State Value Generation with Prompt Learning and Self-Training for Low-Resource Dialogue State Tracking

Ming Gu   51215901012@STU.ECNU.EDU.CN
School of Computer Science and Technology, East China Normal University

Yan Yang[*]   YANYANG@CS.ECNU.EDU.CN
School of Computer Science and Technology, East China Normal University

Chengcai Chen   ARLENECC@XIAOI.COM
Xiaoi Research, Xiaoi Robot Technology Co., Ltd

Zhou Yu   ZY2461@COLUMBIA.EDU
Dialogue NLP Lab, Columbia University

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Recently, low-resource dialogue state tracking (DST) has received increasing attention. First obtaining state values then based on values to generate slot types has made great progress in this task. However, obtaining state values is still an under-studied problem. Existing extraction-based approaches cannot capture values that require the understanding of context and are not generalizable either. To address these issues, we propose a novel **S**tate **VA**lue **G**eneration based framework (**SVAG**), decomposing DST into state value generation and domain slot generation. Specifically, we propose to generate state values and use self-training to further improve state value generation. Moreover, we design an estimator aiming at detecting incomplete generation and incorrect generation for pseudo-labeled data selection during self-training. Experimental results on the MultiWOZ 2.1 dataset show that our method which has only less than 1 billion parameters achieves state-of-the-art performance under the data ratio settings of 5%, 10%, and 25% when limited to models under 100 billion parameters. Compared to models with more than 100 billion parameters, SVAG still reaches competitive results.[1]

Keywords: Dialogue State Tracking; Low-Resource Approach; Self-Training; Prompt Learning; Task-oriented Dialogue systems

1. Introduction

Dialogue State Tracking (DST) is a critical component in task-oriented dialogue systems. It aims to track the dialogue state at every dialogue turn, where the state is represented in forms of a set of (domain-slot, value) pairs. As the rise of new dialogue domains in practice, it poses a big challenge for scenarios with limited resources. Most previous work has attempted to tackle this challenge through cross-domain transfer learning (Wu et al., 2019; Dingliwal et al., 2021) and cross-task transfer learning (Gao et al., 2020; Lin et al., 2021; Shin et al., 2022). Moreover, pre-trained language models (PLMs) adaption methods (Wu et al., 2020a; Su et al., 2022) are proven to be effective for low-resource DST. However, all these methods suffer from domain or task dependencies. Recently,

---

[*] Corresponding Author
1. Our code is available at https://github.com/SLEEPWALKERG/SVAG

prompt learning has been applied to low-resource DST by Yang et al. (2022), the method of which demonstrates the potential of prompt learning for generating the slot type of a given state value with limited training data. Therefore, *"first obtain state values, and then generate slot types"* is pointed to be a promising direction for low-resource DST. The accuracy of state values is critical for the performance of such a two-step method for low-resource DST. However, how to obtain state values correctly is still under explored. Previous approaches (Yang et al., 2022) simply extract state values with the rule-based method, which significantly limits the accuracy and the generalization. We investigate the state values in DST and find that state value generation has three major issues as shown in Figure 1. First, some state values may not be extracted directly such as the state value *"don't care"*. A model should understand semantics, then generate these state values. Second, there may be multiple state values appearing in the utterance but only some of them can represent the user's intention. For example, in Figure 1, *"hotel"* is the state value but the *"guesthouse"* is not. So, a model should have the ability to distinguish them. Third, there are some state values that need to be inferred in context. For example, the state value *"centre"* in hotel booking should be inferred from the dialogue history. Simply extracting this information cannot obtain these correct state values, so we propose not to extract these values but use the generation method to generate correct state values.

Furthermore, since unlabeled dialogue data can be relatively easily obtained in real-world applications, we try to make use of these data to further improve the performance of state value generation. Therefore, we propose to self-train the state value generator to iteratively improve its performance, alleviating the above three issues. For self-training to be effective in the context of generation tasks, it is critical to select confident pseudo-labeled data. Previous methods (Zhu and Hauff, 2022; Mehta et al., 2022) use perplexity or some learned metrics to measure the quality of generated sequences. However, state values are not sequences. Therefore, we try to design a state value estimator to estimate the quality of the generated set of state values.

In this paper, we propose **SVAG**, a **S**tate **VA**lue **G**eneration based framework for low-resource DST with prompt learning and self-training, which decomposes DST into two subtasks: state value generation and domain slot generation. Specifically, we first propose a

Figure 1: Three main issues of state value generation in DST: (a) *"don't care"* should be generated, (b) *"hotel"* should be distinguished from *"guesthouse"*, and (c) *"centre"* should be inferred from the first turn. Words in blue are state values that can be extracted directly.

