Meta-Reinforced Multi-Domain State Generator for Dialogue Systems

Yi Huang, Junlan Feng, Min Hu, Xiaoting Wu, Xiaoyu Du, Shuo Ma
JIUTIAN Team, China Mobile Research
{huangyi,fengjunlan,humin}@chinamobile.com
{wuxiaoting,duxiaoyu,mashuo}@chinamobile.com

Abstract

A Dialogue State Tracker (DST) is a core component of a modular task-oriented dialogue system. Tremendous progress has been made in recent years. However, the major challenges remain. The state-of-the-art accuracy for DST is below 50% for a multi-domain dialogue task. A learnable DST for any new domain requires a large amount of labeled in-domain data and training from scratch. In this paper, we propose a **Me**ta-**Re**inforced Multi-Domain State Generat**o**r (MERET). Our first contribution is to improve the DST accuracy. We enhance a neural model based DST generator with a reward manager, which is built on policy gradient reinforcement learning (R-L) to fine-tune the generator. With this change, we are able to improve the joint accuracy of DST from 48.79% to 50.91% on the MultiWOZ corpus. Second, we explore to train a DST meta-learning model with a few domains as source domains and a new domain as target domain. We apply the model-agnostic meta-learning (MAML) algorithm to DST and the obtained meta-learning model is used for new domain adaptation. Our experimental results show this solution is able to outperform the traditional training approach with extremely less training data in target domain.

1 Introduction

A Dialogue State Tracker (DST) is a core component of a modular task-oriented dialogue system (Young et al., 2013). For each dialogue turn, a DST module takes the user utterance and the dialogue history as input, and outputs a belief estimate of the dialogue state. The dialogue state as of today is simplified as a set of requests and goals, both of which are represented as *(slot, value)* pairs such as *(area, centre)*, *(food, Chinese)* for a user request *I'm looking for a Chinese restaurant in the centre of the city*. A highly accurate DST is crucial to ensure the quality and smoothness of a human-machine dialogue.

Figure 1: An example of dialogue state tracking process for booking a hotel, looking for an attraction and booking a taxi between them. Each turn contains a user utterance (grey) and a system utterance (blue). The dialogue state tracker (yellow) tracks all the *(domain, slot, value)* until the current turn. Blue color texts indicate mentions of slot values appeared at that turn. Best viewed in color.

Budzianowski et al. (2018) recently introduced a multi-domain dialogue dataset Multi-domain Wizard-of-Oz (MultiWOZ), which is more than one order of magnitude larger than all previous annotated task-oriented corpora with around 10k dialogues and involves more than 7 domains. A domain of a task-oriented system is often defined by an ontology, which defines all entity attributes called slots and all possible values for each slot. MultiWOZ presents conversation scenarios much similar to those in real industrial applications. Figure 1 shows an example of a multi-domain dialogue, where a user starts a conversation about hotel reservation and moves on to look for attractions nearby

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7109–7118

of his interest. It adds a layer of complexity to the DST and brings new challenges.

The first new challenge is how to appropriately model DST for a multi-domain dialogue task. Multi-domain DST is in its infancy before MultiWOZ (Rastogi et al., 2017). Most previous work on DST focus on one given domain (Henderson et al., 2013, 2014; Mrkšić et al., 2017; Zhong et al., 2018; Korpusik and Glass, 2018; Liu et al., 2019). As Wu et al. (2019) pointed out, to process the MultiWOZ data, the DST model has to determine a triplet *(domain, slot, value)* instead of a pair *(slot, value)* at each turn of dialogue. MultiWOZ contains 30 *(domain, slot)* pairs over 4,500 possible slot values in total. The prediction space is significantly larger. This change seems quantitative. However, it challenges the foundation of most successful DST models, where DST is casted as a neural model based classification problem, each *(slot, value)* pair is an independent class and the number of classes is relatively limited. When the number of classes is large enough as the case in MultiWOZ, classification-based approaches are not applicable. In real industry scenarios, the prediction space is even larger and it is often not possible to have full ontology available in advance (Xu and Hu, 2018). It's hard to enumerate all possible values for each slot. The second challenge is how to model the commonality and differences among domains. The number of domains is unlimited in real-life. It won't be able to scale up if each new domain requires a large amount of annotated data.

