Scheduled Dialog Policy Learning: An Automatic Curriculum Learning Framework for Task-oriented Dialog System

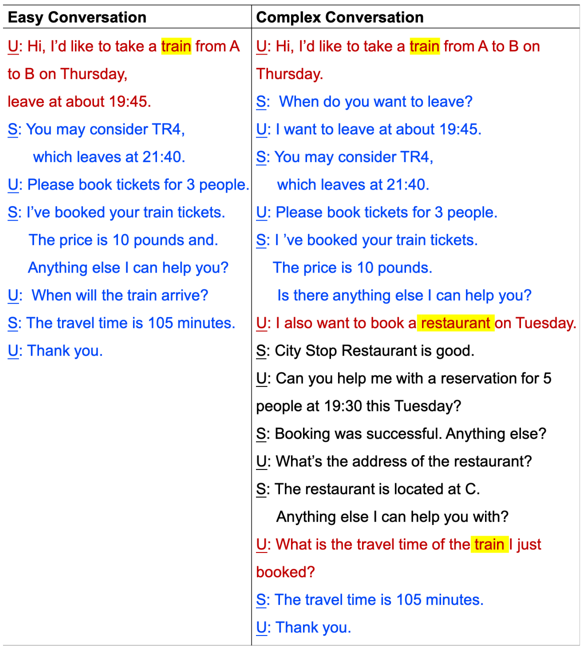
Sihong Liu[1]*, Jinchao Zhang[2], Keqing He[1], Weiran Xu[1]†, Jie Zhou[2]
[1] Beijing University of Posts and Telecommunications, China
[2] Pattern Recognition Center, WeChat AI, Tencent Inc, China
{liusihong,kqin,xuweiran}@bupt.edu.cn
{dayerzhang,withtomzhou}@tencent.com

Abstract

In reinforcement learning (RL) based task-oriented dialogue systems, users act as the environment and the agent learns the policy by interacting with users. However, due to the subjectivity of different users, the complexity of user-generated training conversations varies greatly, which leads to different difficulties for the agent to learn. Therefore, it is necessary for modeling dialogue complexity and make a reasonable learning schedule for efficiently training the agent. Towards that, we propose Scheduled Dialog Policy Learning, an automatic curriculum learning framework for jointing curriculum learning and policy optimization in the task-oriented dialog system. To our best knowledge, it is the first RL framework that improves dialogue policy learning by scheduling its learning process. Specifically, we introduce an automatic measurement to evaluate the dialogue complexity, and based on this automatic measurement, we train the dialog agent from easy dialogues to complex ones. Experiments demonstrate that our approach can be applied to the task-oriented dialogue policy learning and outperforms the previous state-of-the-art model, which increases 9.6% and 10.0% in the accuracy on the dialog success rate, respectively on the MultiWoz and Movie-Ticket Booking datasets.

1 Introduction

Dialog policy learning is an important component of the task-oriented dialogue system, and it determines the agent dialog action responding to the user. This learning process is often formulated as a reinforcement learning problem (Young et al., 2013; Levin et al., 1997; Dhingra et al., 2017; Li et al., 2017; Liu and Lane, 2017; Peng et al., 2018b; Su et al., 2018; Gao et al., 2019; Takanobu et al., 2020), where users act as the environment and the agent learns the policy by interacting with users. Thus, the learning performance of the dialogue policy depends much on users' behaviors.



Easy Conversation	Complex Conversation
U: Hi, I'd like to take a train from A to B on Thursday, leave at about 19:45. S: You may consider TR4, which leaves at 21:40. U: Please book tickets for 3 people. S: I've booked your train tickets. The price is 10 pounds and. Anything else I can help you? U: When will the train arrive? S: The travel time is 105 minutes. U: Thank you.	U: Hi, I'd like to take a train from A to B on Thursday. S: When do you want to leave? U: I want to leave at about 19:45. S: You may consider TR4, which leaves at 21:40. U: Please book tickets for 3 people. S: I 've booked your train tickets. The price is 10 pounds. Is there anything else I can help you? U: I also want to book a restaurant on Tuesday. S: City Stop Restaurant is good. U: Can you help me with a reservation for 5 people at 19:30 this Tuesday? S: Booking was successful. Anything else? U: What's the address of the restaurant? S: The restaurant is located at C. Anything else I can help you with? U: What is the travel time of the train I just booked? S: The travel time is 105 minutes. U: Thank you.