prompt based state value generator, which takes advantage of the PLM to address the three issues above. Second, we propose to self-train the state value generator to further improve its performance and propose a prompt based estimator to filter out noisy pseudo-labeled data during self-training. Finally, a prompt based domain slot generator is proposed to generate the corresponding slot type of a given state value. Experimental results show that SVAG reaches state-of-the-art results on MultiWOZ2.1 under the data ratio settings of 5%, 10%, and 25% when limited to models under 100 billion parameters, demonstrating the superiority of our proposed state value generation based
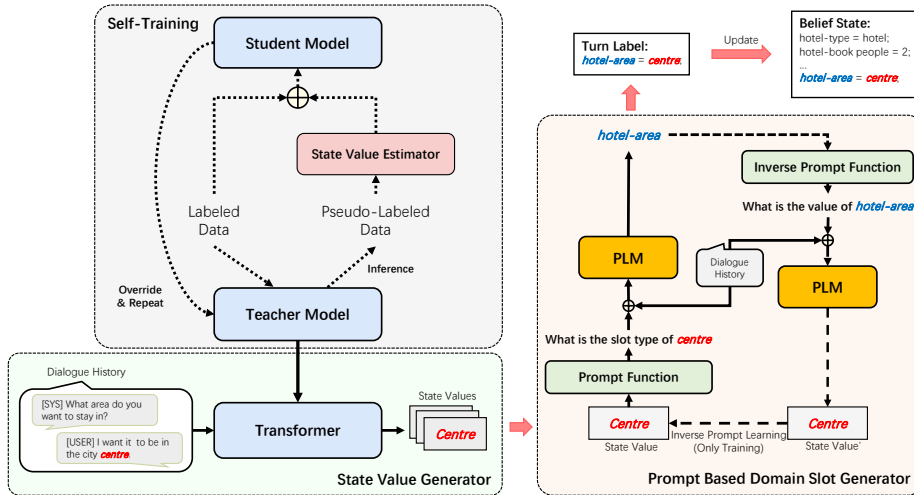
Figure 2: The overview of our proposed framework. There are three main components of our framework: a state value generator, a self-training strategy, and a domain slot generator. Given the dialogue history, the state value generator first generates the state values in the current turn, then the domain slot generator generates the slot type for each generated state value. Finally, we use the turn labels to update the belief state.

method with self-training. In addition, SVAG also achieves competitive results compared to methods based on models with more than 100 billion parameters.

The contributions of this paper are summarized as the following:

- We propose SVAG, an effective and general state value generation based framework for low-resource DST.

- We design an estimator with the goal of measuring both the accuracy and completeness of state value generation to filter out noisy pseudo-labeled data during self-training.

- Experimental results show that SVAG achieves competitive performance in low-resource DST.

2. Framework

In this section, we first set up the notations used throughout the paper. Then we will describe our proposed prompt based framework for low-resource DST which consists of three main components: (1) a state value generator which aims to generate the state values in the current turn; (2) a self-training strategy with a prompt based estimator which aims to boost the performance of the low-resource state value generator with fine-grained selected pseudo-labeled data; (3) a prompt based domain slot generator which aims to generate the corresponding slot type of a given state value. Figure 2 illustrates the whole framework.

Notation. Let us define $D = \{(S_1, U_1), (S_2, U_2), ..., (S_t, U_t)\}$ as the set of system response and user utterance pairs in T turns of a dialogue, where S_t and U_t represent the system's and

user's utterance respectively. Also, we define $T = \{T_1, T_2, ..., T_t\}$ as the turn label for each turn, where the turn label comprises multiple tuples of domain slots s and their associated values v ($T_t = \{(s_1, v_1), (s_2, v_2), ..., (s_n, v_n)\}$). In addition, we define $V_t = \{v_1, v_2, ..., v_n\}$ as all the state values in the t-th turn label.

2.1. State Value Generator

Given the t-th turn utterances $D_t = (S_t, U_t)$ and its history $D_{<t} = (S_{<t}, U_{<t})$, this model aims to generate the state values that the user mentions or confirms at the current turn. Following Guo et al. (2021), we denote the representation of the dialogue history before turn t as $D_{<t} = S_1 \oplus; \oplus U_1 \oplus; ...; \oplus S_{t-1} \oplus; \oplus U_{t-1}$ and $D_t = S_t \oplus; \oplus U_t$ is the input of current turn utterances. Finally, the input of the state value generator can be denoted as:

$$
\begin{aligned}
X_t = & PREFIX \oplus [HISTORY] \oplus D_{<t} \\
& \oplus [TURN] \oplus D_t,
\end{aligned} \tag{1}
$$

where $[HISTORY]$ and $[TURN]$ are two special tokens that indicate the start of the dialogue history before turn t and the start of the t-th turn utterances, respectively. $PREFIX$ in our model is *"get the requests that the user confirmed or mentioned in this turn"*. In addition, $\oplus$ is just a simple concatenation operation.