To overcome these challenges, Wu et al. (2019) proposed a TRAnsferable Dialogue statE generator (TRADE) that generates dialogue states from utterances using a copy mechanism, facilitating knowledge transfer between domains. The prominent difference from previous one-domain DST models is that TRADE is based on a generation approach instead of a close-set classification approach. The generation model parameters are shared among various domains and slots. TRADE is able to help boost the DST accuracy up to 48.62% with the MultiWOZ corpus. It is obvious this accuracy is far from being acceptable.

In this paper, we are motivated to enhance this generation-based approach for two objectives, higher accuracy and better domain adaptability. To improve DST accuracy, we propose a new framework which contains the state generator and reward manager. The state generator follows the same setup of TRADE. The Reward Manager calculates the reward to fine-tune the generator through policy gradient reinforcement learning (PGRL). We use the reward manager to help the generator alleviate the objective mismatch challenge. Objective mismatch is a limitation of encoder-decoder generation approaches, where the training process is set to maximize the log likelihood, but it doesn't assure producing the best results on discrete evaluation metrics such as the DST accuracy. Since MultiWOZ provides data for multiple domains, it enables us to study the long-standing domain adaptability problem. It is a hope we can train a general DST model from multi-domain data and this model can be adapted to a new domain with minimal examples from a new domain. We apply the meta-learning algorithm, MAML, for this study. Our key contributions in this paper are as follows:

- We propose a new framework as the DST model, which contains a neural model based DST generator and a reward manager.
- With our proposal, we are able to improve the joint accuracy of DST from 48.79% to 50.91%, which is 2.12% absolute improvement over the latest state-of-the-art on the MultiWOZ corpus.
- We apply MAML to train a meta-learning DST model with a few domains as the training domains and a new domain as the testing domain. Our experimental results show this solution is able to outperform the traditional training approach with only 30% of the in-domain training data.
- To our knowledge, we are the first to apply RL and MAML into DST.

2 Model MERET

The overview of our model is illustrated in Figure 2. It consists of a generator model and a reward manager.

2.1 The Generator

In this paper, we take TRADE as our baseline. The TRADE model comprises three components: (1) an utterance encoder, (2) a context-enhanced slot classifier, (3) a state generator. We briefly describe the TRADE model in this Section.

The utterance encoder encodes dialogue utterances into a sequence of fixed-length vectors.

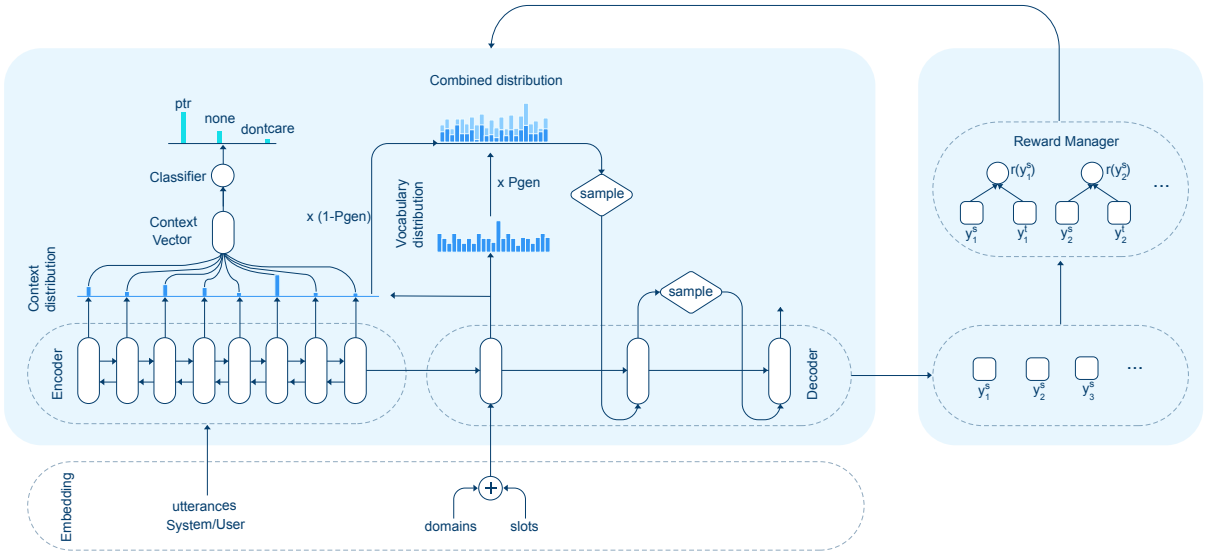

Figure 2: The architecture of the proposed MERET model, which contains a Generator and a Reward Manager in general. The Generator includes (a) an utterance encoder, (b) a context-enhanced slot classifier, and (c) a state generator. The Reward Manager calculates the reward values based on the reward functions to fine-tune the generator through PGRL.

