Deep Reinforcement Learning for On-line Dialogue State Tracking

Zhi Chen, Lu Chen, Xiang Zhou and Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Eng.
SpeechLab, Department of Computer Science and Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, China
{zhenchi713,chenlusz,owenzx,kai.yu}@sjtu.edu.cn

Abstract

Dialogue state tracking (DST) is a crucial module in dialogue management. It is usually cast as a supervised training problem, which is not convenient for on-line optimization. In this paper, a novel companion teaching based deep reinforcement learning (DRL) framework for on-line DST optimization is proposed. To the best of our knowledge, this is the first effort to optimize the DST module within DRL framework for on-line task-oriented spoken dialogue systems. In addition, dialogue policy can be further jointly updated. Experiments show that on-line DST optimization can effectively improve the dialogue manager performance while keeping the flexibility of using predefined policy. Joint training of both DST and policy can further improve the performance.

1 Introduction

A task-oriented spoken dialogue system usually consists of three modules: input, output and control, shown in Fig.1. The input module which consists of automatic speech recognition (ASR) and spoken language understanding (SLU) extracts semantic-level user dialogue actions from user speech signal. The control module (referred to as dialogue management) has two missions. One is to maintain dialogue state, an encoding of the machine's understanding about the conversation. Once the information from the input module is received, the dialogue state is updated by dialogue state tracking (DST). The other is to choose a semantic-level machine dialogue action to response the user, which is called dialogue decision policy. The output consists of natural language generation (NLG) and text-to-speech (TTS)
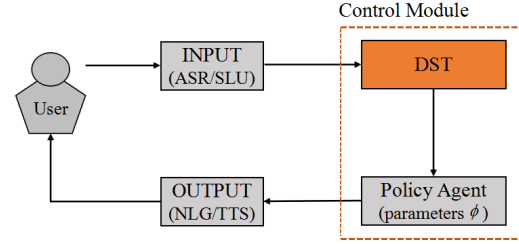

Figure 1: Spoken dialogue system.

synthesis, which convert dialogue action to audio. Dialogue management is an important part of a dialogue system. Nevertheless, there are inevitable ASR and SLU errors which make it hard to track true dialogue state and make decision. In recent statistical dialogue system, the distribution of dialogue state, i.e. *belief state*, is tracked. A well-founded theory for belief tracking and decision making is offered by partially observable Markov Decision Process (POMDP) (Kaelbling et al., 1998) framework.

Previous DST algorithms can be divided into three families: hand-crafted rules (Wang and Lemon, 2013; Sun et al., 2014a), generative models (Bui et al., 2009; Young et al., 2010), and discriminative models (Sun et al., 2014b; Henderson et al., 2014c). Recently, since the Dialog State Tracking Challenges (DSTCs) have provided labelled dialog state tracking data and a common evaluation framework and test-bed, a variety of machine learning methods for DST have been proposed. These methods rely strictly on set of labelled off-line data. Since the labelled data are off-line, the learning process of these supervised learning methods is independent on the dialogue policy module. The key issues of these supervised learning methods are poor generalization and over-tuning. Due to the lack of labels, these approaches can not be easily used for on-line update of DST.

This work marks first step towards employing

the deep reinforcement learning (DRL) method into dialogue state tracking (DST) module. The performance of the DST module is optimized during the conversation between the user and the dialogue system. We call the DRL-based DST module as the tracking agent. In order to bound the search space of the tracking agent, we propose a companion teaching framework (Chen et al., 2017). Furthermore, under this framework, we can train tracking agent and dialogue policy agent jointly with respective deep reinforcement learning (DRL) algorithms in order to make these two agents adaptive to each other.

The paper has two main contributions:

- The paper provides a flexible companion teaching framework which makes the DST be able to be optimized in the on-line dialogue system.
- We can jointly train DST agent and dialogue policy agent with different reinforcement learning algorithms.

The rest of the paper is organized as follows. Section 2 gives an overview of related work. In Section 3, the framework of on-line DST are presented. The implementation detail is represented in Section 4. In Section 5, the joint training process is introduced. Section 6 presents experiments conducted to evaluate the proposed framework, followed by the conclusion in Section 7.

2 Related Work

Recent mainstream studies on dialogue state tracking are discriminative statistical methods. Some of the approaches encode dialogue history in features to learn a simple classifier. Henderson et al. applies a deep neural network as a classifier. Williams proposed a ranking algorithm to construct conjunctions of features. The others of the approaches model dialogue as a sequential process (Williams et al., 2016), such as conditional random field (CRF) (Lee, 2013) and recurrent neural network (RNN) (Henderson et al., 2014c). All of these approaches need massive labelled in-domain data for training, they are belong to off-line and static methods.