Given this input, a bi-directional transformer (Vaswani et al., 2017) encoder then outputs:

$$
H_t = Encoder(X_t), \tag{2}
$$

where $H_t \in \mathbb{R}^{L \times d}$, L is the length of the input sequence and d is the hidden dimension of the encoder. Then the decoder attends to the encoder output H_t and decodes the corresponding state values. The output is naturally a set of state values in our experiment, but this data structure is not supported by a traditional decoder. So we transfer the output to $V_{output} = v_1 \oplus | \oplus v_2 \oplus |...| \oplus v_n$. Then:

$$
\widehat{V}_{output} = Decoder(H_t). \tag{3}
$$

The overall learning objective of the proposed state value generation processing is to maximize the log-likelihood of V_{output} given the dialogue history before the t-th turn $D_{<t}$, the t-th turn dialogue D_t, and the $PREFIX$. That is:

$$
\sum log P(V_{output} | D_{\leq t}, PREFIX). \tag{4}
$$

2.2. Self-Training with a State Value Estimator

The process of our self-training approach is shown in Figure 2. First, the *Teacher* is applied to generate pseudo labels on unlabeled data U. Second, a *Student* is trained on the limited labeled data S and the confident pseudo-labeled data. Lastly, the trained *Student* becomes a new *Teacher*. Multiple iterations are computed till the accuracy no longer increases. The state value generator trained on S acts as the initial *Teacher*. For self-training to be effective in state value generation, it is crucial to carefully select confident pseudo-labeled data to mitigate the risk of reinforcing the model's mistakes.

We analyze the results of our state value generator and find that it sometimes generates only part of the state values or occasionally generates some incorrect values. Among all the bad cases, the
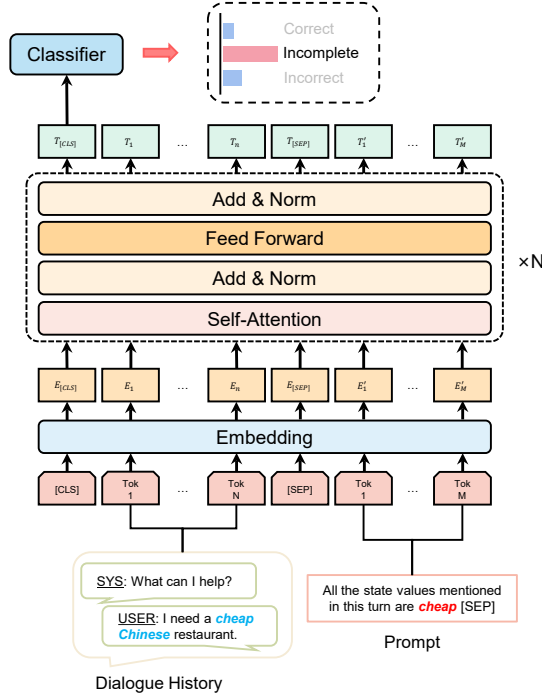
Figure 3: The model architecture of our proposed state value estimator. Given the dialogue history and the generated state values, the model predicts whether all the state values are correctly generated. This figure shows an example that the estimator detects incomplete generation.

incomplete generation problem occurs most frequently. So, to prevent self-training from further reinforcing the model's mistakes of incomplete generation and incorrect generation, we design a low-resource state value estimator with prompt learning to identify these mistakes as well as filter out these noisy pseudo-labeled data. In order to detect whether the model misses some correct state values or generates some incorrect values, we need a large number of negative samples. We propose synthesizing the examples using the limited labeled dataset.

We first describe our proposed state value estimator. Figure 3 illustrates the model architecture. Given a set of state values V, we first convert it to prompt. We manually design a template for the prompt. If there are no state values in the current turn, the prompt is *"there are no values mentioned in this turn."*. Otherwise, the prompt is *"all the values mentioned in this turn are $v_1, ..., v_n$."*. Then the input of the state value estimator can be denoted as:

$$X_t = [CLS] \oplus D_{<t} \oplus . \oplus D_t \oplus [SEP] \\ \oplus PROMPT \oplus [SEP], \tag{5}$$

where $[CLS]$ and $[SEP]$ are two special tokens.

Given this input, the output representation of the encoder is $H_t \in \mathbb{R}^{|X_t| \times d}$, and $h_t^{[CLS]}$ is the output of that corresponds to $[CLS]$. We then feed $h_t^{[CLS]}$ into an output layer for classification, which can be denoted as:

$$P = Softmax(MLP(h_t^{[CLS]})), \tag{6}$$

Dialogue History: (user) I need a train leaving broxbourne on Wednesday. Can you book it for me?
Turn Utterance: (system) Sure! What is your destination?
(user) I need to arrive in cambridge by 11:45.
POSITIVE:
[correct generation] All the values mentioned in this turn are cambridge and 11:45.
NEGATIVE:
[incomplete generation] All the values mentioned in this turn are cambridge ~~and 11:45~~.
[incomplete generation] All the values mentioned in this turn are ~~cambridge and~~ 11:45.
[incomplete generation] There are no values mentioned in this turn.
[incorrect generation] All the values mentioned in this turn are cambridge and Wednesday.

Figure 4: An example of our proposed negative sampling.

where MLP consists of two linear layers and a tanh activation function. In addition, $P \in \mathbb{R}^{|\mathbb{O}|}$ is the probability distribution for each label. In our formulation, $|\mathbb{O}| = 3$, because we set three labels to the estimator. They are (1) correct generation; (2) incomplete generation; (3) incorrect generation. The model is trained with the standard cross-entropy loss.

Finally, we describe how we construct the negative samples. To enhance the model's capability of detecting incomplete generation, which is the key problem of our state value generator, we create much more incomplete generation samples than the others. We randomly remove some values from the ground truth set of state values as incomplete generation samples. Additionally, we add state values that appeared in previous turns to the current state values set as incorrect generation samples. Figure 4 visualizes an example of our proposed negative sampling.