TRADE uses Bi-GRU (Chung et al., 2014), to encode. Instead of initializing by concatenating GloVe embeddings (Pennington et al., 2014), our model explore to use BERT (Devlin et al., 2019) as embedding model. We denote a sequence of dialogue turns as a matrix $X_t = [U_{t-l}, R_{t-l}, ..., U_t, R_t] \in \Re^{|X_t| \times d_{emb}}$, where l is the length of the dialogue history selected, U is the user turn, R represents the system response and d_{emb} indicates the turn-level embedding size. The encoder encodes X_t into a hidden matrix $H_t = [h_1^{enc}, ..., h_{|X_t|}^{enc}] \in \Re^{|X_t| \times d_{hdd}}$, hdd is the hidden size.

The state generator uses GRUs as the decoder, which takes the embedding of the jth (domain,slot) pair as well as the kth word as input and outputs a hidden vector h_{jk}^{dec} at the kth decoding step. This hidden vector is then mapped to distribution over the vocabulary V and over the dialogue history as shown in Eq (1).

$$P_{jk}^{vocab} = \text{Softmax}(E \cdot (h_{jk}^{dec})^\top) \in \Re^{|V|} \quad (1)$$
$$P_{jk}^{history} = \text{Softmax}(H_t \cdot (h_{jk}^{dec})^\top) \in \Re^{|X_t|}$$

These two distributions are combined as Eq (2) as the final results,

$$P_{jk}^{final} = p_{jk}^{gen} \times P_{j_k}^{vocab} + (1 - p_{jk}^{gen}) \times P_{j_k}^{history} \quad (2)$$

The context-enhanced slot classifier takes as input H_t and classifies it into one of the three classes: *ptr, none, dontcare*. With a linear layer parameterized by $W_g \in \Re^{3 \times d_{hdd}}$, the slot classifier for the jth *(domain, slot)* pair is defined as

$$G_j = \text{Softmax}(W_g \cdot (P_{j0}^{history} \cdot H_t)^\top) \in \Re^3 \quad (3)$$

If this slot classifier determines $none$ or $dontcare$, the system ignores any output from the state generator.

Optimization is performed jointly for both the state generator and the slot classifier. The cross-entropy loss is used for both, with L_s representing the loss for the slot classifier and L_g for the generator. They are combined with hyper-parameters η and σ.

$$L_{mix} = \eta L_s + \sigma L_g \quad (4)$$

2.2 A Reward Manager

Generally, the cross-entropy loss is used to train a generator. In our task, the true words Y_j^{label} is used and the cross-entropy loss can be defined as:

$$loss_g = -\sum_{j=1}^{J} \sum_{k=1}^{|Y_j|} \log \left(P_{jk}^{final} \cdot (y_{jk}^{label})^\top \right) \quad (5)$$

where y_{jk}^{label} is the ground truth of the value word for the jth $(domain, slot)$ pair.

In this paper, we propose a RL-based Reward Manager to work the generator. The Reward Manager is used for calculating the reward to fine-tune the Generator through PGRL.

The specific modeling process of reinforcement learning adaptation for DST task is summarized in Algorithm 1: We treat the Generator as the target agent to be trained. The agent interacts with an external environment (utterances, domains, slots and reward manager) by taking actions and receiving environment state and reward. The actions are the choices of tokens for slot value that generates for any given *(domain, slot)* pair. The action space is the vocabulary. Following each action, the reward manager calculates a reward by comparing the generated token to the corresponding ground-truth token. When reaching the last decoding step, the agent updates its parameters towards maximizing the expected reward. RL loss is defined as follows:

$$L_{rl} = -\sum_{j=1}^{J}\sum_{k=1}^{|Y_j|} r(y_{jk}^s)\log\left(P^{final}(y_{jk}^s)\right) \quad (6)$$

where y_{jk}^s is a token sampled from the vocabulary probability distribution and $r(y_{jk}^s)$ means the reward for the sampled token y_{jk}^s, computed by a reward function. Intuitively, the loss function L_{rl} enlarges the probability of the sampled y_{jk}^s if it obtains a higher reward for the kth token in jth *(domain, slot)* pair.