In contrast to dialogue state tracking, the dialogue policy in task-oriented SDS has long been trained using deep reinforcement learning (DRL) which includes value function approximation methods, like deep Q-network (DQN) (Cuayáhuitl et al., 2015; Zhao and Eskenazi, 2016;

Lipton et al., 2016; Fatemi et al., 2016; Chen et al., 2017; Chang et al., 2017), and policy gradient methods, e.g. REINFORCE (Su et al., 2017; Williams et al., 2017), advantage actor-critic (A2C) (Fatemi et al., 2016). (Mnih et al., 2015) under POMDP framework. In our experiments, the dialogue policies of our provided systems are optimized by DQN. Our proposed framework is also inspired by the success of the companion teaching methods (Chen et al., 2017) in the dialogue policy.

In this work, we propose a companion teaching framework to generate the tracker from the on-line dialogue system. But the space of the belief state is continuous, it is difficult to be optimized by normal RL algorithms. Hausknecht and Stone provided an efficient method to extend deep reinforcement learning to the class of parameterized action space MDPs and extend the deep deterministic policy gradient (DDPG) algorithm (Lillicrap et al., 2015) for bounding the action space gradients suggested by the critic. This method greatly reduces the difficulty of the exploration.

The closest method is the Natural Actor and Belief Critic (NABC) algorithm (Jurčíček et al., 2012) which jointly optimizes both the tracker and the policy parameters. However, the tracker in (Jurčíček et al., 2012) uses a dynamic Bayesian Network to represent the dialogue state. In our work, the dialogue management system which includes the DST and the dialogue policy is purely statistical. Recently, Liu and Lane proposed an end-to-end dialogue management system, which directly connects the dialogue state tracking module and the dialogue decision policy module with the reinforcement learning method. There are many problems with this approach in the practical dialogue system, where end-to-end dialogue systems are not as flexible and scalable as the modular dialogue system.

3 On-line DST via Interaction

Discriminative machine learning approaches are now the state-of-the-art in DST. However, these methods have several limitations. Firstly, they are supervised learning (SL) approaches which require massive off-line data annotation. This is not only expensive but also infeasible for on-line learning. Secondly, given limited labelled data, over-tuning may easily happen for SL approaches, which leads to poor generalization. Thirdly, since
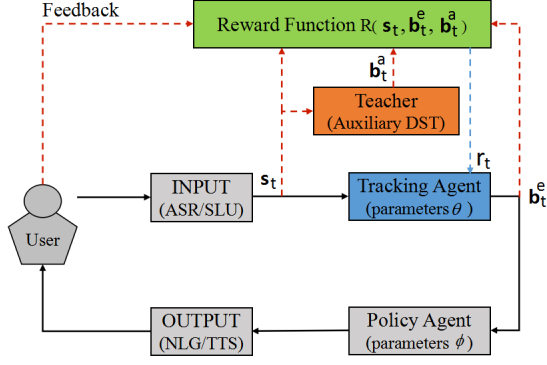
Figure 2: The companion teaching framework for the on-line DST.

SL-based DST approaches are independent of dialogue policy, the DST module can't dynamically adapt to the habit of the user. These limitations prohibit DST module from on-line update. To address this problem, we propose a deep reinforcement learning (DRL) framework for DST optimization via on-line interaction.

Reinforcement learning (RL) has been popular for updating the dialogue policy module in a task-oriented dialogue system for a long time. However, except for a few joint learning model for both DST and policy, RL has not been used specifically for the DST module. In this paper, under the RL framework, we regard the DST as an agent, referred to as *tracking agent*, and the other parts of the dialogue system as the environment. To our best knowledge, this is the first attempt to employ the reinforcement learning framework specifically for on-line DST optimization.

Different from the policy agent, the decision (belief state) made by the tracking agent is continuous. Hence, in this paper, DST is cast as a continuous control problem which has a similar challenge as robot control. There are several advanced algorithms to tackle the continuous control problems, e.g. DDPG algorithm. However, since the continuous belief state is both continuous and high-dimensional, the straightforward application of existing RL algorithms do not work well. In this paper, we borrow the *companion teaching* idea (Chen et al., 2017) to construct a novel RL framework for DST. Here, an auxiliary well-trained tracker, e.g. a traditional tracker trained off-line, is used as the *teacher* to guide the optimizing process of the actual DST agent (the *student*) to avoid over-tuning and achieve robust and fast convergence.