Figure 1: The comparison of the easy conversation (left) and the complex conversation (right) which are task-oriented dialogs between the user (U) and the system (S) sampling from MultiWoz. Comparing to the left conversation, the right one has more turns, intents, slots, and also switches between two domains: *train* (marked as blue) and *restaurant* (marked as dark). The right instance is apparently more complex.

However, due to the subjectivity and open-ended nature of human conversations, the complexity of training dialogues with different users varies greatly (Lison and Bibauw, 2017). Figure 1 shows dialogues with different complexities from MultiWoz (Budzianowski et al., 2018) dataset. Comparing to the left instance, the conversation in the right column has more turns, intents, slots, and also has the switch between two domains: *train*

[*] This work was done when Sihong Liu was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.

[†] Corresponding author

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1091–1102
August 1–6, 2021. ©2021 Association for Computational Linguistics

(marked as blue) and *restaurant* (marked as dark). The right instance is apparently more complex. As different complexity conversations lead to different difficulties for the agent to learn, in this paper, we introduce an automatic curriculum learning framework to improve dialog policy learning by scheduling its learning process where the agent first learns from easy conversations and then gradually manages more complicated ones. There is some related work that examines the curriculum learning in natural language processing (Kocmi and Bojar, 2017; Platanios et al., 2019; Sachan and Xing, 2016; Guo et al., 2019; Cai et al., 2020). Evaluation of training samples complexity is a challenging obstacle when organizing a curriculum. Previous work evaluates training samples mainly applying empirical and heuristic attributes, such as Platanios et al. (2019) defines the difficulty for training sentences concerning the sentence length and word rarity in the neural machine translation. Cai et al. (2020) evaluates the dialogue difficulty by five heuristic conversational attributes.

As the subjectivity and diversity of the conversation task, modeling its complexity by empirical and heuristic attributes is insufficient and relies on prior knowledge on the specific domain or dataset. Inspired by curiosity rewards (Schmidhuber, 1991; Stadie et al., 2015; Sorg et al., 2010; Pathak et al., 2017a), which applied state prediction error to predict uncertainty (Houthooft et al., 2016; Still and Precup, 2012; Wesselmann et al., 2019; Tegho et al., 2018) and improvement (Lopes et al., 2012) in RL tasks, in this paper, we further introduce an automatic measurement to evaluate dialogue complexity by estimating the dialog state differential space without applying any prior knowledge on the domain or dataset. Then we schedule the policy learning curriculum for the dialog agent based on the evaluation. To evaluate the effectiveness of our model for dialogue policy learning, we conduct our experiments on two public task-oriented dialog datasets: MultiWoz (Budzianowski et al., 2018) and Movie-Ticket Booking (Li et al., 2018). Experimental results show that our model reaches 83.3% and 57.0% in the accuracy for dialog successful rate on these two datasets, outperforming the previous state-of-the-art dialog model by 9.6% and 10.0% , respectively. Our main contributions in this work are three-fold:

- We design an automatic measurement to evaluate training dialogues complexity without any prior knowledge on the domain or dataset, which gets rid of empirical and heuristic attributes to model the dialogue complexity.
- Based on the automatic complexity measurement, we propose an automatic curriculum learning framework SDPL to improve the performance and learning efficiency of task-oriented dialogue policy learning.
- We conduct experiments on two task-oriented dialog corpus and results show the superiority of our model to the state-of-the-art baselines. Especially, it increases 9.6% and 10.0% in accuracy on dialog successful rate, respectively.