We also define a combined loss function:

$$L = \mu L_{rl} + \lambda L_{mix} \quad (7)$$

where L_{rl} is defined as the reinforcement learning loss, L_{mix} is the cross-entropy loss from TRADE, μ and λ are the combined hyper-parameters. Algorithm 1 shows how this method works.

Algorithm 1 REINFORCE algorithm

Input: Dialogue history sequence X, ground-truth output slot value sequences Y, a pre-trained model π_θ.

Output: Trained model $\pi_{\theta'}$ with REINFORCE algorithm.

Training Steps:
Initialize π_θ with random weights θ;
Pre-train π_θ using cross-entropy loss of generator and classifier on dataset (X, Y);
Initialize $\pi_{\theta'} = \pi_\theta$.
while not done **do**
  Select a batch of size N from X and Y;
  for each slot **do**
    Sample $\{Y^s = (y_1^s, \cdots, y_{|Y_j|}^s)\}_1^N$ from the final probability distribution of vocabulary;
    Compute reward $\{r(y_1^s), \cdots, r(y_{|Y_j|}^s)\}_1^N$ defined in the Reward Manager;
  end for
  Compute L_{rl} and L using Eq (6) and Eq (7);
  Update the parameters of network with learning rate ρ, $\theta' \leftarrow \theta' + \rho\nabla_{\theta'} L_{\theta'}$;
end while
Testing Steps:
for batch of X and Y **do**
  Generate the output $\widehat{Y}$;
end for
return The evaluated model $\pi_{\theta'}$;

3 MAML-adaptive DST

The traditional paradigm of supervised learning is to train a model for a specific task with plenty of annotated data. Meta-learning aims at learning new tasks with few steps and little data based on existing tasks. MAML (Finn et al., 2017) is the most popular meta-learning algorithm. It has been successfully employed in various tasks. We propose to apply MAML to perform dialogue state tracking for new domains. The MAML algorithm tries to build an internal representation of multiple tasks and maximize the sensitivity of the loss function when applied to new tasks, so that small update of parameters could lead to large improvement of new task loss value. In this paper, we explore how it works with DST, a key component in task-oriented dialogue systems.

MAML is compatible for any model training based on gradient descent. We can denote the baseline model as M. Training a typical gradient descent model M involves (1) providing training data and initializing parameters of M; (2) computing a given objective loss; (3) applying gradient descent to the loss to update M parameters. With MAML, the training steps becomes: (1) Initialize M and making nd copies of M to be M'_d; (2) Select training data from each domain and updating M'_d parameters based on gradient descent and a loss function; (3) Calculate a loss for each domain with their updated temporary model M'_d; (4) Sum up the new loss from each training domain to be a total loss; (5) Update parameters of the original M based on the total loss; (6) Repeat above steps until M converges.

Algorithm 2 shows step-by-step how MAML combines with our model MERET. Suppose we consider nd dialogue domains, we take ntr do-

mains as source domains for meta-training and nts domains as target domains for meta-testing. For each source domain, we divide the source domain data into D_d^{train} as the support dataset and D_d^{valid} as the query dataset, d is the domain index. α, β are two hyper-parameters for MAML, α as the learning rate for each domain and β as the learning rate for meta-learning update.

There are two cycles. The outer cycle is for meta-learning, updating model parameters of M. The inner cycle is for task learning, updating the temporary model M'_d of each domain d. For task learning, we select K examples from D_d^{train} for each domain d, evaluate the gradient of the loss function as Eq (7), update the parameters θ'_d with respect the K examples (Step 4). After each domain model is updated once, the M model parameters are updated using the sum of the loss with respect to K' examples sampled from each D_d^{valid}. Specifically, we sum the loss of M'_d in each domain to obtain the meta loss L_M,

$$L_M = \sum_d L_d(M'_d, D_d^v) \tag{8}$$

Finally, we minimize the meta loss for updating the current model M until an ideal meta-learned model M is achieved,

$$M \leftarrow M - \beta \nabla_M \sum_d L_d(M'_d, D_d^v) \tag{9}$$

To adapt to a new domain, we start with the meta-learned model M instead of initializing randomly, new-domain training data is used to update model parameters as multiple batches and the learnt task model is fit for the new domain.