The companion teaching RL-DST framework is shown in Fig.2, where $\mathbf{b}^a$ is the auxiliary belief state produced by the auxiliary DST model and $\mathbf{b}^e$ is the exploration belief state produced by the tracking agent. The difference between $\mathbf{b}^a$ and $\mathbf{b}^e$ will be fed into the reward signal to significantly reduce the search space of the tracking agent.

It is also worth comparing the proposed framework with end-to-end dialogue systems which can also support on-line update. Firstly, the modular structure of the RL-DST framework allows more flexible and interpretable dialogue management models to be used. For example, interpretable dialogue policy, such as rule-based policy, can be easily used with arbitrary DST models. This flexibility is practically very useful. Secondly, due to the use of a teacher DST model, the optimizing process of the tracking agent requires few dialogue data and the training is more robust.

3.1 Input & Output

To avoid confusion with the concepts of the policy agent, we replace *input* and *output* for *state* and *action* of the tracking agent respectively. In this work, only semantic-level dialogue manager is considered. Thus, the input is semantic features of each slot, which are extracted from system action, spoken language understanding (SLU) output and context from the previous turn. The output of the tracking agent is belief state of the corresponding slot at the current turn. In contrast to the system action of the policy agent, the output of the tracking agent, i.e. belief state, is continuous. In this paper, the input of the tracking agent is represented as $\mathbf{s}$ and the output as $\mathbf{b}^e$.

3.2 Tracking Policy

The *tracking policy* denotes a mapping function between $\mathbf{s}$ and $\mathbf{b}^e$ which aims to maximize the expected accumulated reward. Since the search space of the tracking agent is continuous, deterministic reinforcement learning algorithms, such as DDPG algorithm, is used to optimize the tracking policy as in the robotic control problem (Lillicrap et al., 2015; Gu et al., 2017).

3.3 Reward Signal

Dialogue system reward is usually defined as a combination of **turn penalty** and **success reward** (Cuayáhuitl et al., 2015; Zhao and Eskenazi, 2016). The policy agent can be effectively optimized using the two reward signals. However, for the tracking agent, due to the large search space

caused by the continuous output, the two signals are not sufficient to achieve fast and robust convergence. To address this problem, we design another **basic score** reward signal to constrain the search space of the tracking agent. Therefore, the overall reward of the tracking agent consists of three kinds of signals:

Turn Penalty, denoted as $r^{\texttt{tp}}$, is a negative constant value to penalize long dialogues. The assumption here is that shorter dialogue is better.

Success Reward, denoted as $r^{\texttt{sr}}$, is a delayed reward for the whole dialogue at the last turn. When the conversation between the user and the machine is over, the user gives an evaluation value to judge the performance of the dialogue system. If the whole conversation has not achieved the user's goal, the success reward will be zero. Otherwise, success reward will be a positive constant value.

Basic Score, denoted as $r^{\texttt{bs}}$, is used to reduce the search space for tracking agent. As shown in Fig.2, an auxiliary DST is used. We use the auxiliary belief state $\mathbf{b}^{\texttt{a}}$ to guide the exploration of the tracking agent. If the exploration belief state $\mathbf{b}^{\texttt{e}}$ is far away from the auxiliary belief state, a penalty is given as in equation (1). Thus, basic score is inversely proportional to the distance between auxiliary belief state and exploration belief state.

$$r^{\texttt{bs}} = -\alpha ||\mathbf{b}^{\texttt{e}} - \mathbf{b}^{\texttt{a}}||_2$$

where $|| \cdot ||_2$ is L2 distance and $\alpha \geq 0$ is referred to as **trust factor**. With larger α, performance of the tracking agent is closer to the auxiliary DST model.

In the middle of a conversation, the immediate reward of the exploration belief state is $r^{\texttt{tp}} + r^{\texttt{bs}}$, the immediate reward of the last turn is $r^{\texttt{sr}}$.

4 Implementation Detail

In the companion teaching RL-DST framework, the auxiliary DST can make use of arbitrary well trained DST model and the tracking agent can be optimized by any deterministic reinforcement learning algorithm. In this section, we will introduce the dialogue tasks as well as the specific algorithm implementations, though the actual algorithms are not constrained to the below choices. Note that, the tracking agent, i.e. the DST to be optimized, takes a form of deep neural network in this paper.

In this work, we evaluate the proposed framework on the task-oriented dialogue systems in the restaurant/tourism domain in DSTC2/3 (Henderson et al., 2014a,b). These systems are $slot-based$ dialogue systems. There are three slot types: **goal constraint**, **request slots** and **search method**. The **goal constraint**s are constraints of the information/restaurant which the user is looking for. The **search methods** describe the way the user is trying to interact with the system. The **request slots** are demands which the user has requested. The three different types of slots have different influences on the dialogue performance. Therefore, we use multiple tracking agents, each agent per type, to represent dialogue tracking policy instead of only one overall tracking agent. Each agent has its own input and output. The final overall output is simply the concatenation of the outputs from all agents.

