MTSS: Learn from Multiple Domain Teachers and Become a Multi-Domain Dialogue Expert

Shuke Peng,[1,2] **Feng Ji,**[2] **Zehao Lin,**[1,2] **Shaobo Cui,**[2] **Haiqing Chen,**[2] **Yin Zhang**[1*]
[1]College of Computer Science and Technology, Zhejiang University
[2]DAMO Academy, Alibaba Group
{pengsk, georgelin, zhangyin98}@zju.edu.cn, {zhongxiu.jf, yuanchun.csb, haiqing.chenhq}@alibaba-inc.com

Abstract

How to build a high-quality multi-domain dialogue system is a challenging work due to its complicated and entangled dialogue state space among each domain, which seriously limits the quality of dialogue policy, and further affects the generated response. In this paper, we propose a novel method to acquire a satisfying policy and subtly circumvent the knotty dialogue state representation problem in the multi-domain setting. Inspired by real school teaching scenarios, our method is composed of multiple domain-specific teachers and a universal student. Each individual teacher only focuses on one specific domain and learns its corresponding domain knowledge and dialogue policy based on a precisely extracted single domain dialogue state representation. Then, these domain-specific teachers impart their domain knowledge and policies to a universal student model and collectively make this student model a multi-domain dialogue expert. Experiment results show that our method reaches competitive results with SOTAs in both multi-domain and single domain setting.

1 Introduction

Spoken Dialogue Systems (SDS) are widely used as assistants to help users in processing daily affairs such as booking tickets or reserving hotels. A typical dialogue system consists of three key components: spoken language understanding (SLU), dialogue manager (DM), and natural language generation (NLG)(Maes 2005; Maes and Gopalakrishnan 2006). Within the procedure above, dialogue state representation is crucial since DM needs a precise representation of the present dialogue state to select an appropriate action. There are mainly two types of approaches for dialogue state representation: the state tracking approach and the hidden vector approach. The state tracking approach is to use a belief state tracker to extract the ontology from users' utterances (Sun et al. 2014; Mrksic et al. 2017; Zhong, Xiong, and Socher 2018). Those extracted ontology, known as slots, are used as the state representation. The hidden vector approach, more popular utilized in end-to-end dialogue systems, is to use the hidden vector compressed from users' utterance as the state presentation (Serban et al. 2016; Yao, Zweig, and Peng 2015). The aforementioned approaches are almost satisfactory in a single domain setting dialogue task such as tickets booking since the number of the slots, and the entities are relatively small in a single domain setting. Nevertheless, the performance of existing dialogue state representation approaches deteriorates rapidly when it comes to multi-domain setting. For the state tracking approach, the ontology space grows enormous in multi-domain dialogue systems. This growing ontology space leads to the accuracy degeneracy of dialogue state tracking, which limits the performance of dialogue systems. As for the hidden state representation approach, the human-labelled semantic information cannot be fully used. Besides, a hidden state representation is almost a black box which makes the dialogue system incomprehensible and hard to debug. The poor-quality and inaccurate multi-domain dialogue state representation severely limits the quality of multi-domain dialogue policy and further affects the overall performance of dialogue systems.

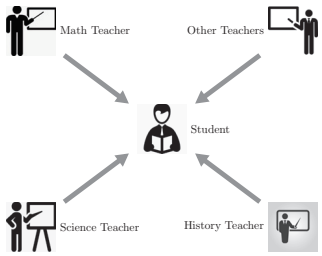

Figure 1: Learning scenarios in school

To build a satisfactory multi-domain dialogue system, we propose a model named Multiple Teachers Single Student (MTSS) to subtly circumvent the complex multi-domain dialogue state representation problem and learn a quality dialogue policy in a multi-domain setting. We use multiple teacher models (one for one domain to learn a satisfying domain-specific dialogue policy) to teach a student model to become a multi-domain dialogue expert. Our intuition comes from a real-life scenario in which a student has to learn many subjects such as Math, History and Science (see Figure 1). Usually, there is a full-time teacher

*Corresponding Author: Yin Zhang, zhangyin98@zju.edu.cn

regarding each subject. These teachers impart their professional knowledge of their respective subjects to a student. In other words, this student acquires a comprehensive understanding of all subjects by learning from these teachers. This well-educated student can achieve high performance in all subjects. This MTSS learning pattern is well-suited for the multi-domain dialogue systems. More specifically, **firstly**, for each domain of a multi-domain dialogue corpus, an individual teacher model is employed to learn dispersed dialogue knowledge and semantic annotations as the extra information in this single domain. Each domain teacher takes dialogue history utterances and human-labelled semantic from its corresponding domain as the dialogue state. Based on these domain-specialized dialogue state representation, these *customized* teachers can acquire a high-quality dialogue policy. **Secondly**, these well-trained domain-specific teachers in first step *impart* their learnt knowledge and dialogue policy to a universal student model through text-level guiding and policy-level guiding. We use knowledge distillation (Hinton, Vinyals, and Dean 2015; Kim and Rush 2016) to implement this guiding process. By learning from these domain-specific teachers, the universal student model acquires multi-domain knowledge and labelled semantic information and it finally becomes a multi-domain dialogue expert.

