules applied. "-" indicates no public number is available. Best results are marked in bold.

This is not salable and not consistent with current trend in TOD with more domains and slots available. DaP also has a sequential (seq) variant that also predicts all slots at once, but performs worse on JGA.

4.3 Comparison of Description Types

We now study whether the quality of D3ST is sensitive to the schema description types. For this, we run the same experiment as in Section 4.2 with D3ST large and xxl, but using three different types of descriptions: human-readable language descriptions, schema element names (abbreviations) as defined in the original schema, and random strings. The random string descriptions are generated by simply randomly permuting the character sequences of the original element names. This experiment is designed to check how a model with only memorization capability without any understanding of schema element semantics does on seen and unseen schemas. An example of all three description type comparisons can be found in Appendix A.

Type	M2.1	M2.2	M2.3	M2.4	SGD
Language	54.5	55.9	58.6	70.8	80.0
	57.8	58.7	60.8	75.9	86.4
Name	55.1	55.8	59.6	72.2	73.7
	57.5	57.9	60.4	75.4	79.7
Random	20.1	9.0	12.1	16.9	37.4
	57.6	56.1	59.3	73.6	64.8

Table 2: Comparison between D3ST models using different types of descriptions on MultiWOZ and SGD. “Language”, “Name” and “Random” correspond to using detailed language description, schema element name and random strings respectively. Each type contains two rows, corresponding to the results given by “large” and “xxl” models. Note that the "Random" experiments for "large" models had trouble converging, and we instead report their JGA at 85k steps.

Table 2 compares the performance with different description types. It can be seen that using language descriptions consistently outperforms other types, aligned with our expectation that natural and human-readable descriptions contain richer semantics and are aligned with the pretraining objective, enabling LM to perform better. Element names are less readable than full descriptions, but still retain some semantics: they preform well but fall short of full descriptions. On the other hand, using random strings performs worst on average, even on MultiWOZ where the training and test schema are the same (and the model is allowed to memorize descriptions from training). With random strings, there is the extra challenge of identifying the correct slot id for each value to predict, since each example has a random shuffling of the slot ids. Indeed, we observed that training "large" models on random names is hard to converge, and instead of reporting their final results, we stopped these experiments early and reported their JGA at 85k steps. The xxl models did not encounter the same issue; we suspect that it was easier for larger models to memorize slot name permutations.

In constrast to MultiWOZ, SGD requires models to generalize to unseen tasks and domains in the evaluation datasets. Here, using random strings undermined quality significantly. In general, meaningless inputs hurt performance and lead to less generalization. We therefore suggest instructing the model with semantically rich representations, in particular, language descriptions.

One more observation we make is that, on large MultiWOZ models, using element names had better JGA than using a full language description. This trend does not hold on SGD, and also reverses when trained with xxl. We hypothesize that this is a result of input sequence length: on MultiWOZ we feed slots descriptions from all domains as prefix, and when full language description is utilized, the input sequence becomes excessively long. Using element names shortens the length, making a moderate-size model easier to learn. In contrast, input sequence lengths on SGD are lower than that on MultiWOZ, since only active domains are provided as part of the input.

4.4 Data Efficiency

Properly designed prefixes or prompts have been shown to significantly improve an LM’s data efficiency (Radford et al., 2019; Liu et al., 2021; Wei et al., 2021). We investigate how different types of description prefixes vary in performance in low-resource regimes by running experiments with large and xxl models on SGD with 0.16% (10-shot), 1%, and 10% of training data. For the 0.16% experiment, we randomly select 10 samples from each training domain to increase the domain diversity, totalling 260 examples. For other experiments the samples are uniformly sampled across the entire training set. We sample from three random seeds for each experiment.

Type	0.18%	1%	10%
Language	6.1 ± 0.7	36.7 ± 2.0	73.1 ± 0.2
	51.0 ± 0.2	79.4 ± 0.4	83.0 ± 0.1
Name	5.0 ± 0.2	28.0 ± 2.7	69.7 ± 0.3
	47.7 ± 0.5	74.9 ± 1.4	78.6 ± 0.7

Table 3: Data efficiency of D3ST using natural language and element name descriptions, trained and evaluated on SGD. Each description type contains two rows, corresponding to the results given by “large” and “xxl” models. The metric is JGA.