2.3. Prompt Based Domain Slot Generator

In this section, we describe the domain slot generation method. We use a prompt based model to generate the domain slot. Following Yang et al. (2022), we also add an inverse prompt mechanism to this model which aims to enhance the PLM to better understand the DST task. In our model, we use prompt function $f(v) =$ *"what is the slot type of [v]"* while $I(s) =$ *"what is the value of [s]"* is the inverse prompt (Yang et al., 2022). Given the state value v, we first construct the input of the PLM, which can be denoted as:

$$X_t = D_{<t} \oplus . \oplus D_t \oplus [SEP]$$
$$\oplus f(v) \oplus [SEP], \tag{7}$$

The overall learning objective of the domain slot generation processing is to maximize the log-likelihood of domain slot s given the dialogue history before the t-th turn $D_{<t}$, the t-th turn dialogue D_t, and the value-based prompt $f(v)$. The loss function can be denoted as:

$$\mathcal{L} = -\sum logP(s|D_{\leq t}, f(v)). \tag{8}$$

Because each turn label T_t may contain multiple (s, v) pairs, each pair of them constructs an instance for training and testing in the domain slot generation model. While testing, the value is generated by our proposed state value generator.

To achieve better domain slot generation performance under low-resource scenarios, we add inverse prompt learning while training like Yang et al. (2022). Just simply replace $f(v)$ with $I(s)$ in

X_t, then we get the inverse input. Similarly, the loss function for the inverse prompt mechanism is:

$$\widetilde{\mathcal{L}} = -\sum logP(v|D_{\leq t}, I(s)). \tag{9}$$

Finally, the loss function $\mathcal{L}^*$ consists of both loss functions in prompt learning and inverse prompt learning, which can be denoted as:

$$\mathcal{L}^* = \mathcal{L} + w \times \widetilde{\mathcal{L}}, \tag{10}$$

where w is a weight used to adjust the influence of the inverse prompt learning.

2.4. Belief State Updating

Previous sections have described how we get the turn labels. In this section, we describe how we use these turn labels to update the belief state. If the domain-slot-value is not consistent with the previous turns, then we update the corresponding value in the existing belief state. Otherwise, if they don't have the domain-slot-value tuple, we append it to the existing belief state. For example, in Figure 2, *"hotel-area"* doesn't exist in the previous belief state, we just add *"hotel-area-centre"* to it.

3. Experimentation

3.1. Datasets and Metrics

Datasets We conduct our experiments on the MultiWOZ 2.1 dataset (Eric et al., 2020). It is a multi-domain task-oriented dialogue dataset which contains 8438 dialogues for training, 1000 dialogues for validating, and 1000 dialogues for testing. Following existing work (Wu et al., 2019), only five domains (restaurant, hotel, attraction, taxi, train) are used in our experiments because the other two domains have very few dialogues and only appear in the training set.

Metrics The standard metric (Wu et al., 2019), joint goal accuracy (JGA) is used in our experiments. This metric compares the whole predicted belief state to the gold one at each dialogue turn. If and only if all the predicted states match the ground truth states exactly for all domains, the prediction is treated as correct. In addition, we use a turn level accuracy (TLA) metric to evaluate the performance of state value generation. This metric compares the predicted state values to the gold ones. Only if all the predicted state values match the ground truth values, the prediction is considered correct.

3.2. Implementation Details

We implement the state value generator based on T5 (Raffel et al., 2020) as well as the domain slot generator and the state value estimator is based on RoBERTa (Liu et al., 2019). We use the pre-trained checkpoint from *transformers* library[2]. Additionally, we also use *pytorch lightning* library[3] to implement our framework. All models are trained using the AdamW (Loshchilov and Hutter, 2017) optimizer with a linear learning rate decay. The peak learning rate of the two generators is 5e-5 while 2e-5 is for the state value estimator. We conduct our experiments under the data ratio setting of 1%, 5%, 10%, and 25%. For domain slot generation, we set w in Eq 10 to 0.1 (Yang et al., 2022).