To sum up, the contributions are summarized as follows:

- We propose a novel multi-domain dialogue system. Our model subtly circumvents the knotty multi-domain dialogue state representation problem by using multiple teacher models to learn domain-specific dialogue knowledge. With their acquired knowledge and policies, these domain-specific teacher models collectively make a single student model become a multi-domain dialogue expert.
- Based on MTSS, we propose a novel approach to transferring the knowledge of domain teacher models to this single student model. These teacher models guide the student model not only from the text-level but also from policy-level, which collaboratively pass the teachers' knowledge and policies to the student model.

2 Related work

Multi-domain dialogue systems Recently, multi-domain dialogue systems have attracted increasing attention. The rule-based multi-domain dialogue systems (Pakucs 2003) are faced with the insufficiency of the scalability. With the development of deep learning, some multi-domain dialogue systems models are proposed based on neural network (Wen et al. 2016; Ultes et al. 2017). Zhao, Xie, and Eskénazi (2019) propose the Latent Action Reinforcement Learning (LaRL) model, which uses reinforcement learning to train a policy module to select the best latent action. The Hierarchical Disentangled Self-Attention (HDSA) (Chen et al. 2019) model uses hierarchical dialogue act representation to deal with the large size of dialogue acts. Both two works were applied in the MultiWOZ (Budzianowski et al. 2018) dataset and achieved excellent results.

The representation of dialogue states A commonly-used approach to representing dialogue states is to use the multi-hot embedding vector of human-defined features as the state representation. This type of approach needs an external dialogue state tracker to recognize correct features from users' utterance. Many works have been done on this issue, such as a rule-based state tracker (Sun et al. 2014) or a Neural Belief Tracker (NBT) (Mrksic et al. 2017). Some works are focusing on state trackers that track user intent and slot values in multi-domain settings (Rastogi, Hakkani-Tür, and Heck 2017; Goel et al. 2018). In addition to using human-defined features as the dialogue state representation, another approach is to use the hidden state vector generated directly from the raw text as the state representation. Without handcrafted features, Hierarchical Recurrent Encoder-Decoder (HRED) based dialogue systems(Sordoni et al. 2015; Serban et al. 2016; 2017) encode the dialogue history into a hidden vector to represent the current dialogue state in open-domain dialogue systems.

The Teacher-student Framework The teacher-student framework was first applied in the neural network by Hinton, Vinyals, and Dean (2015) in the knowledge distillation approach. In the teacher-student framework, a massive teacher model transfers their knowledge to a much smaller student model or several assembled teacher models collectively transfer their knowledge to a student model. Recent works show that knowledge distillation based teacher-student method works well in a language model (Kim and Rush 2016). Tan et al. (2019) proposed a multi-teacher single-student architecture to solve the multilingual neural machine translation problem. Individual models are built as teachers, and the multilingual model is trained to fit both the ground truth and the outputs of individual models simultaneously through knowledge distillation. In this way, the student model can reach comparable or even better accuracy in each language pair than these teacher models. Our work adopts a similar architecture, but we focus on multi-domain dialogue systems, which is more challenging since it involves complicated multi-domain dialogue policy learning.

3 The Framework of Multiple Teachers Single Student (MTSS) Model

In this section, we present the framework of our proposed Multiple Teachers Single Student (MTSS) model in Section 3.1 and detail the teacher and the student component in Section 3.2 and Section 3.3 respectively. We leave how the multiple teacher models impart their acquired knowledge to the student model in Section 4.

3.1 The Overview of MTSS

The overview of MTSS is presented in Figure 2 (For a clear illustration, we only plot two teacher models in the figure, which is sufficient to illustrate the whole framework and the working procedure). MTSS consists of two types of components: the student model and the teacher model. There are N teacher models and one single student in MTSS, where N is the number of dialogue corpus domain. In other words, each teacher model in MTSS is associated with one domain of the dialogue corpus. In the training phase, the teacher model and the student model are trained with different input:

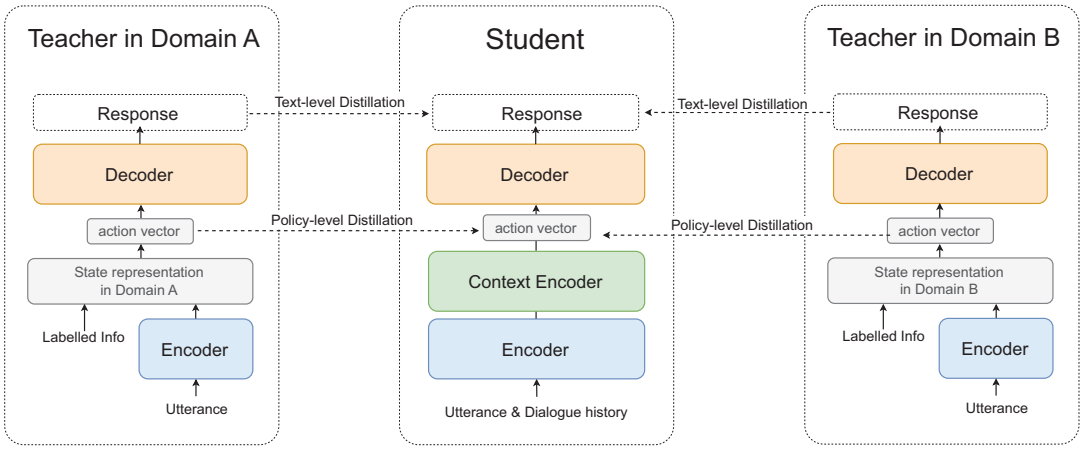

Figure 2: The teacher-student framework that transfers the knowledge from teachers to the student.

- The teacher takes the utterance and the human-labelled states as the input. The states labelled by human are of the highest accuracy, provide the teacher model sufficient information in dialogue policy decision and responding.
- The student takes the utterance and the history dialogues as the input.

These well-trained teacher model impart their knowledge in both text-level and policy-level. The text-level guidance is to make the student generate a similar response as the teacher models while the policy-level is to make the student learn the policies of these teachers, which make sure the student model can fully *assimilate* the knowledge of teachers. We will introduce the details of interactions between the student model and teacher models in Section 4.

After the training phase, the student model has acquired sufficient multi-domain knowledge and a satisfying multi-domain dialogue policy. At the testing phase, the student model only takes raw context utterances as input and can generate high-quality responses.

3.2 Multiple Teachers: One Teacher for One Domain

The structure of the teacher model We adopt Budzianowski et al. (2018) as the basic structure. As shown in Figure 3, it contains three parts: the encoder, the decoder and a middle policy model that takes both the utterance representation u_t as well as the human-defined feature e_t as the input. The feature consists of two vector representations. The first part is the belief state vector $\mathbf{v}_\mathrm{b}$, where each dimension of the vector stands for the one-hot value of a specific slot in each domain, a slot value receiving from the user. If the slot value appears, the corresponding value in the vector is set to 1. Otherwise, the value is 0. Thus all values of $\mathbf{v}_\mathrm{b}$ stand for necessary information the system keep at the current state. At every turn, the belief state is updated according to the semantic labelling of the users' utterances. Another construct of the state is the database pointer vector $\mathbf{v}_\mathrm{kb}$, where a database pointer vector stands for the number of the corresponding entities that match the request of the user. We use a 4-dimensional one-hot embedding vector, and each position embedding means separately 0, 1, 2 and more than 3 candidate entities. We concatenate three vectors: the utterance vector $\mathbf{v}_t^u$, the belief state $\mathbf{v}_\mathrm{b}$, and the database pointer $\mathbf{v}_\mathrm{kb}$, to get the vector of the current state s_t in the conversation.

Then we feed the concatenated vector to the policy model. The vector is processed with a nonlinear layer with $\tanh$ as the activation function, and the action vector $\mathbf{a}_t$ is generated from this layer:

$$\mathbf{a}_t = \tanh(\mathbf{w} \cdot [\mathbf{v}_t^u; \mathbf{v}_\mathrm{b}; \mathbf{v}_\mathrm{kb}]),$$

where $[;]$ stands for concatenation. The action $\mathbf{a}_t$ is finally delivered to the decoder module and the response is generated with an addition of the attention mechanism. We train teacher models individually in each domain. Thus the meaning of the belief state differs in teachers. After the teachers are well pre-trained in all domains, we take the teachers as the guidance to train the student model using the teacher-student framework.

Training of the teacher model The teacher model directly learns from the ground truth. For a teacher model, given the user utterance u and the state representation s, the purpose of the model is to minimize the negative log likelihood loss between the generated response $\hat{r}$ with a ground truth response $r = \{w_0^r, w_1^r, ..., w_m^r\}$. That can be written as:

$$J_\mathrm{NLL}(\hat{r}|u,s) = -\sum_{i=0}^{m} \sum_{\hat{w}_i \in \mathcal{V}} \mathbb{1}\{\hat{w}_i = w_i^r\} \log p(\hat{w}_i|u, s, w_{0\sim i-1}^r; \phi), \quad (1)$$

where the $\mathcal{V}$ is the vocabulary of all possible words, ϕ is the parameters of the teacher model and the symbol $\mathbb{1}\{\cdot\}$ stands for the indicator function.