Algorithm 2 MAML algorithm

Input: D_d^{train}; D_d^{valid}; α; β.
Output: Trained model M with MAML algorithm.

1: **while** not done **do**
2: **for** each domain d **do**
3: Select a batch of size from D_d^{train} and D_d^{valid} to get D_d^t and D_d^v;
4: Pre-update model with gradient descent: $M'_d \leftarrow M - \alpha \nabla_M L_d(M, D_d^t)$
5: Compute $L_d(M'_d, D_d^v)$ using D_d^v;
6: **end for**
7: Update the current model M: $M \leftarrow M - \beta \nabla_M \sum_d L_d(M'_d, D_d^v)$
8: **end while**
9: **return** meta-learned model M;

4 Experiments

4.1 Dataset and Evaluation Matrix

In this paper, we use MultiWOZ as our training and testing corpus. MultiWOZ is a fully-labeled collection of human-human written conversations spanning over multiple domains and topics. It contains 8438 multi-turn dialogues with on average 13.7 turns per dialogue. It has 30 *(domain, slot)* pairs and over 4,500 slot values. We use the most frequent five domains *(restaurant, hotel, attraction, taxi, train)* in our experiments.

Two common metrics to evaluate DST models are joint goal accuracy and slot accuracy. Joint accuracy measures the accuracy of dialogue states, where a dialogue state is correctly predicted only if all the values of for all the *(domain, slot)* pairs are correctly predicted. Slot accuracy is the accuracy of the *(domain, slot, value)* tuples. Joint accuracy is a more challenging metric.

4.2 Implementation Details

For all experiments, we choose Bi-GRU networks with a hidden size of 768 to be the encoder and the decoder. The model is optimized using Adam (Kingma and Ba, 2015) with a learning rate of 0.001. We reduce the learning rate to half if the validation loss increases. We set the batch (Ioffe and Szegedy, 2015) size to 32 and the dropout (Zaremba et al., 2014) rate to 0.2. Different reward functions have been tried through the experiment progress. We choose a binary reward that a positive value is given when the output token equals the target and a punishment otherwise, 1 and -0.1 respectively. We evaluate the model every epoch and adopt early stopping on the validation dataset. In meta-training phase, we set different numbers of updating M' due to the differences in slot complexity for each domain. The model was implemented in the pyTorch.

4.3 Multi-domain Results

Table 1 shows our experimental results with MERET. MERET achieves the joint goal accuracy of 50.91%, which is 2.12% above the latest state-of-the-art DST model COMER and is 2.29% higher than TRADE. Table 1 also shows accuracies of a few latest systems on the same corpus. MERET is also able to obtain the best slot accura-

DST Models	Joint Acc	Slot Acc
MultiWOZ Benchmark (Budzianowski et al., 2018)	25.83	–
GLAD (Zhong et al., 2018)	35.57	95.44
HyST (ensemble)(Goel et al., 2019)	44.22	–
TRADE (Wu et al., 2019)	48.62	96.92
COMER (Ren et al., 2019)	48.79	–
MERET	**50.91**	**97.07**
-BERT	50.35	96.98
-RL	50.09	97.01

Table 1: The evaluation of existing multi-domain DSTs on MultiWOZ. MERET has the highest joint accuracy, which surpasses current state-of-the-art model. The baseline for the MultiWOZ dataset is taken from Budzianowski et al. (2018)

New Domain (Proportion)	Training Model	Joint Acc	Slot Acc
Taxi (1%)	Training from scratch	60.57	73.25
	Fine-tuning TRADE	59.03	78.65
	MERET	**64.37**	**83.20**
Attraction (1%)	Train from scratch	27.88	63.43
	Fine-tuning TRADE	29.05	62.24
	MERET	**43.10**	**74.32**

Table 2: Evaluation on *taxi* and *attraction* new domains. MERET outperforms learning from scratch and TRADE fine-tune with the same data on both new domains.