---

2. https://huggingface.co/t5-large; https://huggingface.co/roberta-base

3. https://www.pytorchlightning.ai

Model	Param. Size	Unlabeled MultiWOZ Data	External Data	Data ratio			
				1%	5%	10%	25%
TRADE (Wu et al., 2020b)		No	-	10.4	27.7	32.6	38.5
BERT (Wu et al., 2020a)		No	-	6.4	19.6	32.9	40.8
BERT* (Mi et al., 2021)		No	-	8.0	-	21.2	-
SGPDST (Lee et al., 2021)		No	-	32.1	43.1	46.9	-
TOD-BERT (Wu et al., 2020a)		No	ToD Data	9.9	28.6	39.5	44.3
TOD-BERT* (Mi et al., 2021)		No	ToD Data	8.4	-	25.5	-
DS2 (Shin et al., 2022)	<1B	No	Summary Data	33.8	44.2	45.4	-
PL-DST (Yang et al., 2022)		No	Labeled ToD Data	**44.3**	44.7	44.7	45.4
Self-Sup (Wu et al., 2020b)		Yes	-	19.5	30.6	34.5	40.2
BERT-ST (Mi et al., 2021)		Yes	-	8.8	-	23.9	-
TOD-BERT-ST (Mi et al., 2021)		Yes	ToD Data	9.9	-	28.3	-
SVAG (Our Method)		Yes	-	31.9	**45.1**	**47.6**	**50.0**
SVAG w/o ST		No	-	31.9	43.5	44.6	48.2
IC-DST GPT-Neo 2.7B (Hu et al., 2022)		No	-	16.7	26.9	31.7	-
IC-DST CodeGen 2.7B (Hu et al., 2022)	<100B	No	-	20.7	29.6	33.8	-
SM2-3B (Chen et al., 2023)		No	Labeled ToD Data	38.1	39.9	39.9	-
SM2-11B (Chen et al., 2023)		No	Labeled ToD Data	38.4	44.6	46.0	-
IC-DST CodeX-davinc 175B (Hu et al., 2022)	>100B	No	-	43.1	47.1	48.7	-

Table 1: Comparison of different models' JGA scores for low-resource DST on MultiWOZ 2.1 (Eric et al., 2020) under different ratios of training data. We report the averaged JGA score of SVAG over three runs. Bolded numbers indicate highest performance on models under 1 billion parameters. *: taken from Mi et al. (2021).

To select the best checkpoint of the state value estimator for pseudo-labeled data selection, we validate the model with our synthesized validation dataset. The F1-score of correct generation examples is used to choose the best model. We choose the threshold of our state value estimator based on the blank rate[4] of the pseudo-labeled data as it gets higher when the threshold goes up. Finally, the threshold of the estimator is set to 0.98.

3.3. Baseline Models

We compare our proposed method with several strong baselines for low-resource DST.

TRADE (Wu et al., 2020b) uses a Seq2Seq model to decode the corresponding value for each predefined slot with a soft copy mechanism. It is trained with limited MultiWOZ data.

BERT (Wu et al., 2020a) treats DST as a multi-class classification problem using a predefined ontology and is trained with limited MultiWOZ data.

SGPDST (Lee et al., 2021) uses domain, slot, and slot description as prompt and fine-tunes a PLM to generate the corresponding value with limited MultiWOZ data.

TOD-BERT (Wu et al., 2020a) continues BERT's pre-training on several external task-oriented dialogue (ToD) datasets, then adopts the model to low-resource DST like BERT (Wu et al., 2020a).

DS2 (Shin et al., 2022) uses a template based method to convert dialogue states to summaries and reformulate DST as dialogue summary. They first fine-tune the model with dialogue summary datasets and then fine-tune it with limited MultiWOZ data.

---

4. blank here means that there are no state values that should be generated at the current turn

Data ratio	ST-iter.	No. of samples	TLA	JGA
1%	-	578	66.71%	**30.77%**
	1	+30,852	**67.28%**	30.55%
	2	+5,963	66.22%	29.03%
5%	-	2,807	74.20%	42.43%
	1	+38,918	**76.13%**	**45.06%**
	2	+2339	75.65%	44.29%
10%	-	5,626	75.37%	43.24%
	1	+28,956	**77.66%**	**47.48%**
	2	+2,482	77.65%	47.27%
25%	-	13,932	78.23%	49.02%
	1	+20,334	**78.95%**	**50.61%**
	2	+1,255	78.60%	49.62%
Full data	-	56668	79.87%	51.41%

Table 2: Model performance over multiple self-training iterations under different data ratio settings. *"ST-iter."* denotes the iteration of self-training.

PL-DST (Yang et al., 2022) uses a prompt based method to generate the slot type for a given state value with limited MultiWOZ data. Their model is based on SOLOIST which is pre-trained with several external ToD datasets and adds DST to pre-training tasks. Moreover, their model ignores domain information and might use the ground truth domain while evaluating.

Self-Sup (Wu et al., 2020b) adds two self-supervised objectives for TRADE using the rest of unlabeled MultiWOZ data to improve its performance in low-resource scenarios.

BERT-ST (Mi et al., 2021) proposes a text augmentation technique for the limited MultiWOZ data and then self-trains BERT (Wu et al., 2020a) by using the rest of unlabeled MultiWOZ data and the augmented data.

TOD-BERT-ST (Mi et al., 2021) proposes the same method as BERT-ST (Mi et al., 2021) on top of TOD-BERT. So it not only utilizes the rest of unlabeled MultiWOZ data but also uses external ToD data for pre-training.

IC-DST (Hu et al., 2022) reformulates DST as a text-to-SQL task and uses in-context learning to prompt a CodeX model with limited MultiWOZ data.

SM2 (Chen et al., 2023) stabilizes in-context learning by using meta-learning with external labeled ToD data and then leverages in-context learning to DST with limited MultiWOZ data.