3.3 Single Student: A Universal Multi-domain Dialogue System

The structure of the student model The universal dialogue system, also the student model is the final produced

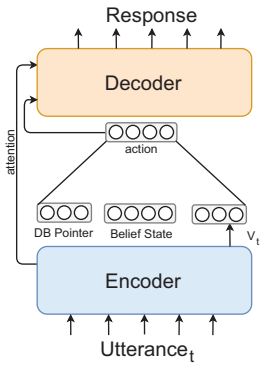

Figure 3: The teacher model pre-trained from each domain

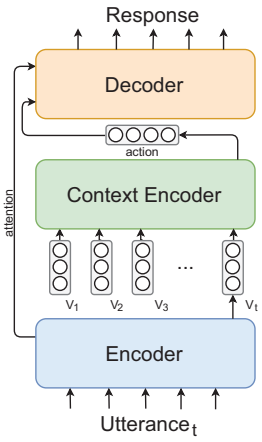

Figure 4: The student model architecture

model of our framework. The universal model takes no extra state information as the input. And it should have the ability to model the whole context, summarize the history states directly from the text. Under such consideration, we adopt the HRED (Sordoni et al. 2015; Serban et al. 2016) model as our universal dialogue system's base architecture. We use an encoder module to encode the user utterance to a latent vector representation and summarize all utterances' vectors with a context-level encoder in hierarchical encoder-decoder architecture, as shown in Figure 4. At the time t, for an utterance u_t contains m words $(\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_m)$. The encoder is an LSTM (Hochreiter and Schmidhuber 1997) network:

$$\mathbf{h}_t = \mathbf{v}^w_{tm} = \mathrm{LSTM}_e(\mathbf{h}_0; \mathbf{w}_{t0}, \mathbf{w}_{t1}, ...\mathbf{w}_{tm}),$$

Then we consider the last hidden state of the LSTM as the utterance representation vector $\mathbf{v}^u_t = \mathbf{h}_t$, and take the hierarchical encoder as the context-level policy module. The action $\mathbf{a}_t$ is made based on the all history utterances. We use another LSTM as the context-level encoder:

$$\mathbf{a}_t = \mathrm{LSTM}_c(\mathbf{v}^u_0, \mathbf{v}^u_1, ..., \mathbf{v}^u_t)$$

The action $\mathbf{a}_t$ is in the form of an abstract latent vector, serving as the guidance for the dialogue system to make proper responses. By regarding the context-encoder output as the action representation, we'll see how this representation facilitates the performance of our model using the teacher-student framework.

The action is fed into the generation part lately. The NLG module regards the action as the initial state of LSTM and generates the final response r_t. With the addition of the attention mechanism, the decoder model can be written as:

$$\mathbf{v}^r_i = \mathrm{LSTM}_d(\mathbf{a}_t, \mathbf{v}^w_{0\sim m}, \mathbf{v}^r_{0\sim i-1}),$$

where $\mathbf{v}^w_j$ is the output of the encoder in the position of the j-th word .

The guidance from ground truth for the student model Same as the training process of teacher model, the student model learns for the ground truth too. In contrast to the input for a teacher model, there is no explicit state representation as an input for the student. Instead, the student needs to summarize the hidden state from the context input itself. In addition to the guidance from the ground truth, the student model also learns from domain teachers, which will be elaborated in Section 4.

4 How Does The Single Student Learn from Domain Teachers?

In this section, we elaborate on the methods of transferring the knowledge from domain teachers to the student model. This transferring process can also be viewed as knowledge distillation (Hinton, Vinyals, and Dean 2015; Kim and Rush 2016) from teacher models to the single student model. These domain-specific teachers cooperatively guide the student model from both text-level (Section 4.1) and policy-level (Section 4.2), which makes sure the student can fully absorb the knowledge of these domain-specific teachers.

4.1 Text-level Guiding

We expect that the student should output a similar response as the teachers do. At each timestep, the student model is expected to generate the same output distribution as the teachers do. To enforce this objective, we use the cross entropy loss to measure the probability similarity between the output distributions of student and the teachers. The loss of the text-level distillation is:

$$J_{\mathrm{KD}} = -\sum_{i=0}^{m} \sum_{w^r_i \in \mathcal{V}} p(w^r_i | u, s, w^r_{0\sim i-1}; \phi) \log p(w^r_i | u, c, w^r_{0\sim i-1}; \theta), \quad (2)$$

in which ϕ is the parameter of the teacher models and θ is the parameter of the student model. And $\mathcal{V}$ is the whole vocabulary. For the grounding truth of the training data, the generation part of the model learns only the one-hot value at each position. For text-level distillation, the guidance from the teachers' output applies a smoother distribution of the probability of words. The distillation brings naturalness and correctness for the dialogue generation.