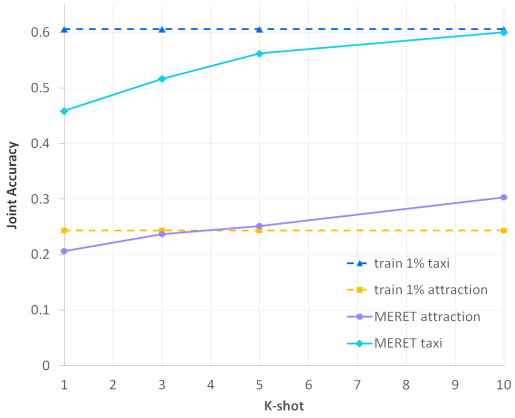

Figure 3: K-shot results of different experimental settings. Performance of our model surpasses training from scratch on attraction domain with K=5.

cy 97.07% which is slightly higher than TRADE, but not substantial. To prove the effectiveness of our structure, we conduct ablation experiments in different setups. MERET-BERT(remove BERT, acc 50.35%, +1.73%) has the same embedding Glove with TRADE, the improvement here mainly comes from RL, benefitting from the reward manager, which provides an ability for the entire model to explore rather than to be greedy at every single step and overcomes the existing limitation of encoder-decoder generation approach as mentioned in the intro. MERET-RL(remove RL, acc 50.09%, +1.47%) shows the increment due to embedding changes, which uses BERT instead of Glove, integrating powerful pre-trained language representation of BERT. We can see that MERET's advantage mainly comes from the RL. The way we employ RL with the generator in this paper is a good baseline. We are encouraged by these experimental results for future exploration in this line of research.

4.4 New Domain Results

To test the effectiveness of MERET, we choose *hotel, train* and *restaurant* as the source domains, *taxi* and *attraction* as the target domains. For each source domain, we utilize 3000 dialogues on average and 200 dialogues for training and testing. We utilize 30 dialogues (1% of source domain) for training on new domains with the pre-trained model. In our experiments, we conducted comparison studies with three setups, (1) Training a MERET model from scratch using 1% sampled data from each target domain, (2) Meta-training a MERET model using the source domain data and then fine-tuning with 1% sampled data from each target domain, (3) Training a TRADE model using the source domain data and then fine-tuning

(a) Error type of TRADE. (b) Error type of MERET.

Figure 4: Distributions of different error type for two models' comparison.

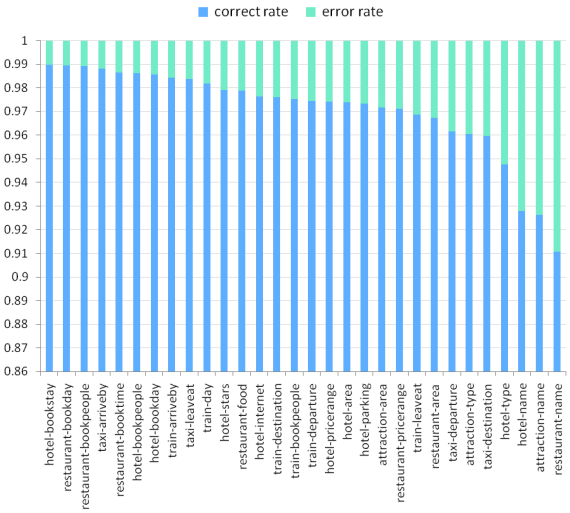

Figure 5: Overview of correct-error rate for multi-domain slots. The book stay slot in hotel domain and name slot in restaurant domain has the highest and lowest correct rate respectively, 98.97% and 91.06% correspondingly.

with 1% sampled data from each target domain. Experimental results are listed in Table 2. MERET achieves substantial higher accuracy, 64.7% joint goal accuracy for the Taxi domain and 43.10% for the Attraction domain, comparing to the other two setups. Similar advantages are obtained for slot accuracies for both target domains.

To explore the K-shot performance of the MERET model, we conduct experiments to measure the impact of the number of training examples from the target domain. We meta-train MERET with source domains and meta-test on the *taxi* and *attraction* domain. The number of training samples K from the target domains varies from 1 to 10. We use $K = (1, 3, 5, 10)$ as the testing point. Figure 3 illustrates our experiments. It's natural that the accuracy increases as the training data increases. We can observe that the accuracy with $K = 5$ of

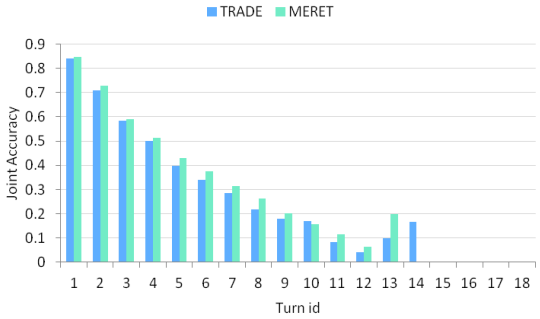

Figure 6: The changes of joint accuracy over dialogue turns. The performance of our model MERET gradually emerges as the number of dialogue turns increases with the help of RL maximizing reward expectations.

the *attraction* domain surpasses the accuracy with training MERET from scratch using 1% (30 dialogues) of the *attraction* domain data. This demonstrates our model's capability to achieve good performance with a fraction of the target data.