4.1 Auxiliary Polynomial Tracker

In this paper, a polynomial tracker is used as the auxiliary DST. It is also referred to as *Constrained Markov Bayesian Polynomial* (CMBP) (Yu et al., 2015) which is a hybrid model combining both data-driven and rule-based models. CMBP has small number of parameters and good generalization ability. In CMBP, the belief state at current turn is assumed to be dependent on the observations of the current turn and the belief state of the previous turn. A general form of CMBP is shown as below:

$$b_{t+1}(v) = \mathcal{P}(P_{t+1}^+(v), P_{t+1}^-(v), \tilde{P}_{t+1}^+(v), \tilde{P}_{t+1}^-(v), b_t^r, b_t(v))$$
$$\texttt{s.t.constraints} \tag{1}$$

where $\mathcal{P}(\cdot)$ is a polynomial function, $b_{t+1}(v)$ which denotes the probability of a specific slot taking value v at the $(t+1)^{th}$ turn is a scalar value and constraints include probabilistic constraints, intuition constraints and regularization constraints. And there are six probabilistic features for each v defined as below

- $P_{t+1}^+(v)$: sum of scores of SLU hypotheses informing or affirming value v at turn $t+1$

- $P_{t+1}^-(v)$: sum of scores of SLU hypotheses denying or negating value v at turn $t+1$

- $\tilde{P}_{t+1}^+(v) = \sum_{v' \notin \{v, None\}} P_{t+1}^+(v')$

- $\tilde{P}_{t+1}^-(v) = \sum_{v' \notin \{v, None\}} P_{t+1}^-(v')$

- b_t^r: probability of the value being 'None' (the value not mentioned) at turn t

- $b_t(v)$: belief of "the value being v at turn t"

In this paper, polynomial order is 3. The coefficients of polynomial $\mathcal{P}(\cdot)$ are optimized by the off-line pre-collected training data. Each slot type in DSTC2/3 (goal, request, method) has its own polynomial model, represented by $\mathcal{P}_g(\cdot)$, $\mathcal{P}_r(\cdot)$ and $\mathcal{P}_m(\cdot)$ respectively. The belief state of different slot-value pairs within the same slot type is updated by the same polynomial. For example, in our work, we set $\mathcal{P}_g(\cdot)$ as:

$$b_{t+1}(v) = (b_t(v) + P_{t+1}^+(v) * (1 - b_t(v))) *$$
$$(1 - P_{t+1}^-(v) - \tilde{P}_{t+1}^+(v))$$

An example of updating belief state of the slot *pricerange* using polynomial tracker is shown in Fig.3.

Figure 3: Example with polynomial dialogue state tracking.

4.2 Tracking Agents

Three types of the slots (goal, request, method) in DSTC2/3 are not to affect each other. Therefore, the DST tracking agent can be decomposed into three independent tracking agents for DSTC2/3 tasks, represented by **TA_G**, **TA_R** and **TA_M** in Fig.4. These tracking agents have individual RL components as described in section 3. The three tracking agents correspond to the auxiliary DST trackers $\mathcal{P}_g(\cdot)$, $\mathcal{P}_r(\cdot)$ and $\mathcal{P}_m(\cdot)$ respectively. Note that the forms of the DST tracking agents are deep neural networks instead of polynomials.

In this paper, the input of each tracking agent which represents as $\mathbf{s}^g$, $\mathbf{s}^r$ and $\mathbf{s}^m$ is consistent with the input of polynomial tracker represented in equation (1) where each slot is represented by six
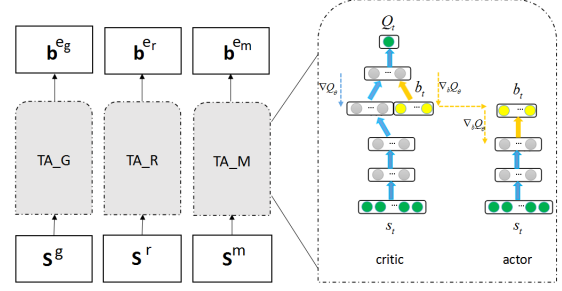

Figure 4: Multi-agent Tracking in DSTC2/3.

probabilistic features. The output of each tracking agent represented by $\mathbf{b}^{eg}$, $\mathbf{b}^{er}$ and $\mathbf{b}^{em}$ in Fig.4 is belief state of corresponding slots at the next turn. In this work, we adopt deep deterministic policy gradient (DDPG) algorithm to optimize these three tracking agents. The flexibility of our framework is that we can optimize the selective parts of DST module, the other parts of belief state can still be produced by the auxiliary polynomial DST. We can also figure out which parts of DST module have the bigger effect on the dialogue performance.