3.4. Main Results

Following the previous work (Wu et al., 2019), we randomly select limited labeled data from the training set to simulate the low-resource scenarios with three different random seeds (10, 20, and 48). We conduct our experiments using 1%, 5%, 10%, and 25% data. Note that 1% data has only 84 dialogues. Table 1 shows the JGA score of our method and other baselines in different data ratio settings. Among the models using unlabeled MultiWOZ data, our framework SVAG achieves state-of-the-art performance. We observe that SVAG has a measurable improvement over TRADE

Dialogue history: [user] I would like a reservation for 2 to the Peking restaurant.
Current Turn Utterances: [sys] OK, and what day and time would you like that reservation?
[user] I would like to make a reservation for Saturday at 11:45. And there has been a change in plans, I will be dining alone.
Without-ST prediction: Saturday, 11:45
With-ST prediction: Saturday, 11:45, **1**
Ground truth: Saturday, 11:45, 1

Dialogue history: [user] I am staying in Cambridge soon and would like to stay at a and b guest house. ... [sys] Your booking is successful! ... [sys] there are actually 7 museums in that area. [user] Great, can I get the postcode, entrance fee and address of 1 of them?
Current Turn Utterances: [sys] cafe jello gallery has a free entrance fee. The address is cafe jello gallery, 13 magdalene street and the post code is cb30af. Can I help you with anything else?
[user] Yes please. I need a taxi to commute.
Without-ST prediction: cafe jello gallery
With-ST prediction: cafe jello gallery, **a and b guest house**
Ground truth: a and b guest house, cafe jello gallery

Table 3: Examples that self-training outperforms without self-training in state value generation.

(Wu et al., 2020b) and BERT (Wu et al., 2020a). SVAG also outperforms TOD-BERT (Wu et al., 2020a), DS2 (Shin et al., 2022), and PL-DST (Yang et al., 2022) under the data ratio setting of 5%, 10%, and 25%, although all of them are enhanced by external data.

In particular, we observe that SGPDST (Lee et al., 2021) and DS2 (Shin et al., 2022) achieves a higher JGA score than ours under the data ratio setting of 1%. SGPDST (Lee et al., 2021) achieves better performance because it not only uses domain slot information but also leverages slot description to facilitate the low-resource DST, making the model better understand the task. However, it is costly to collect all these information. Our model doesn't rely on a given schema. DST is naturally a process of summarizing important information. Therefore, DS2 (Shin et al., 2022) achieves great performance in extreme low-resource scenarios because it reformulates DST as dialogue summary with rule-based templates from dialogue states and pre-trains the model on external summary data. However, their model cannot be applied to general scenarios since such great performance comes from the manual rules and the extra dialogue summary data. Additionally, constructing manual rules for new domains is costly. Conversely, our method does not rely on any other annotation, providing a more general and efficient solution for low-resource DST.

We also observe that PL-DST (Yang et al., 2022) achieves the best result under the data ratio setting of 1% because it is based on SOLOIST (Peng et al., 2021) which is pre-trained on several external ToD datasets and adds DST to pre-training tasks. Furthermore, their framework also ignores domain information and might use the ground truth domain while evaluating, so it is unfair to compare it to all the other methods. Domain prediction is actually not that easy if there are close-by domains that share the same slot types, such as in MultiWOZ. In addition, it simply extracts state values with a rule-based method, which significantly limits their model's generalization. We observe that the more data, the better SVAG is, which demonstrates the effectiveness of our method.

Moreover, we observe that SVAG significantly outperforms Self-Sup (Wu et al., 2020b) and BERT-ST (Mi et al., 2021) with a great margin under all data ratio settings. These two models are comparable to our method since both of them use unlabeled MultiWOZ data and don't use any external data. The higher JGA scores of SVAG demonstrate not only the superiority of our proposed

---

Dialogue history: [sys] Saigon city is Asian oriental and is in the north as the hotel is. It is expensive. Shall I book it for you ? [user] Yes, please . I need reservations for 12:30 on Friday. There are 5 in my group.
Generated state values: Friday, 5, 12:30
Ground truth: Friday, 5, 12:30, **Saigon city**
Correct Score: 0.08
Incomplete Score: 0.92 ✓
Incorrect Score: 0.00

Dialogue history: [user] I would like to visit a park on the north side. [sys] Sure , we have milton country park located in the north in milton. ... [user] OK, great thanks. I also need to find a train going to Cambridge
Generated state values: milton country park
Ground truth: Cambridge
Correct Score: 0.03
Incomplete Score: 0.02
Incorrect Score: 0.95 ✓

Table 4: Examples that our proposed state value estimator correctly detects different errors in state value generation. The three scores are the estimator's predicted probability of the three types of generation.

state value generation based method for low-resource DST but also the efficiency of our proposed state value estimator based self-training for utilizing unlabeled MultiWOZ data. We also observe that SVAG outperforms TOD-BERT-ST (Mi et al., 2021) which not only uses unlabeled MultiWOZ data but also incorporates external ToD data into pre-training, indicating that SVAG can make better use of the PLM's strong capability of comprehension to fulfill the DST task.

Recently, as the rise of large language models (LLM), many in-context learning based approaches have been proposed for DST. Although SVAG has less than 1B parameters, SVAG still outperforms SM2 (3B & 11B)(Chen et al., 2023) under the data ratio settings of 5% and 10%. Additionally, SVAG outperforms IC-DST (<100B) (Hu et al., 2022) under all data ratio settings. SM2 achieves a much higher JGA socre than ours under the data ratio setting of 1% because it uses several labeled ToD datasets for meta-learning and the base model of it is much bigger than ours. Moreover, SVAG also achieve comparable performance compared to IC-DST(175B) (Hu et al., 2022) under the data ratio setting of 5% and 10%, although its parameters are more than 100 times ours.