2 Related work

2.1 RL-based Task-oriented Dialog Policy Learning

Learning policies for the task-completion dialogue is often formulated as a reinforcement learning (RL) problem (Levin et al., 1997; Young et al., 2013; Li et al., 2017; Peng et al., 2018a,b; Su et al., 2018; Gao et al., 2019; Li et al., 2017; Peng et al., 2018b; Su et al., 2018; Takanobu et al., 2020; Li et al., 2020) and this learning process can be regard as a Markov Decision Process (MDP), where the agent interacts with a user through a sequence of actions to accomplish a pre-defined user goal.

Since reinforcement learning requires much interaction for training, a user simulator is often applied to interact with the agent providing the simulated user response in each dialogue turn based on a user goal (Li et al., 2017; Liu and Lane, 2017; Peng et al., 2018a; Su et al., 2018). Specifically, given a user goal G, at each time step t, the agent observes the current dialogue state s_t, and chooses an action a to execute, using the policy $\pi(a|s_t)$. Then the user responds the user action a_u which is sampled from the user goal G. Next, the agent receives reward r_t, observes the response a_u and updates to next state s_{t+1}. And the agent learns and updates the policy aiming to maximize its total discounted rewards. As the entire dialogue is around a user goal and a user goal corresponds to several user-generated dialogues, thus in this paper, we make the complexity evaluation of user goals for curriculum learning feeding different complexity dialogues generated by different complexity user goals to schedule the dialog agent training [1].

[1] Refer to the appendix for details on the user simulator

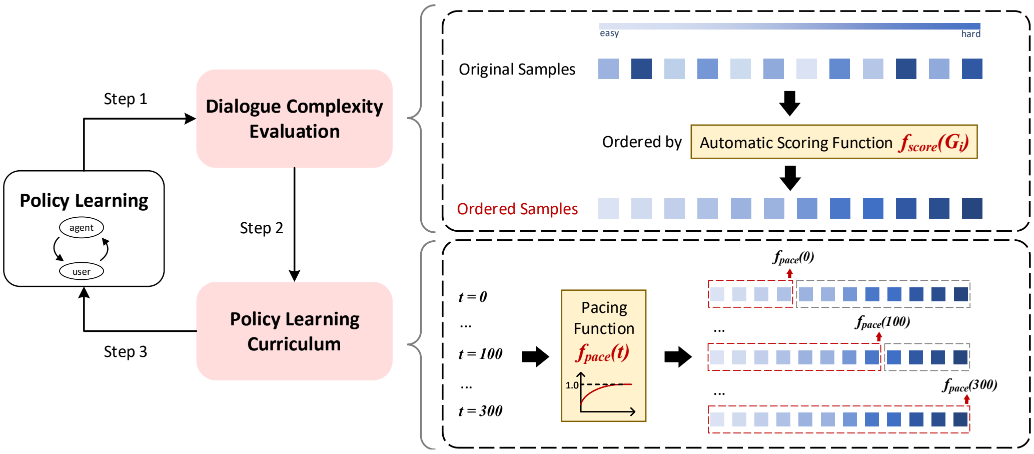


Figure 2: Illustration of proposed curriculum learning framework SDPL. It is defined by two stages. At **dialogue complexity evaluation stage**, an automatic scoring function $f_{score}(G_i)$ defines the instances G_i complexity (darker/lighter blue indicate easy/hard complexity) from easy to hard. Then at the **policy learning curriculum stage**, a pacing function $f_{pace}(t)$ schedules the percentage of training instances available for feeding to the policy learning according to the training step t. The dialog agent interacts with the user to learn the dialog policy on the scheduled instances and updates its learning capacity to execute a new dialog complexity evaluation.