4.3 DDPG for Tracking Policy

In order to optimize three tracking agents in Fig.4 which have continuous and high dimensional output spaces, we use DDPG algorithm (Lillicrap et al., 2015) which is an actor-critic, model-free algorithm based on the deterministic policy gradient that can operate over continuous action spaces. This algorithm combines the actor-critic approach with insights from the DQN algorithm which has a replay buffer and adopts the soft-update strategy.

During training of the three agents, there are three experience memories for each tracking agent respectively. The format of the data in memories is $(\mathbf{s}_t, \mathbf{b}_t^e, r_t, \mathbf{s}_{t+1})$. $\mathbf{s}_t$ is slot feature vector and $\mathbf{b}_t^e$ is the exploration belief state of corresponding slots. The immediate reward r_t is produced by reward function $R(\mathbf{s}_t, \mathbf{b}_t^e, \mathbf{b}_t^a)$ at each turn, presented at Section 3.3.

The DDPG algorithm uses the deterministic policy gradient (DPG) method to update the deep neural networks. There are two functions in DDPG algorithm: the actor policy function $\pi(\mathbf{s}_t|\theta)$ deterministically maps the input to output, and the critic function $Q(\mathbf{s}_t, \mathbf{b}_t^e|\beta)$ is learned using the Bellman equation as in Q-learning which aims to minimize the following loss function,

$$L(\beta) = \mathbb{E}_{\mathbf{s}_t, \mathbf{b}_t^e, r_t}[(Q(\mathbf{s}_t, \mathbf{b}_t^e|\beta) - y_t)^2] \quad (2)$$

where $y_t = r_t + \lambda Q(\mathbf{s}_{t+1}, \pi(\mathbf{s}_{t+1}|\beta)), r_t)$ is immediate reward at t^{th} turn and $\lambda \in [0, 1]$ is discount factor.

The target of the actor policy is to maximize the cumulative discounted reward from the start state, denoted by the performance objective $J(\pi) = \mathbb{E}[\sum_{t=1}^{T} \gamma^{t-1} r_t | \pi]$. Silver et al. proved that the following equation is the actor policy gradient, the gradient of the policy' s performance:

$$\nabla_\theta J \approx \mathbb{E}_{\mathbf{s}_t}[\nabla_{\mathbf{b}^e} Q(\mathbf{s}, \mathbf{b}^e|\beta)|_{\mathbf{s}=\mathbf{s}_t, \mathbf{b}^e=\pi(\mathbf{s}_t)} \nabla_\theta \pi(\mathbf{s}|\theta)|_{\mathbf{s}=\mathbf{s}_t}] \tag{3}$$

where $\nabla_{\mathbf{b}^e} Q(\mathbf{s}, \mathbf{b}^e|\beta)$ denotes the gradient of the critic with respect to actions and $\nabla_\theta \pi(\mathbf{s}|\theta)$ is a Jacobian matrix such that each column is the gradient $\nabla_\theta[\pi(\mathbf{s}|\theta)]_d$ of the dth action dimension of the policy with respect to the policy parameters θ. The implementation details of the DDPG algorithm are provided in (Lillicrap et al., 2015).

5   Joint Training Process

In Section 4, we discuss the implementation details of the on-line DST in DSTC2/3 cases. During the learning process of the tracking agent, the dialogue policy is fixed and the tracker keeps changing. Since the DST is part of the environment for the dialogue policy agent, when the tracking agent is optimized, the environment of the dialogue policy agent is also changed. Thus, we can choose to further optimize the dialogue policy in order to get even more improved dialogue system performance. This is referred to as *joint training* of DST and policy. The process of joint training consists of four phases: the pre-training of the dialogue policy agent, the pre-training of the tracking agent, the training of the dialogue policy agent and the training of the tracking agent. The details of the joint training shows in algorithm 1.

6   Experiments

Two objectives are set for the experiments: (1) Verifying the performance of the SDS with the optimized on-line DST. (2) Verifying the performance of the dialogue system which jointly train the DST and the dialogue policy.