The results are given in Table 3. From the table we have the following observations:

- Using human-readable language descriptions consistently outperforms other types of representations, indicating better data efficiency with semantically-rich descriptions.

- With just 0.18% of the data, xxl models can already reach more than half of their full quality (from Table 1). At 1%, we observe quality close to using 100% data. Increasing to 10% only yielded marginal gains.

- Larger models are much more data efficient than smaller ones, as can be seen from the big gap between "large" and "xxl" models.

4.5 Zero-shot Transfer to Unseen Tasks

To assess our approach's zero-shot transfer ability to unseen tasks, we conduct the following set of experiments:

MultiWOZ cross-domain transfer Following a setup similar to TransferQA (Lin et al., 2021a) and T5DST (Lin et al., 2021b), we run the "leave-one-out" cross-domain zero-shot transfer evaluation on MultiWOZ 2.1.[6] For each domain, we train a model on examples excluding that domain, and evaluate it on examples including it. Table 4a shows our results in comparison with the baselines.[7] It can be seen that our approach achieves the best cross-domain transfer performance with significant gains across almost all domains.

SGD unseen service transfer The SGD benchmark contains numerous services and some domains only present in the test set. We present the results for zero-shot transfer to these domains and services in Table 4b. Note that D3ST base has worse JGA on unseen domains when fairly compared to DaP and SGP-DST. However, D3ST has superlative JGA on seen domains, even better than paDST (with data augmentation and hand-crafted rules). In addition, increasing the size of D3ST further increases both seen and especially unseen JGA, indicating better generalization. At xxl, JGA on unseen domains is almost equal to paDST.

Cross-dataset transfer In this setup, we evaluate if a model trained on one dataset can be directly applied to another dataset. To this end, we train a model on SGD then directly evaluate on the MultiWOZ 2.4 test set, and vice versa[8]. In both cases we use the xxl model from Section 4.2, and report the numbers in Table 4.

Despite obvious schema differences and domain mismatch between MultiWOZ and SGD, our model trained on MultiWOZ already achieves zero-shot quality on SGD close to the BERT-baseline (Rastogi et al., 2020) with 25.4% JGA. Our model trained on SGD and evaluated on MultiWOZ shows similarly strong zero-shot results. Both results are much lower than the state of the art for both datasets however, due to differing biases defined in schemata between the two datasets, and from latent knowledge that isn't captured from a schema alone.

Domain	JGA		
	D3ST	TransferQA	T5DST
Attraction	**56.4**	31.3	33.1
Hotel	21.8	**22.7**	21.2
Restaurant	**38.2**	26.3	21.7
Taxi	**78.4**	61.9	64.6
Train	**38.7**	36.7	35.4
Avg	**46.7**	35.8	35.2

(a) Cross-domain (leave-one-out) transfer on MultiWOZ.

Model	**JGA**		
	Overall	Seen	Unseen
SGD Baseline	25.4	41.2	20.0
DaP (ind)	71.8	83.3	68.0
SGP-DST	72.2	87.9	66.9
Team14▲	77.3	90.0	73.0
paDST■	**86.5**	92.4	**84.6**
D3ST (base)	72.9	92.5	66.4
D3ST (large)	80.0	93.8	75.4
D3ST (xxl)	86.4	**95.8**	83.3

(b) JGA on seen versus unseen services for SGD. ▲ and ■ have the same meaning as in Table 1.

Transfer	**JGA**
SGD→MultiWOZ	28.9
MultiWOZ→SGD	23.1

(c) Cross-dataset transfer b/w SGD and MultiWOZ 2.4.

Table 4: Zero-shot transfer evaluation results from three different setups.

Qualitative Evaluation In addition to quantitatively evaluating zero-shot transfer, we qualitatively examined examples of D3ST transferring to novel domains. We handcrafted a few dialogues for domains very different from the ones seen in the SGD dataset (e.g. conference submission, internet provider, e-commerce retailer). We designed the dialogues to be as stylistically realistic as pos-

[6]For zero-shot evaluation, Lin et al. (2021a) and Lin et al. (2021b) experimented on MultiWOZ 2.1 and 2.0 respectively. While our models are trained and evaluated on MultiWOZ 2.1, we include results from both of them for comparison.