3.5. Effectiveness of Self-Training

In this section, we will analyze the effectiveness of self-training for low-resource state value generation. We run 2 iterations of self-training in our experiment. Table 2 reports the performance of our state value estimator based self-training under different data ratio settings. We observe that a significant improvement on TLA is made in the first iteration. We then analyze the pseudo-labeled data selected by the estimator and find that both the quantity and the quality of pseudo-labeled data with high confidence in the first iteration are high. In the second iteration, pseudo-labeled data with high confidence is greatly reduced but noisy examples are increased, resulting in a little decrease

in the performance. In addition, we observe the performance of our estimator gets better with the increase of data ratio settings, which makes the performance degradation less obvious.

In the last line of table 2, we also report the results under the full data setting. We observe that our method achieves a close performance over the full data setting when 25% data is available, which indicates the effectiveness of our proposed state value generator with a state value estimator based self-training strategy. Specially, under the data ratio setting of 1%, we observe that JGA is not proportional to TLA. It is because we update the belief state with the turn labels. The earlier the error occurs, the greater the impact on JGA due to the error accumulation.

Table 3 shows two examples of our proposed state value generator with self-training. In the first example, the user briefly informs the system that he/she will book the table for one person by saying *"alone"*. The model without self-training misses the state value *"1"*. After self-training, the model can better understand semantics and generate it. In the second example, the user informs that he/she needs a Taxi to commute from the guest house to the attraction by saying *"I need a Taxi to commute"*. The model without self-training misses the guest house, while the model after self-training can better infer information through context. To sum up, our proposed state value estimator based self-training can significantly enhance the model's capability of natural language understanding (NLU) and generate the correct state values in low-resource scenarios.

3.6. Effectiveness of the State Value Estimator

In this section, we will analyze the effectiveness of our proposed state value estimator. Table 4 shows two examples of the state value estimator, in which the estimator correctly identifies incomplete generation and incorrect generation. It demonstrates that our proposed state value estimator correctly distinguishes incomplete generation from correct generation, which mitigates the risk of reinforcing incomplete generation for the state value generator.

We also compare the performance of the state value estimator based pseudo state values selection with that of *vanilla* self-training. For each experiment, we *randomly* sample an equal number of examples for vanilla self-training. Table 5 summarizes the results. We observe that the state value estimator based self-training improves over vanilla self-training under all data settings. It not only demonstrates the effectiveness of our proposed state value estimator based self-training but also confirms that our negative sampling based synthesized dataset is effective to train the estimator with the purpose of evaluating both the accuracy and completeness.

Data ratio	Selection strategy	TLA
1%	-	66.71%
	vanilla	67.25%
	value estimator	**67.28%**
5%	-	74.10%
	vanilla	75.72%
	value estimator	**76.13%**
10%	-	75.37%
	vanilla	76.23%
	value estimator	**77.66%**
25%	-	78.23%
	vanilla	78.80%
	value estimator	**78.95%**

Table 5: Comparing performance in terms of Turn Level Accuracy between vanilla and state value estimator based pseudo state values selection strategies. Selection strategy *"-"* denotes training without self-training.

4. Related Work

4.1. Low-Resource Dialogue State Tracking

Various approaches have been proposed to low-resource DST. One line of these methods is cross-domain transfer (Wu et al., 2019; Dingliwal et al., 2021), which aims at transferring knowledge from one domain to the others. The reason why these methods do work is that many domains share a lot of common slots. The other line of work can be summarized as cross-task transfer (Gao et al., 2020; Lin et al., 2021; Shin et al., 2022). These methods try to leverage data from another task to facilitate the low-resource DST. For example, Gao et al. (2020) modeled DST as machine reading comprehension (MRC). They first pre-trained a model on MRC data, then further trained the model with DST data. In addition, PLM adaption methods have been proven to be efficient for low-resource DST. Wu et al. (2020a) continued BERT's pre-training on several ToD datasets, then adopted the obtained TOD-BERT to DST task. However, all these methods suffer from domain or task dependencies. Additionally, various in-context learning based methods (Hu et al., 2022; Chen et al., 2023) have been proposed to leverage LLMs to low-resource DST. However, these methods are highly dependent on the inference language model and the cost of inference is much higher too. Recently, Yang et al. (2022) introduced a prompt based slot generation framework for low-resource DST. However, they obtained state values by a rule-based method, which limits their model's generalization. Different from these methods, we propose a state value generation based method for low-resource DST.