6.1   Dataset

The proposed framework is evaluated on domains of DSTC2/3 (Henderson et al., 2014a,b). In

---

Algorithm 1 The process of joint training

1: Initialize dialogue policy $Q(\phi)$, **TA_G** tracking agent $Q(\beta^g), \pi(\theta^g)$, **TA_R** tracking agent $Q(\beta^r), \pi(\theta^r)$, **TA_M** tracking agent $Q(\beta^m), \pi(\theta^m)$
    // *pre-train dialogue policy agent*
2: Set polynomial method as the tracker of the system
3: **for** $episode = 1 : N_1$ **do**
4:    Update dialogue policy using DQN algorithm
5: **end for**
    // *pre-train tracking agents*
6: **for** $episode = 1 : N_2$ **do**
7:    Update the actors $\pi(\theta^g)$, $\pi(\theta^r)$, $\pi(\theta^m)$ of the tracking agents by minimizing mean squared error with the output of polynomial tracker $\mathbf{b}^{a_g}$, $\mathbf{b}^{a_r}$ and $\mathbf{b}^{a_m}$.
8: **end for**
    // *optimize tracking agents*
9: Set multi-tracking agent as the tracker of the system
10: **for** $episode = 1 : N_3$ **do**
11:    Update the critics $Q(\beta^g)$, $Q(\beta^r)$, $Q(\beta^m)$ of the multi-tracking agent by minimizing equation (2)
12:    Update the actors $\pi(\theta^g)$, $\pi(\theta^r)$, $\pi(\theta^m)$ of the multi-tracking agent by equation (3)
13: **end for**
    // *optimize dialogue policy agent*
14: **for** $episode = 1 : N_4$ **do**
15:    Update dialogue policy using DQN algorithm
16: **end for**

DSTC2, there are 8 requestable slots and 4 informable slots. In DSTC3, there are 12 requestable slot and 8 informable slots. Therefore the task in DSTC3 is more complex. Furthermore, the semantic error rate in DSTC3 is higher than the semantic error rate in DSTC2.

Based on the datasets in DSTC2/3, an agenda-based user simulator (Schatzmann et al., 2007a) with error model (Schatzmann et al., 2007b) was implemented to emulate the behaviour of the human user and errors from the input module.

6.2   Systems

In our experiments, six spoken dialogue systems with different DST models were compared:

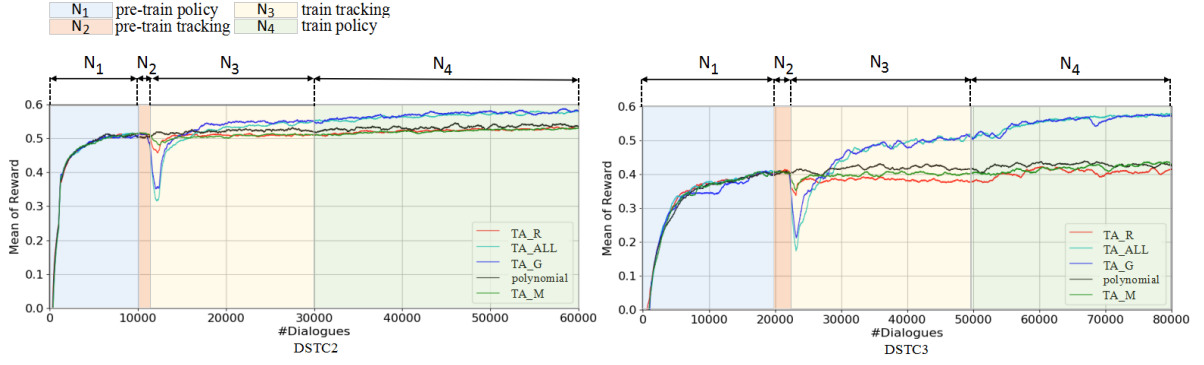- **Polynomial** is the baseline system. The *poly-*

Figure 5: The learning curves of joint training dialogue systems and baseline system in DSTC2 (left) and DSTC3 (right).

nomial DST as described in section 4.1 is used. The corresponding policy agent is a two-layer DQN network with 128 nodes per layer.

- **TA_G** is a DST tracking agent as in Fig.4. It only estimates the belief state of **goal constraint** and the other two parts of belief state are produced by the polynomial tracker.

- **TA_R** is a DST tracking agent as in Fig.4. It only estimates the belief state of **request slots** and the other two parts of belief state are produced by the polynomial tracker.

- **TA_M** is a DST tracking agent as in Fig.4. It only estimates the belief state of **search method** and the other two parts of belief state are produced by the polynomial tracker.

- **TA_ALL** is a DST tracking agent as in Fig.4. Here, the whole belief state is directly produced by the above three tracking agents.

- **TA_noteaching** is similar to **TA_ALL** except the **basic score** reward signal is not used. This is equivalent to directly on-line directly train a neural network DST tracker.