4.2 Policy-level Guiding

We also expect that the universal model can acquire the dialogue policies of these teachers. In other words, we expect that the teacher models and the student model should have

Models	Restaurant		Hotel		Train		Attraction		Taxi		General	
	BLEU	ER	BLEU	ER	BLEU	ER	BLEU	ER	BLEU	ER	BLEU	ER
Teachers												
Universal teacher	16.5	**69.89**	14.1	52.52	22.3	**63.19**	13.1	58.96	15.7	48.03	19.8	-
Individual teachers	**20.5**	68.60	**16.4**	**56.43**	**23.1**	60.31	**16.6**	**67.65**	**17.7**	**86.68**	**23.0**	-
Students												
HRED(No teacher)	17.1	**54.82**	15.0	44.95	17.2	47.27	**16.8**	**71.78**	15.5	**76.64**	**22.7**	-
HRED-MTSS	**18.1**	50.89	**16.5**	**45.91**	**19.9**	**56.19**	16.3	66.82	**16.4**	64.85	19.9	-

Table 1: Performance of different teachers and different students in each domain. A universal multi-domain teacher model trains on the whole dataset and several individual teacher models train in each domain. ER: entity recall.

Domain	Number of Turns	
	Train	Test
Restaurant	13471	1571
Hotel	12943	1506
Train	10612	1735
Attraction	7054	1061
Taxi	2996	419
Hospital	593	0
Police	463	4
General	8646	1072

Table 2: Number of turns in each domain when MultiWOZ is split.

similar action vector if provided with similar input. We use the action $\mathbf{a}^{\mathrm{T}}$ from the teachers' policy output as the extra information to train the student's policy. For $\mathbf{a}^{\mathrm{T}}$ and $\mathbf{a}^{\mathrm{S}}$ are both in the form of latent vectors. In the training phase, we use mean squared error (MSE) loss to force the student to learn the policies of the teachers:

$$J_{\mathrm{KD}-\pi} = \sum_{i=0}^{k} (a_i^{\mathrm{T}} - a_i^{\mathrm{S}})^2, \tag{3}$$

We use both the ground truth (Section 3.3) and the teachers' guidance as the training target. We add the text-level distillation loss and the policy-level distillation loss to the loss of the ground truth. To adjust the effect of the teachers and balance the weights of the different losses, we apply a weight scalar α_1 to the text-level distillation loss and another weight scalar α_2 to the policy-level one. Finally, the combination training loss J_θ of the student model can be illustrated as:

$$J_\theta = J_{\mathrm{NLL}} + \alpha_1 J_{\mathrm{KD}} + \alpha_2 J_{\mathrm{KD}-\pi}, \tag{4}$$

then we train the student model to minimize the combination loss J_θ to implement the guiding of teacher models.

5 Experiments

In this section, we elaborate the experiment settings (Section 5.1), the baselines we use (Section 5.2), and the analysis of experimental results (Section 5.3).

5.1 Experiment Settings

Dataset We choose MultiWOZ (Budzianowski et al. 2018), a multi-domain human-human conversation corpus, as our dataset. The MultiWOZ dataset consists of dialogue turns in 7 domains, respectively including restaurant, hotel, attraction, taxi, train, hospital and police. The conversation in MultiWOZ aims at satisfying users' intents, and informs the necessary information the user needs about some entities. An episode of conversation contains around 14 turns of dialogues between the user and the system. Several episodes' topics are limited in one domain from beginning to the end turn, while others' are switching among the conversation in 2 to up 5 domains. In each domain, there are about 4 slots that the system can receive from the user and about 3 properties of the entity the system should provide to the user. For example, in a restaurant domain, the user can choose the area, the price range and the food type of a restaurant, and the information the system should offer about the restaurant includes the address, the reference number, the phone number and other essential properties.

To test the response quality of the models, we take a preprocessing on the dataset: we replace the names of the entities and their property values with placeholders. Then we manually generate the belief states and the database pointers, as the extra inputs of teachers, from the human labelled semantics. All the dialogue turns are split to 7 specific domains based on the domain tags, which are given by MultiWOZ dataset and are determined by entities in the dialogue turns.