[7]When skipping the `train` domain, we postprocess predictions for slots `train-departure` and `train-destination` by ignoring the suffix "train station". This is semantically correct and improves JGA.

[8]Note that the SGD dataset defines the services that will occur in each dialogue, whereas MultiWOZ expects models to be able to predict any of its domains for all dialogues. To make it compatible between SGD and MultiWOZ for cross-task zero-shot transfer, we limit the schema prefix for MutliWOZ to domains that appear in the current dialogue.

sible for customer service scenarios. We tasked the xxl model trained on SGD (from Table 1) with inferring their dialogue states, and share one example in Table 5. More examples can be found in Table A2 of Appendix B. We observe that the model performs surprisingly well across all of our handcrafted dialogues, even though the domains are very different from the training data.

Domain: Conference Submission

Input: `0:name of the conference 1:title of the paper 2:the first author of the paper 3:research areas for the paper 4:email for openreview account i1:submit a paper to a conference i2:check if a paper has been accepted [user] hi, i'd like to submit a paper for a conference [system] that's great. which conference would you like to submit to? [user] i'd like to submit to` **`acl 2022`** `[system] ok. could you share the title of your paper and the name of your first author? [user] the paper is` **`"description-driven task-oriented dialog modeling"`**`, and the first author is` **`grace hopper`** `[system] great, thank you. note that this year, we require all paper authors to be registered on openreview. could you give the email for your openreview account? [user] sure, its` **`gracehopper@gmail.com`**

Prediction: `[states] 0:acl 2022 1:description-driven task-oriented dialog modeling 2:grace hopper 4:gracehopper@gmail.com [intents] i1`

Table 5: An example of D3ST performing zero-shot transfer to a hypothetical "Conference Submission" domain. The predicted dialogue state is entirely correct. Boldface and color were added for visual clarity.

4.6 Robustness to Variations of Descriptions

Since there are many ways to provide descriptions for a given schema, a natural question to raise about this approach is how robust the model is against different choices of descriptions. The recently proposed SGD-X benchmark (Lee et al., 2021b) is designed specifically for the study of this problem. SGD-X contains five variations of the original SGD, each one using a different set of schema descriptions provided by different crowd-source workers. To assess the robustness of D3ST, we use the large and xxl models evaluated in Section 4.2 and decode test sets from each of the five variants of SGD-X. A robust model is expected to have smaller fluctuations in predictions across schema variants for the same dialogue context, as measured by Schema Sensitivity `SS(JGA)` defined in Lee et al. (2021b),. which calculates the average variation coefficient of JGA at turn level. A lower `SS(JGA)` value implies less fluctuation and more robustness.

We compare the robustness of models using different prompt types in Table 6. From the numbers we see that using the most human-readable natural language descriptions not only achieves the highest average accuracy over all SGD-X test set variants, but also enjoys the smallest `SS(JGA)` at the same model size. This indicates that description-driven models are more robust. On the other hand, using element names and random names have progressively lower mean accuracy and higher sensitivity to schema changes.

Size	Orig	v1	v2	v3	v4	v5	Avg v1-5	SS(JGA)
large	80.0	79.9	79.4	76.5	71.9	69.1	75.3	0.26
xxl	86.4	85.5	85.1	73.9	75.5	68.9	77.8	0.27

(a) Natural language description

Size	Orig	v1	v2	v3	v4	v5	Avg v1-5	SS(JGA)
large	73.7	72	69.5	66.4	61.1	65.7	66.9	0.37
xxl	79.7	80.8	76.6	74.2	61.2	72.3	73.0	0.35

(b) Element name description

Size	Orig	v1	v2	v3	v4	v5	Avg v1-5	SS(JGA)
large	37.4	29.3	34.6	28.0	25.2	25.0	28.4	0.74
xxl	64.8	67.8	68.8	72.9	58.1	68.1	67.1	0.51

(c) Random description

Table 6: Robustness comparison for various description types. SS(JGA) refers to schema sensitivity for JGA.