In traditional supervised-learning based DST approaches, metrics such as accuracy or L2 norm are used for evaluation. However, on-line DST optimization does not require semantic annotation and the optimization objective is to improve dialogue performance. Hence, in this paper, metrics for dialogue performance are employed to evaluate on-line DST performances. There are two metrics used for evaluating the dialogue system performance: average length and success rate. For the reward in section 3.3, at each turn, the turn penalty is -0.05 and the dialogue success reward is 1. The summation of the two rewards are used for evaluation, hence, the reward in below experiment tables are between 0 and 1. The trust factors of the

basic score in **TA_G**, **TA_R** and **TA_M** tracking agents are 0.2, 0.2, 4 in DSTC2 and 0.07, 0.07, 4 in DSTC3. For each set-up, the moving reward and dialogue success rate are recorded with a window size of 1000. The final results are the average of 25 runs.

6.3 DRL-based DST Evaluation

In this subsection, we evaluate the performances of the systems with five different on-line DST models (**TA_G**, **TA_R**, **TA_M**, **TA_ALL** and **TA_noteaching**). The dialogue policy agents of these five systems are optimized by DQN for N_1 (10000/20000 in DSTC2/3) episodes/dialogues with the same polynomial trackers. Next, these five systems start to train tracking agents. In the first N_2 (1000 in DSTC2/3) episodes, we pre-train *actor* part of DDPG with the output of polynomial tracker using mean squared error (MSE) in all tracking agents. After pre-training, tracking agents of these five systems are optimized by DDPG for N_3 (19000/29000 in DSTC2/3) episodes. In the *polynomial* SDS, the dialogue policy agent is optimized for $N_1 + N_2 + N_3$ episodes.

In Fig.5, after the tracking agents are optimized for almost 10000 episodes, the tracking agents in these four on-line DST systems achieve the convergence nearly in DSTC2/3. It demonstrates that the companion teaching framework for the on-line DST is efficient. In Table 1 and Table 2 [1], the tracking agents in the **TA_ALL** system and the **TA_G** system improve the performances of the SDS significantly in DSTC2 and DSTC3. The tracking agents in the **TA_ALL** and the **TA_G**

---

[1]Because of the limitation of space, the table only shows the standard deviation of the reward in this work, the standard deviations of the success rate and the turn can be reflected by the standard deviation

DST	Success	#Turn	Reward
Polynomial	0.769	5.013	0.519 ± 0.016
TA_ALL	**0.775**	4.474	**0.551** ± 0.018
TA_G	0.767	**4.375**	0.548 ± 0.020
TA_R	0.763	5.057	0.510 ± 0.022
TA_M	0.765	5.121	0.509 ± 0.018
TA_noteaching	−	−	−

Table 1: The performances of tracking agents in DSTC2. The symbol '-' means the dialogue system crashed.

DST	Success	#Turn	Reward
Polynomial	**0.744**	6.566	0.415 ± 0.077
TA_ALL	0.713	**4.117**	**0.507** ± 0.083
TA_G	0.719	4.290	**0.505** ± 0.075
TA_R	0.701	6.438	0.379 ± 0.028
TA_M	0.731	6.540	0.404 ± 0.021
TA_noteaching	−	−	−

Table 2: The performances of tracking agents in DSTC3. The symbol '-' means the dialogue system crashed.

DST	Success	#Turn	Reward
Polynomial	0.784	4.995	0.535 ± 0.015
TA_ALL	**0.810**	4.566	**0.581** ± 0.022
TA_G	0.805	**4.497**	**0.580** ± 0.015
TA_R	0.782	5.052	0.530 ± 0.014
TA_M	0.782	5.051	0.530 ± 0.020

Table 3: The performances of joint training in DSTC2.

DST	Success	#Turn	Reward
Polynomial	0.754	6.580	0.425 ± 0.071
TA_ALL	0.795	**4.317**	**0.578** ± 0.064
TA_G	**0.800**	4.579	0.571 ± 0.068
TA_R	0.747	6.654	0.414 ± 0.069
TA_M	0.759	6.605	0.429 ± 0.022

Table 4: The performances of joint training in DSTC3.

learned a tracking policy which can track the goal of the user accurately. Thus, compared with the *polynomial* system, the length of dialogue in these two systems decrease sharply. The rewards in these two systems increase significantly. The performances of the **TA_R** system and the **TA_M** system are similar with *polynomial* system. We can conclude that **goal constraint** plays a more important role in dialogue state than **request slots** and **search method**. The **TA_noteaching** system crashed during the optimizing process of the tracking agents. It reflects the effectiveness of our proposed companion teaching framework.