Models	Multi-domain		
	BLEU	Inform	Success
Comparisons			
Seq2seq	16.7	65.7	44.4
HRED	17.5	70.7	60.9
Seq2seq + MDBT	13.1	69.3	30.0
Seq2seq + TRADE	13.2	65.9	34.6
HRED + MDBT	13.1	68.8	35.5
HRED + TRADE	13.7	70.8	41.8
HRED-MTSS(ours)	**18.7**	**77.5**	**64.9**
State-of-the-art models			
LaRL + TRADE	12.4	**79.5**	44.7
HDSA + TRADE	**20.1**	76.4	**65.9**
Models with manual states			
Seq2seq + Manual states	17.8	75.4	62.8
HRED + Manual states	19.3	75.2	66.2
HDSA + Manual states	**22.9**	**82.3**	**75.1**

Table 3: Performance on the multi-domain environment.

Models	Restaurant		Hotel		Train		Attraction	
	Inform	Success	Inform	Success	Inform	Success	Inform	Success
Seq2seq + TRADE	88.6	57.9	90.9	42.4	72.1	60.8	63.9	55.3
HRED + TRADE	**91.8**	74.4	81.7	50.5	76.2	62.6	76.8	65.4
HDSA + TRADE	78.5	68.6	**91.4**	**85.3**	81.4	80.4	**93.9**	**82.1**
HRED-MTSS(ours)	87.4	**81.2**	86.8	81.5	**85.1**	**83.4**	86.6	74.5

Table 4: Results on different domains

For the dialogue turns that don't belong to these 7 domains, they are included into a generic domain. In other words, we have 8 separate dialogue turn sets, each set corresponds to an individual domain. We train 8 individual teachers for each domain. Table 2 shows the number of training and testing turns in each domain after the dataset is split. Besides, following the pre-processing instruction of MultiWOZ, all dialogue turns are delexicalized, which means all the slot values are replaced with placeholders.

Experiment Settings We construct two vocabularies from the dataset, the input one and the output one. For the input vocabulary, we discard the words appear less than 5 times. About 1300 words remain in input vocabulary. For the output vocabulary, we limited the size to 500. We use two types of embeddings for the input and the output vocabularies. The embedding size is set to 50. The hidden layer size of LSTM layers in all involved models is set to 150. The teacher models are the Seq2seq architecture, the encoder and the decoder are 150 dimensions hidden layer of LSTM networks as well. For each teacher model, we trained it on its respective domain, and find the model which has the best entity matching recall rate as the guidance. For the student model, we use Adam optimizer, and the learning rate is 0.005. As for α_1 and α_2 in Equation 4, both α_1 and α_2 are set to 0.005 for balancing the guidance from the ground truth and the teacher models. To test the stability and get reliable results, we repeat each experiment setting 3 times and some of them for 5 times.

Training Strategies In the training phase of the teacher models, we found that the sub-dataset of some domains are limited. For instance, the sub-dataset of the police domain only accounts for 0.82% of all training data, which results in poor performance of these teacher models. To solve this problem, we use a warm-start strategy: we use a pre-trained model T_{all} trained on the whole the training dataset as the starts, and each teacher model is fine-tuned from T_{all}. This warm-up strategy ensures the domain-specific teachers have equal or higher performance than T_{all}.

Evaluation Metrics To measure the performance of different models, we use several examined metrics to evaluate the generated response.

1. BLEU: we calculate BLEU-4 (Papineni et al. 2002) scores to measure the similarity between the real response and the generated one.
2. Inform rate and Success rate: We use two metrics that are suggested by Budzianowski et al. (2018), as the estimations for the MultiWOZ dataset in the dialogue context to text task. Both the measurements are on the episode-level. The Inform rate indicates whether the dialogue system suggests suitable entities according to the user's intent in an episode. The Success rate illustrates if the system provides all the correct properties for the user requests after a success informing.
3. Entity Recall: Entity Recall (ER) measures the recall score of the entities between the generated response and the ground truth. ER is a turn-level metrics and used to evaluate the performances of the teachers.

5.2 Baselines

- **Seq2Seq**: the vanilla Seq2Seq model (Cho et al. 2014).
- **HRED**: the HRED architecture proposed in Sordoni et al. (2015).
- **Seq2Seq + MDBT**: the Seq2Seq model with the Multi-domain Belief Tracker (MDBT) (Ramadan, Budzianowski, and Gasic 2018) as the state tracking model.
- **Seq2Seq + TRADE**: the Seq2Seq model with the Transferable Dialogue State Generator (TRADE) (Wu et al. 2019) as its state tracker model.
- **HRED + MDBT**: the HRED model with MDBT as its state tracker model.
- **HRED + TRADE**: the HRED model with TRADE as its state tracker model.
- **LaRL + TRADE**: the Latent Action Reinforcement Learning (LaRL) (Zhao, Xie, and Eskénazi 2019) method with TRADE as its state tracker model.
- **HDSA + TRADE**: the Hierarchical Disentangled Self-Attention (HDSA) (Chen et al. 2019) model with TRADE as its state tracker model.
- **HRED-MTSS (Our model)**: the HRED student model training with a Multiple Teachers Single Student framework.
- **Seq2Seq/HRED/HDSA + Manual states** Those three comparisons use the same models mentioned above. Instead of the dialogue state extracted by model-based state tracker, we use the human-labelled dialogue states as the model input in the test setting. In a real dialogue situation, there is not human labelling in the user's text. So this setting can be considered an idealized setting to figure out the upper bound performance the models can reach.