2.2 Curriculum Learning

Curriculum learning (Bengio et al., 2009) is a learning strategy in machine learning, which learns from easy instances and then gradually handles harder ones. Curriculum learning has been used to natural language generation (Cai et al., 2020; Liu et al., 2018) and question answering task (Sachan and Xing, 2016). Cai et al. (2020) utilizes heuristic dialogue attributes to represent the dialogue complexity and propose several curricula, which achieve high performance. And Sachan and Xing (2016) proposes several heuristic strategies to model different complexity QA pairs. As the conversation task is more complex and analytic, in this paper, we propose an automatic curriculum framework to measure dialogue complexity automatically which can get rid of empirical and heuristic attributes and schedule different complexity training dialogues to improve the policy learning.

2.3 Curiosity Rewards

For many RL applications, state error prediction is used for curiosity rewards (Schmidhuber, 1991; Stadie et al., 2015; Sorg et al., 2010; Pathak et al., 2017a) [8, 9, 10, 5] to predict uncertainty (Houthooft et al., 2016; Still and Precup, 2012; Wesselmann et al., 2019; Tegho et al., 2018) and improvement (Lopes et al., 2012), improving the state exploration efficiency. And curiosity rewards are applied to tackle reward sparseness problem and state exploration in many tasks. Especially, Wesselmann et al. (2019) applied the curiosity rewards to the dialog policy to replace random exploration and stabilize training. While in this paper, we applied the curiosity reward as the measurement to evaluate the dialogue complexity.

3 Scheduled Dialog Policy Learning

We propose Scheduled Dialog Policy Learning (SDPL), a flexible and practical framework on joint curriculum learning and policy optimization for task-oriented dialog systems.

3.1 Overview

The overview of the full model is depicted in Figure 2, it schedules and designs the curriculum for dialogue policy learning in mainly two stages:

1) **Dialog Complexity Evaluation.** For the input training instances, we firstly measure original dialog instances complexity by an automatic scoring function, and arrange to sort these instances by their complexity scores, obtaining the ordered training instances. 2) **Policy Learning Curriculum.** After obtaining the ordered training instances, we apply a pacing function to schedule the percentage of instances available for training at each training step. Then these scheduled percentage of instances are fed to the dialogue agent to learn the dialog policy from easy to complex. As the training progressing, the agent updates its learning capacity and executes a new dialog complexity evaluation.

and the user goal.

In the subsequent subsection, we systematically describe how to utilize the SDPL for dialogue policy learning. In Section 3.2, we propose an automatic dialogue complexity evaluation approach. Then, we show in Section 3.3 how the automatic curriculum learning framework can contribute to the dialogue policy learning by automatically modeling dialogue complexity and making a reasonable learning schedule for training the dialog agent.

3.2 Dialogue Complexity Evaluation

The complexity of different training dialogues varies greatly which leads to different difficulties for the agent to learn. Therefore, it is necessary for modeling dialogue complexity and scheduling different complexity dialogues for efficiently training. In this paper, we propose a novel measurement to evaluate the dialogue complexity automatically. Inspired by Pathak et al. (2017b), for the agent instant learning capacity, we estimate the dialog state differential space that the agent can explore for current training samples as their training complexity. Less differential space means less error between the real state and predicted state which indicates current agent masters these samples where they are easy to learn for the current learning capacity.

Specifically, during the RL-based dialog policy learning process, in each step t, the agent observes current dialogue state s_t, then it executes the dialog action a_t estimating from current policy learning network. The agent then receives reward r_t [2] and updates to the next state s_{t+1}. Finally, the agent aims to maximize the expected sum of rewards which can be formulated as:

$$\underset{\theta_Q}{max} E_{\pi(s_t,a_t;\theta_Q)}[\sum_t r_t], \tag{1}$$

where $\pi(s_t, a_t; \theta_Q)$ is the policy learning network parameterized by θ_Q. To estimate the state differential space between the real state and predicted state, we first encode the current state and real next state into state feature vectors. Next, we apply a feature predicting neural network to obtain the predicted next state feature encoding which is formulated as:

$$\hat{\phi}(s_{t+1}) = g\Big(\phi(s_t), a_t; \theta_m\Big), \tag{2}$$

where $\phi(\cdot)$ is a state encoding network which transforms the one-hot state variables into a feature space suitable for learning. And $\hat{\phi}(s_{t+1})$ is the predicted next state feature of the real next state feature $\phi(s_{t+1})$. We optimize parameters θ_m by minimizing the mean square error between them:

$$m\Big(\phi(s_{t+1}), \hat{\phi}(s_{t+1})\Big) = \eta \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|^2, \tag{3}$$

where $\eta > 0$ is a scaling factor. The function $m(t)$ calculates feature encoding differentials of states between the real next state feature and estimated next state feature in the latent feature space, which represents the state differential space that agent needs explore. Less state differential space means less error between the real and predicted state feature which indicates that current agent masters these states are easy to learn for the current agent learning capacity. Therefore, we apply the $m(t)$ as the complexity evaluation for the current agent towards current training samples.

We jointly optimize the reinforcement learning process and complexity evaluation as:

$$\underset{\theta_Q}{max} E_{\pi(s_t,a_t;\theta_Q,\theta_m)}[\sum_t r_t + \sum_t m(t)]. \tag{4}$$

We further formulate the complexity evaluation into the task-oriented dialogue system to measure the dialogue complexity (lines 3-8 in Algorithm 1). In the dialogue system, the user and system interact with each other in a dialog session to fulfill the user goal G_i, where $G_i \in \mathcal{D}_{total}$ and $\mathcal{D}_{total}$ is a user goal set, i is the user goal index. Each user goal G_i corresponds to its K generated dialog sessions $\{\tau_1, ..., \tau_k\}$. And during the reinforcement learning process, each dialog session τ_k can be seen as a trajectory of state-action tuples $\{(s_t, a_t, r_t, s_{t+1}), ...\}$, $t \in \{0, 1, ..., N-1\}$, where N stands for the number of turns in each dialogue session. Therefore, based on the Eq. 3, for each dialog τ_k, we obtain its complexity score by an automatic scoring function which can be formulated as:

$$f_{score}(m_{\tau_k}) = \sum_{t=1}^{N} m_{\tau_k}(t). \tag{5}$$

During the RL-based dialog policy learning, as training dialogues are user-generated based on the user goal, we also evaluate the complexity of user goals for later policy curriculum learning. For a given user goal G_i, we obtain its complexity score

[2] In the RL based dialogue system, reward measures the degree of whether the dialogue is successful. In our experiment, the agent obtains a reward of 80 when it success, and obtained a reward of -40 when failed, and the agent receives the reward of -1 at each turn to encourage shorter dialogues.

Algorithm 1 Scheduled Dialogue Policy Learning

```
Require: Dialog user goal set $\mathcal{D}_{total}$ with each user goal $\{G_i\}$, $N$
Ensure: Dialog policy $\pi(s, a; \theta_Q, \theta_m)$
initialize $\pi(s, a; \theta_Q, \theta_m)$
for $n$=1:$N$ do
    # Dialogue Complexity Evaluation starts
    user starts dialogues $\tau_1 ... \tau_k$ based on $G_i$
    update the dialogue process by $\pi(s, a; \theta_Q, \theta_m)$
    store training tuples $\{(s_t, a_t, r_t, s_{t+1}), ...\}$ for each dialog $\tau_k$
    compute complexity score for $\tau_k$ and $G_i$ by $f_{score}$ in Sec. 3.2
    sort $\mathcal{D}_{total}$ with the complexity score to obtain $\mathcal{D}_{order}$
    # Policy Learning Curriculum starts
    sample batches from $\mathcal{D}_{order}$ by $f_{pace}$ in Eq. 7.
    user starts dialogues $\tau_1 ... \tau_k$ based on $G_i^{'}$
    agent updates $\pi(s, a; \theta_Q, \theta_m)$ by interacting with the user
end for
```

by calculating its generated dialog sessions average complexity evaluation which is formulated as:

$$f_{score}(G_i) = \frac{1}{K} \sum_{k=1}^{K} f_{score}(m_{\tau_k}). \tag{6}$$

We sort the user goal set $\mathcal{D}_{total}$ by complexity scores, obtaining the forward complexity ordered user goal set $\mathcal{D}_{order}$. Based on the ordered training set, we further propose the policy learning curriculum to schedule the training of the agent.