5 Conclusion

We advocate using human-readable language descriptions in place of abbreviated or arbitrary notations for schema definition in TOD modeling. We believe this schema representation contains more meaningful information for a strong LM to leverage, leading to better performance and improved data efficiency. To this end, we propose a simple and effective DST model named "Description-Driven Dialogue State Tracking" (D3ST), which relies fully on schema descriptions and an index-picking mechanism to indicate active slots or intents. Our experiments verify the effectiveness of description-driven dialogue modeling in the following ways. First, D3ST achieves superior quality on MultiWOZ and SGD. Second, using language descriptions outperforms abbreviations or arbitrary notations. Third, the description driven approach improves data-efficiency, and enables effective zero-shot transfer to unseen tasks and domains. Fourth, using language for schema description improves model robustness as measured by the SGD-X benchmark.

6 Ethical Considerations

We proposed a more efficient way of building TOD systems by leveraging language descriptions. Our intended use cases include developing automated conversational agents for customer service centers, hotel and ticket booking systems, etc. Our experiments are conducted on publicly available task-oriented conversation datasets in English, covering common domains like restaurant reservation, movie tickets, hotel reservation etc. We hope our work contributes to improving TOD system language understanding quality while reducing reliance on large amounts of annotated data.

References

Chunyuan Li Xiujun Li Jinchao Li Michael Zeng Jianfeng Gao Baolin Peng, Chenguang Zhu. 2020. Few-shot natural language generation for task-oriented dialog.

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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Xinya Du, Luheng He, Qi Li, Dian Yu, Panupong Pasupat, and Yuan Zhang. 2021. QA-driven zero-shot slot filling with weak supervision pretraining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 654–664, Online. Association for Computational Linguistics.

Yue Feng, Yang Wang, and Hang Li. 2021. A sequence-to-sequence approach to dialogue state tracking. 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 1714–1725, Online. Association for Computational Linguistics.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89, Online. Association for Computational Linguistics.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation.

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 Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

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

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, and Pierre-luc et al. Cantin. 2017. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Comput. Archit. News*, 45(2):1–12.

Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge.

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, Online. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021a. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2021b. Sgd-x: A benchmark for robust generalization in schema-guided dialogue systems.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074, Online. Association for Computational Linguistics.

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

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul 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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.

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

Yue Ma, Zengfeng Zeng, Dawei Zhu, Xuan Li, Yiying Yang, Xiaoyuan Yao, Kaijie Zhou, and Jianping Shen. 2020. An end-to-end dialogue state tracking system with machine reading comprehension and wide deep classification.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems.

Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems.

Fei Mi, Yitong Li, Yasheng Wang, Xin Jiang, and Qun Liu. 2021. Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions.

Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tür. 2020. Language model is all you need: Natural language understanding as question answering.

Alec Radford, Jeff 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(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Yu-Ping Ruan, Zhen-Hua Ling, Jia-Chen Gu, and Quan Liu. 2020. Fine-tuning bert for schema-guided zero-shot dialogue state tracking.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490, Florence, Italy. Association for Computational Linguistics.

Yexiang Wang, Yi Guo, and Siqi Zhu. 2020. Slot attention with value normalization for multi-domain dialogue state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3019–3028, Online. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners.

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, Florence, Italy. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, 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, Online. Association for Computational Linguistics.

Yan Zeng and Jian-Yun Nie. 2021. Jointly optimizing state operation prediction and value generation for dialogue state tracking.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective sequence-to-sequence dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Example of Description Types

An example of the different description types for a single example can be found in Table A1.

B Zero-shot Transfer to Novel Domains

Qualitative examples showcasing zero-shot transfer to novel domains can be found in Table A2.