6.4 Joint Training Evaluation

In this subsection, we evaluate the performances of the systems (except for the **TA_noteaching** system) which jointly train dialogue policy agent and tracking agent. In the first $N_1 + N_2 + N_3$ episodes, training processes of five models have been mentioned in Section 6.3. As shown in Fig.5, in the latter N_4 (30000 in DSTC2/3) episodes, four models which contain tracking agents stop optimizing corresponding tracking agents and start to optimize dialogue policy agent and the baseline system continues to train dialogue policy agent. In Fig.5, we compare the above five systems and the final performances show in Table 3 and Table 4. Compared with the results of the optimized tracking agents in Table 1 and Table 2, the success rates in the **TA_ALL** system and the **TA_G** system increase significantly. It demonstrates that the

dialogue policies in the **TA_ALL** and the **TA_G** have adapted the optimized tracking agents respectively.

Compared the results in DSTC3 with the results in DSTC2, we can find that the boost of performance in DSTC3 is larger than that in DSTC2. The reason is that the semantic error rate of SLU in DSTC3 is higher than that in DSTC2, therefore the belief state tracker plays a more important role in DSTC3. These results also indicate that our proposed DRL-based tracker is robust to the input errors of SDS.

7 Conclusion

This paper provides a DRL-based companion teaching framework to optimize the DST module of the dialogue system. Under this framework, the tracker can be learned during the conversations between the user and the SDS rather than produced by the off-line methods. We can also choose to jointly train dialogue policy agent and the tracking agent under this framework. The experiments showed that the proposed companion teaching framework for the on-line DST system achieved promising performances in DSTC2 and DSTC3.

References

Trung H Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. 2009. A tractable hybrid ddn–pomdp approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering* 15(2):273–307.

Cheng Chang, Runzhe Yang, Lu Chen, Xiang Zhou, and Kai Yu. 2017. Affordable on-line dialogue

policy learning. In *Proceedings of EMNLP*. pages 2200–2209.

Lu Chen, Runzhe Yang, Cheng Chang, Zihao Ye, Xiang Zhou, and Kai Yu. 2017. On-line dialogue policy learning with companion teaching. *Proceedings of EACL* page 198.

Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. 2015. Strategic dialogue management via deep reinforcement learning. *NIPS DRL Workshop* .

Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. In *Proceedings of SIGDIAL*. pages 101–110.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *Proceedings of ICRA*. IEEE, pages 3389–3396.

Matthew Hausknecht and Peter Stone. 2015. Deep reinforcement learning in parameterized action space. *arXiv preprint arXiv:1511.04143* .

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of SIGDIAL*. pages 263–272.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014b. The third dialog state tracking challenge. In *SLT Workshop*. IEEE, pages 324–329.

Matthew Henderson, Blaise Thomson, and Steve Young. 2013. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of SIGDIAL*. pages 467–471.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014c. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of SIGDIAL*. pages 292–299.

Filip Jurčíček, Blaise Thomson, and Steve Young. 2012. Reinforcement learning for parameter estimation in statistical spoken dialogue systems. *Computer Speech & Language* 26(3):168–192.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2):99–134.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of SIGDIAL*. pages 442–451.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* .

Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. 2016. Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking. *arXiv preprint arXiv:1608.05081* .

Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. *arXiv preprint arXiv:1708.05956* .

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007a. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Proceedings of NAACL*. ACL, Morristown, NJ, USA, pages 149–152.

Jost Schatzmann, Blaise Thomson, and Steve Young. 2007b. Error simulation for training statistical dialogue systems. In *ASRU*. IEEE, pages 526–531.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *Proceedings of ICML*. pages 387–395.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of SIGDIAL*. pages 147–157.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014a. A generalized rule based tracker for dialogue state tracking. In *SLT Workshop*. IEEE, pages 330–335.

Kai Sun, Lu Chen, Su Zhu, and Kai Yu. 2014b. The sjtu system for dialog state tracking challenge 2. In *Proceedings of SIGDIAL*. pages 318–326.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of SIGDIAL*. pages 423–432.

Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse* 7(3):4–33.

Jason D Williams. 2014. Web-style ranking and slu combination for dialog state tracking. In *Proceedings of SIGDIAL*. pages 282–291.

Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of ACL*. volume 1, pages 665–677.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language* 24(2):150–174.

Kai Yu, Kai Sun, Lu Chen, and Su Zhu. 2015. Constrained markov bayesian polynomial for efficient dialogue state tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23(12):2177–2188.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *Proceedings of SIGDIAL*. pages 1–10.