4.5 Analysis and Discussion

We analyze the wrong predictions and draw a heat map of distributions for the slot classifier considering the importance of its determining to the final output. From the map in Figure 4, we can see the main cause of the error-maker is the classifier's inertia of omit-prediction from *ptr* to *none*, which stands up to 47.3% proportion. The over-prediction cause comes in the next, with a 27.3% rate. Value on the diagonal of the lower-left corner shows the mis-prediction rate of the generator. Combined with the comparison of the two pictures, we can get the point that our proposed model has a higher generative ability over state value.

An overview correct-error analysis of multi-domain for slots is shown in Figure 5. The number-related slots book stay in *hotel* domain and book day in *restaurant* domain have the highest correct rates, 98.97% and 98.94%, respectively.The name-related slots in the *restaurant*, *attraction*, and *hotel* domains have the highest error rates, 8.94%, 7.36%, and 7.21%, respectively. It is because these slots usually have a large number of possible values set and high annotation errors. The type slot of *hotel* domain also keeps a high error rate in different experiments, even if it is an easy task with only two possible values in the ontology. The reason is that labels of the *(hotel, type)* pair are usually missing in the dataset. We further show the performance of our model over different dialogue turn in Figure 6. As the number of dialogue turn increases,

User:	I'm looking for a jamaican restaurant in the east.
System:	There are no jamaican restaurants in the east. Would you like to try another food type or area?
User:	I'm looking for a place that serves jamaican food in the east. If not, ***italian*** will do.
System:	There is one Italian place in the east, Pizza Hut Fen Ditton.
TRADE prediction:	{ (restaurant, area, east), (restaurant, food, ***jamaican***) }
MERET prediction:	{ (restaurant, area, east), (restaurant, food, ***italian***) }

Table 3: Case study for state Generator. We can find that with the same context, MERET outperforms TRADE in terms of state generation for DST.

the influence of context gradually appears for the final results due to the abilities of different models. We can see that MERET outperforms TRADE gradually. This is especially true when the context length is long. Our model can carry information over multiple turns which will be used for state generator with the help of RL maximizing rewards expectations in a better way. We sample one typical dialogue from MultiWOZ to demonstrate the effectiveness of MERET in the case study. Due to limited space, we present the same key parts derived from two models and the details are shown in Table 3. We observe that the constraint for food slot is dynamic and MERET is sensitive to capture this context information with the advantage of RL-based fine-tune state Generator, which reinforces in greater exploration for DST and maximizes reward expectation in a better way.

5 Related Work

Mrkšić et al. (2017) propose neural belief tracking (NBT) framework without relying on hand-crafted semantic lexicons. The model uses Convolutional Neural Networks (CNN) or Deep Neural Networks (DNN) as dialogue context encoder and makes a binary decision for *(slot,value)* pairs. Zhong et al. (2018) propose global-local modules to learn representations of the user utterance and system actions and calculate similarity between the contextualized representation and the *(slot,value)* pair. Xu and Hu (2018) utilize pointer network to track dialogue state, which proposes a conception of unseen states and unknown states earlier. Chao and Lane (2019) use BERT as dialogue context encoder and get contextualized representation, which is passed to the classification module and get three classes: none, dontcare, span. When the class is span, the start and end positions of slot values are obtained in the dialogue context. However, Both Xu and Hu (2018) and Chao and Lane (2019) suffers from the fact that they can not get correct answer when the value does not exist in the input. Wu et al. (2019) propose an approach that the model generates a sequence of value from utterances by copy mechanism, which can avoid the case that the value is not in the input. It also uses a three-way classifier to get a probability distribution over none, dontcare, ptr classes. Ren et al. (2019) achieve state-of-the-art performance on the MultiWOZ dataset by applying a hierarchical encoder-decoder structure for generating a sequence of belief states. The model shares parameters and has a constant inference time complexity.