Distill weights		Multi-domain		
α_1	α_2	**BLEU**	**Inform**	**Success**
0.01	0.005	17.0	71.7	63.5
0.005	0.01	**18.9**	73.6	61.2
0.005	0.005	18.7	**77.5**	**64.9**
0.0025	0.005	18.1	73.1	63.9
0.01	0	17.0	72.2	62.0
0.005	0	18.3	72.2	63.4
0	0.01	18.2	77.1	64.7
0	0.005	18.3	74.6	63.2
0	0	17.5	70.7	60.9

Table 5: Results of adopting different distillation strategies. The last column is the results of a model without distilling.

5.3 Experimental Results and Analysis

Results on a multi-domain environment The comparison between our model with the different baseline models is shown in Table 3. From the table, we can see that compared with the baselines such as the Seq2Seq or the HRED model, our model (HRED-MTSS) gets the best performance in the multi-domain settings. By adding a teacher-student framework, the informing rate and success rate receive 6.8% and 4.0% improvements respectively over the original HRED model. While compared with the state-of-the-art results achieved by HDSA or LaRL with the TRADE state tracker, HDSA+TRADE slightly outperforms our model in certain but not all metrics. We have to state that

- HDSA uses pre-trained models such as BERT (Devlin et al. 2019). However, BERT not only boosts its performance but also brings bloated model and high latency problems in real scenario deployments.
- LaRL uses the reinforcement learning method, which aims to maximize the long-term return, i.e., the Inform rate and the Success rate in the dialogue context. LaRL can achieve high scores in one aforementioned metrics but fail in the BLEU score and utterance fluency.

Additionally, in the setting of manual states, our model reaches equal or higher results than the Seq2seq and the HRED model. Adding an external state tracker to the Seq2Seq model and the HRED model increases the inform rate but has no help for the dialogue success rate.

Results on single domain environments As shown in Table 4, we also test our models' performance in 4 major single domains of MultiWOZ: restaurant, hotel, attraction and train. When compared with a Seq2Seq and HRED model, our model achieves the best success rate in all domains and outperforms in the attraction domain and train domain under the metrics of inform rate . We believe that it is due to the application of an individual teacher in each domain in the training phrase, which results in a better performance in this domain than the universal one. And compared with the HDSA model with the TRADE state tracker, our model is better in 2 of all 4 domains, the restaurant domain and the train domain.

Individual teachers' performances We compare the performance between different teachers, a universal multi-domain teacher trained on the whole dataset and the individual teachers trained on respective domains. Table 1 shows the experimental results of two kinds of teachers in 5 specific domains and 1 generic domain (The rest 2 domains lack testing data). From the table, we can see that for all domains, the individual teachers get higher BLEU scorers than the universal one. As for the entity matching recall metrics, the individual teachers perform better in 3 of all 5 specific domains. In the restaurant domain, the individual model gets the competitive result over the universal one. The universal model achieves higher entity recall rate than the individual teacher only in the train domain. Results show that the fine-tuned individual teachers significantly outperform the universal model most of the time, while the universal model gets slight advantages only in a few domains. We also compare the student's performance with the teachers' and a raw model. Experimental results show that the HRED model applied with MTSS framework, compared with the vanilla HRED model, achieves more satisfying performance in domains whose dataset size is large (The dataset size of first 5 domains is in a descending order from left to right in Table 1).

Effect of distillation weights From Table 5, we can see the results of using different guiding weights for text-level (α_1) and policy-level (α_2). Compared with the model without distillation ($\alpha_1 = 0, \alpha_2 = 0$), text-level distillation ($\alpha_1 \neq 0$, $\alpha_2 = 0$) and policy-level distillation ($\alpha_1 = 0$, $\alpha_2 \neq 0$) can bring improvements respectively. Besides, when applied with both distillation methods together with their weights $\alpha_1 = 0.005$ and $\alpha_2 = 0.005$, the model gets the highest performance in both the inform rate and the success. Both the two distillation methods help with the student model.

6 Conclusions

In this paper, we propose a novel approach to building a high-quality multi-domain dialogue system based on a teacher-student framework. We utilize multiple domain-specific teacher models to help a single student model become a multi-domain dialogue expert, which circumvent the knotty multi-domain dialogue state representation problem. To fully take advantage of the knowledge of the teacher models, we creatively make the teacher model impart their knowledge to the student in both text-level and policy-level. To discover the potential of the teacher-student framework, we would focus on adopting the framework to the SOTA dialogue models in our future work.