4.2. Pseudo-Labeled Data Selection in Self-Training

For self-training, how to deal with noisy, low-quality pseudo labels is crucial to the final performance. Self-training is originally designed for classification problems (He et al., 2020). It is easy for these classification models to filter out noisy labels by their 'confidence', which is the predicted probability of the label. For self-training in NLG, confidence estimation has been defined as the task of evaluating the quality of the whole sequence of words in the target sentence (Zhu and Hauff, 2022). Zhu and Hauff (2022) tried to use both sentence perplexity and the BERT-based fluency score to represent the quality. Mehta et al. (2022) repurposed BLEURT to be a quality estimator. However, our work aims to evaluate the quality of a set of generated state values, rather than the sequences. We propose a prompt based estimator for measuring both the accuracy and completeness of the generated set of state values.

5. Conclusion and Future Work

In this paper, we propose SVAG, a prompt based framework for low-resource DST which consists of a state value generator and a domain slot generator. We reveal three issues in state value generation and propose a prompt based state value generator to alleviate them. In order to make use of large amounts of unlabeled dialogue data, we propose to self-train the state value generator. In addition, a state value estimator is designed to filter out noisy pseudo-labeled data during self-training. Moreover, we synthetically generate datasets for training the estimator with the goal of detecting incomplete generation and incorrect generation. Experimental results on MultiWOZ2.1 illustrate the superiority of SVAG over previous approaches for low-resource DST.

In the future work, we plan to mitigate exposure bias in state value generation to alleviate the problem of incomplete generation.

Acknowledgments

References

Derek Chen, Kun Qian, and Zhou Yu. Stabilized in-context learning with pre-trained language models for few shot dialogue state tracking. In Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023, pages 1506–1519. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-eacl.115.

Saket Dingliwal, Shuyang Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, and Dilek Hakkani-Tür. Few shot dialogue state tracking using meta-learning. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 1730–1739. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.eacl-main.148.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020, pages 422–428. European Language Resources Association, 2020. URL https://aclanthology.org/2020.lrec-1.53/.

Shuyang Gao, Sanchit Agarwal, Tagyoung Chung, Di Jin, and Dilek Hakkani-Tür. From machine reading comprehension to dialogue state tracking: Bridging the gap. CoRR, abs/2004.05827, 2020. URL https://arxiv.org/abs/2004.05827.

Jinyu Guo, Kai Shuang, Jijie Li, and Zihan Wang. Dual slot selector via local reliability verification for 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 139–151. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.12. URL https://doi.org/10.18653/v1/2021.acl-long.12.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=SJgdnAVKDH.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. In-context learning for few-shot dialogue state tracking. In Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 2627–2643. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.findings-emnlp.193.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Dialogue state tracking with a language model using schema-driven prompting. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4937–4949, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main. 404. URL https://aclanthology.org/2021.emnlp-main.404.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul A. Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. Zero-shot dialogue state tracking via cross-task transfer. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 7890–7900. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.622. URL https://doi.org/10.18653/v1/2021.emnlp-main.622.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. CoRR, abs/1711.05101, 2017. URL http://arxiv.org/abs/1711.05101.

Sanket Vaibhav Mehta, Jinfeng Rao, Yi Tay, Mihir Kale, Ankur Parikh, and Emma Strubell. Improving compositional generalization with self-training for data-to-text generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 4205–4219. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.289. URL https://doi.org/10.18653/v1/2022.acl-long.289.

Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. Self-training improves pre-training for few-shot learning in task-oriented dialog systems. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 1887–1898. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.142. URL https://doi.org/10.18653/v1/2021.emnlp-main.142.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. SOLOIST: building task bots at scale with transfer learning and machine teaching. Trans. Assoc. Comput. Linguistics, 9:907–824, 2021. doi: 10.1162/tacl_a_00399. URL https://doi.org/10.1162/tacl_a_00399.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. Dialogue summaries as dialogue states (ds2), template-guided summarization for few-shot dialogue state tracking. In Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May

22-27, 2022, pages 3824–3846. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.302. URL https://doi.org/10.18653/v1/2022.findings-acl.302.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. Multi-task pre-training for plug-and-play task-oriented dialogue system. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 4661–4676. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.319. URL https://doi.org/10.18653/v1/2022.acl-long.319.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 808–819. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1078. URL https://doi.org/10.18653/v1/p19-1078.

Chien-Sheng Wu, Steven C. H. Hoi, Richard Socher, and Caiming Xiong. 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 2020, Online, November 16-20, 2020, pages 917–929. Association for Computational Linguistics, 2020a. doi: 10.18653/v1/2020.emnlp-main.66. URL https://doi.org/10.18653/v1/2020.emnlp-main.66.

Chien-Sheng Wu, Steven C. H. Hoi, and Caiming Xiong. Improving limited labeled dialogue state tracking with self-supervision. In Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of Findings of ACL, pages 4462–4472. Association for Computational Linguistics, 2020b. doi: 10.18653/v1/2020.findings-emnlp.400. URL https://doi.org/10.18653/v1/2020.findings-emnlp.400.

Yuting Yang, Wenqiang Lei, Juan Cao, Jintao Li, and Tat-Seng Chua. Prompt learning for few-shot dialogue state tracking. CoRR, abs/2201.05780, 2022. URL https://arxiv.org/abs/2201.05780.

Peide Zhu and Claudia Hauff. Unsupervised domain adaptation for question generation with DomainData selection and self-training. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 2388–2401, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.183. URL https://aclanthology.org/2022.findings-naacl.183.