Reinforcement learning is a way of training an agent during interaction with the environment by maximizing expected reward. The idea of policy gradient algorithm has been applied in training of sequence to sequence model. Ranzato et al. (2016) propose MIXER algorithm, which is the first application of REINFORCE algorithm (Williams, 1992) in training sequence to sequence model. However, an additional model, which is used to predict expected reward, is required in MIXER. Rennie et al. (2017) proposed a self-critical method for sequence training (SCST). It directly optimizes the true, sequence-level, evaluation metric, and avoids the training of expected future rewards estimating model. Paulus et al. (2018) applied SCST in summary generation, which improved the rouge value of generated result. SCST algorithm was also used by Zhao et al. (2018) for improving story ending generation. Keneshloo et al. (2018) present some of the most recent frameworks that combine concepts from RL and deep neural networks and explain how these two areas could benefit from each other in solving complex seq2seq tasks.

Meta-learning aims at learning target tasks with little data based on source tasks. This algorithm is

compatible with any model optimized with gradient descent so that it has a wide range of applicability. Meta-learning has been applied in various fields such as image classification (Santoro et al., 2016; Finn et al., 2017) and robot manipulation (Duan et al., 2016; Wang et al., 2016), etc. In the field of natural language processing, some exploratory work (Gu et al., 2018; Huang et al., 2018; Qian and Yu, 2019; Madotto et al., 2019) have been proposed in recent years. Most of them are focused on the generation-related tasks and machine translation. To our knowledge, few related work in dialogue state tracking (DST) was found till now. We propose to apply model-agnostic meta-learning (MAML) (Finn et al., 2017) algorithm for training a DST meta-learning model with a few domains as the training domains and a new domain as the testing domain to achieve multi-domain adaptation.

6 Conclusion

We introduce an end-to-end generative framework with pre-trained language model and copy-mechanism, using RL-based generator to encourage higher semantic relevance in greater exploration space for DST. Experiments on multi-domain dataset show that our proposed model achieves state-of-the-art performance on the DST task, exceeding current best result by over 2%. In addition, we train the dialogue state tracker using multiple single-domain dialogue data with rich-resource by using the MAML. The model is capable of learning a competitive and scalable DST on a new domain with only a few training examples in an efficient manner. Empirical results on MultiWOZ datasets indicate that our solution outperforms non-meta-learning baselines training from scratch, adapting to new few-shot domains with less data and faster convergence rate.

In future work, we intend to explore more with the combination of RL and DST on the basis of reward designing, trying to explore more in the internal mechanism. In the long run, we are interested in combing many tasks into one learning process with meta-learning.

Acknowledgments

We thank Zhiqiang Yang and Chao Deng for their insightful discussion and great support. We also thank all anonymous reviewers for their constructive comments.

References

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. pages 5016–5026.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. arXiv:1907.03040.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. 2016. Rl ^{2}: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. 2019. Hyst: A hybrid approach for flexible and accurate dialogue state tracking. *CoRR*, abs/1907.00883.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Steve J. Young. 2013. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 467–471.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 732–738.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456.

Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K. Reddy. 2018. Deep reinforcement learning for sequence to sequence models. *CoRR*, abs/1805.09461.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Mandy Korpusik and James R. Glass. 2018. Convolutional neural networks for dialogue state tracking without pre-trained word vectors or semantic dictionaries. In *2018 IEEE Spoken Language Technology Workshop*, pages 884–891.

Qingbin Liu, Shizhu He, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Copy-enhanced heterogeneous information learning for dialogue state tracking. *CoRR*, abs/1908.07705.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5454–5459.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 2639–2649.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations*.

Abhinav Rastogi, Dilek Hakkani-Tur, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *IEEE Automatic Speech Recognition and Understanding Workshop*.

Liliang Ren, Jianmo Ni, and Julian J. McAuley. 2019. Scalable and accurate dialogue state tracking via a hierarchical sequence generation. *CoRR*, abs/1909.00754.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1179–1195.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 1842–1850.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. 2016. Learning to reinforcement learn. arXiv:1611.05763.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

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.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1448–1457, Melbourne, Australia. Association for Computational Linguistics.

Steve Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.

Yan Zhao, Lu Liu, Chunhua Liu, Ruoyao Yang, and Dong Yu. 2018. From plots to endings: A reinforced pointer generator for story ending generation. In *Natural Language Processing and Chinese Computing - 7th International Conference*, pages 51–63.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.