Acknowledgments

This work was supported by the NSFC (No. 61402403), Alibaba Group through Alibaba Innovative Research Program, Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Chinese Knowledge Center for Engineering Sciences and Technology, Engineering Research Center of Digital Library, Ministry of Education, and the Fundamental Research Funds for the Central Universities.

References

Budzianowski, P.; Wen, T.; Tseng, B.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gasic, M. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *EMNLP 2018*, 5016–5026. ACL.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *ACL 2019*, 3696–3709. ACL.

Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *EMNLP 2014*, 1724–1734. ACL.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *NAACL-HLT 2019*, 4171–4186. ACL.

Goel, R.; Paul, S.; Chung, T.; Lecomte, J.; Mandal, A.; and Hakkani-Tür, D. Z. 2018. Flexible and scalable state tracking framework for goal-oriented dialogue systems. *CoRR* abs/1811.12891.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *CoRR* abs/1503.02531.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Kim, Y., and Rush, A. M. 2016. Sequence-level knowledge distillation. In Su, J.; Carreras, X.; and Duh, K., eds., *EMNLP 2016*, 1317–1327. ACL.

Maes, S. H., and Gopalakrishnan, P. 2006. System and method for providing network coordinated conversational services. US Patent 7,003,463.

Maes, S. H. 2005. Conversational networking via transport, coding and control conversational protocols. US Patent 6,934,756.

Mrksic, N.; Séaghdha, D. Ó.; Wen, T.; Thomson, B.; and Young, S. J. 2017. Neural belief tracker: Data-driven dialogue state tracking. In Barzilay, R., and Kan, M., eds., *ACL 2017*, 1777–1788. ACL.

Pakucs, B. 2003. Towards dynamic multi-domain dialogue processing. In *EUROSPEECH 2003 - INTERSPEECH 2003*. ISCA.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, 311–318. ACL.

Ramadan, O.; Budzianowski, P.; and Gasic, M. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In Gurevych, I., and Miyao, Y., eds., *ACL 2018*, 432–437. ACL.

Rastogi, A.; Hakkani-Tür, D.; and Heck, L. P. 2017. Scalable multi-domain dialogue state tracking. In *ASRU 2017*, 561–568. IEEE.

Serban, I. V.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In Schuurmans, D., and Wellman, M. P., eds., *AAAI 2016*, 3776–3784. AAAI Press.

Serban, I. V.; Sordoni, A.; Lowe, R.; Charlin, L.; Pineau, J.; Courville, A. C.; and Bengio, Y. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In Singh, S. P., and Markovitch, S., eds., *AAAI 2017*, 3295–3301. AAAI Press.

Sordoni, A.; Bengio, Y.; Vahabi, H.; Lioma, C.; Simonsen, J. G.; and Nie, J. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In Bailey, J.; Moffat, A.; Aggarwal, C. C.; de Rijke, M.; Kumar, R.; Murdock, V.; Sellis, T. K.; and Yu, J. X., eds., *CIKM 2015*, 553–562. ACM.

Sun, K.; Chen, L.; Zhu, S.; and Yu, K. 2014. A generalized rule based tracker for dialogue state tracking. In *SLT 2014*, 330–335. IEEE.

Tan, X.; Ren, Y.; He, D.; Qin, T.; Zhao, Z.; and Liu, T. 2019. Multilingual neural machine translation with knowledge distillation. In *ICLR 2019*. OpenReview.net.

Ultes, S.; Rojas-Barahona, L. M.; Su, P.; Vandyke, D.; Kim, D.; Casanueva, I.; Budzianowski, P.; Mrksic, N.; Wen, T.; Gasic, M.; and Young, S. J. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In Bansal, M., and Ji, H., eds., *ACL 2017*, 73–78. ACL.

Wen, T.; Gasic, M.; Mrksic, N.; Rojas-Barahona, L. M.; Su, P.; Vandyke, D.; and Young, S. J. 2016. Multi-domain neural network language generation for spoken dialogue systems. In Knight, K.; Nenkova, A.; and Rambow, O., eds., *NAACL HLT 2016*, 120–129. ACL.

Wu, C.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; and Fung, P. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *ACL 2019*, 808–819. ACL.

Yao, K.; Zweig, G.; and Peng, B. 2015. Attention with intention for a neural network conversation model. *CoRR* abs/1510.08565.

Zhao, T.; Xie, K.; and Eskénazi, M. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In Burstein, J.; Doran, C.; and Solorio, T., eds., *NAACL-HLT 2019*, 1208–1218. ACL.

Zhong, V.; Xiong, C.; and Socher, R. 2018. Global-locally self-attentive dialogue state tracker. *CoRR* abs/1805.09655.