Language	0:playback device on which the song is to be played 0a) bedroom speaker 0b) tv 0c) kitchen speaker 1=name of the artist the song is performed by 2=name of the song 3=album the song belongs to 4=genre of the song i0=search for a song based on the name and optionally other attributes i1=play a song by its name and optionally artist [user] i want to find a movie. [system] what is your location. [user] santa rosa. i want to see it at 3rd street cinema. [system] i found 3 movies. does hellboy, how to train your dragon: the hidden world or the upside interest you? [user] how to train your dragon: the hidden world is perfect. can you find me some songs from the album summer anthems. [system] i found 1 song you may like. how about no other love from the album summer anthems by common kings? [user] that would be great. [system] play the song now? [user] play it on the bedroom device.
Name	0:music_2-genre 1:music_2-playback_device 1a) bedroom speaker 1b) kitchen speaker 1c) tv 2:music_2-album 3:music_2-artist 4:music_2-song_name i0:music_2-playmedia i1:music_2-lookupmusic [user] i want to find a movie. [system] what is your location. [user] santa rosa. i want to see it at 3rd street cinema. [system] i found 3 movies. does hellboy, how to train your dragon: the hidden world or the upside interest you? [user] how to train your dragon: the hidden world is perfect. can you find me some songs from the album summer anthems. [system] i found 1 song you may like. how about no other love from the album summer anthems by common kings? [user] that would be great. [system] play the song now? [user] play it on the bedroom device.
Random	0:e-e_ciugs2mrn 1:psuekc_l-2imceyibaca_dv 1a) bedroom speaker 1b) kitchen speaker 1c) tv 2:umm2uisc_bal- 3:satriti_2-sumc 4:_-onassng2_cemmui i0:aeusmmci2-adipl_y i1:miiu_2olosckucp-ums [user] i want to find a movie. [system] what is your location. [user] santa rosa. i want to see it at 3rd street cinema. [system] i found 3 movies. does hellboy, how to train your dragon: the hidden world or the upside interest you? [user] how to train your dragon: the hidden world is perfect. can you find me some songs from the album summer anthems. [system] i found 1 song you may like. how about no other love from the album summer anthems by common kings? [user] that would be great. [system] play the song now? [user] play it on the bedroom device.
States	[states] 1:1a 2:summer anthems 4:no other love [intents] i0

Table A1: Examples of the same SGD dialogue with different description types. "Language" uses a detailed natural language description, "Name" uses the schema element name, and "Random" is generated from a random shuffling of the slot name. Note that the categorical slot value enumeration is unaffected in "Random", and that all three description types would have the same target slots and intents.

Domain	Internet Provider
Inputs	0:email address of the account 1:whether professional help is needed for internet installation 1a) true 1b) false 2:whether to bundle services on the same plan 2a) true 2b) false 3:download speed of the internet plan 4:whether services are for residential or business use 4a) residential 4b) business 5:the address to provide services to i0=buy or change an internet plan i1:file a formal complaint [user] hi there - my internet contract is up for renewal, and i'm interested in exploring other plan options. [system] happy to help. is this for your home or for a business? [user] home [system] what's the email associated with your account? [user] noamchomsky@hotmail.com [system] thanks. your current plan is 25 mbps download speed for $53 / month. the two other plans are 50 mbps for $63 / month and 100 mbps for $73 / month. would you interested in either of those? [user] i'm interested in upgrading to the 50 mbps plan. [system] great. for $10 / month more, would you like to include our basic cable plan? [user] no thanks. i'll need to talk this over with my partner. thank you for your help.
States	[states] 0:noamchomsky@hotmail.com 3:50 mbps 4:4a [intents] i0
Domain	E-Commerce Retailer
Inputs	0:phone number associated with the customer's account 1:a coupon code to apply to the purchase 2:the reason for the product return 2a) accidental purchase 2b) malfunction 2c) preference 3:the retail product to purchase or to be returned 4:date the product was purchased 5:identifier associated with the purchase i0:return a product i1:purchase a product [system] hi how can i help you today? [user] hello - i recently purchased a glow in the dark ball that i'd like to return. [system] no problem. i'm happy to help. can you provide the order number or date of purchase please? [user] 1ozdl3v260lkq, and i purchased it last week on nov 1, 2021 [system] thanks. and what's the reason for the return? [user] the ball seems to be broken. it doesn't actually glow in the dark. [system] sorry to hear about that. we'll process the return and you should receive a refund within 10 business days. is there anything else i can do for you? [user] no, thanks for your help!
States	[states] 2:2b 3:glow in the dark ball 4:nov 1, 2021 5:1ozdl3v260lkq [intents] i0

Table A2: Two more examples of D3ST trained on SGD performing zero-shot transfer to novel domains. The only error is in the "Internet Provider" example, where the model misses that the slot for "whether to bundle services on the same plan" should be false. We hypothesize that "bundle" is industry jargon that the model fails to associate with the dialogue context.