3.3 Automatic Curriculum Learning Framework for Dialog Policy Learning

Our proposed automatic curriculum learning framework is shown as Algorithm 1. During the dialogue policy learning process, we first initialize the RL-based dialog policy as $\pi(s, a; \theta_Q, \theta_m)$. For each training step n, we start from the Dialogue Complexity Evaluation process. The user starts dialogues $\tau_1 ... \tau_k$ based on a user goal G_i. Then, we update the dialogue process based on the current policy, and store training tuples $\{(s_t, a_t, r_t, s_{t+1}), ...\}$ for each dialog τ_k. After that, we arrange to sort each user goal in the original training set from easy to complex according to the automatic scoring function f_{score} in Sec. 3.2, obtaining the ordered training set $\mathcal{D}_{order}$.

After obtaining the ordered training set, we start the policy learning curriculum process based on a pacing function $f_{pace}(t)$ aiming to schedule training instances that feeding to the policy learning network. A batch of training instances is sampled from the top $f_{pace}(t)$ percentage of the ordered training set. Following Platanios et al. (2019), we define the pacing function $f_{pace}(t)$ as:

$$f_{pace}(t) \triangleq min(1, \sqrt{t\frac{1-c_0^2}{N} + c_0^2}) \tag{7}$$

where $c_0 > 0$ is set to 0.01 and N is the duration of curriculum learning. At the early stage of the training process, the dialogue policy learning model learns from the instances drawing from the front (easy) part of the curriculum. With the advance of the curriculum, the complexity gradually increases, more complex training instances appear.

Based on sampled training instances by Eq. 7, the user starts dialogues from ordered sampling user goal $G_i^{'}$, while the agent interacts with the user to update and learn the dialogue policy $\pi(s, a; \theta_Q, \theta_m)$ from easy to complex, then conducts new dialogue complexity evaluation iteratively. Note that, although we describe the framework in two components for ease of understanding, in fact, the whole framework can be trained in an end-to-end manner. Thus, the proposed automatic curriculum learning framework is capable of not only scheduling the training samples to optimize the dialogue policy learning model, but also can be applied to other RL based training scenarios.

	Movie-Ticket Booking	MultiWoz
Dialogues	2,890	8,438
Intents	11	13
Slots	29	25
Avg. turns per dialogue	7.5	13.68
Domains	1	7

Table 1: The statistics of two public datasets, Movie-Ticket Booking and MultiWoz.

4 Experiments

4.1 Datasets

We conduct the experiments on two public datasets, Movie-Ticket Booking (Li et al., 2018) and MultiWoz (Budzianowski et al., 2018). Movie is a single domain dataset, and its goal is to build a dialogue system to help users find information about movies and book movie tickets. MultiWoz is a multi-domain, multi-intent task-oriented dialog corpus that contains seven domains, including Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train. A user may change his goal during the session, hence MultiWoz provides more complicated dialogs closer to real-world conversations. We show the full statistics in Table 1 [3].

[3] Refer to the appendix for details on datasets.

4.2 Evaluation Metrics

We adopt three metrics to evaluate the quality of policy learning: success rate, average turns, average reward. Success rate is the task completion rate – the fraction of dialogues that ended successfully. The dialog is successfully when all the requested information has been filled in, and the booked entities match all the indicated constraints given by the user goal. Average turns is the average length of the dialogue. Average reward is the average reward that agent received during the conversation.

4.3 Baselines

To verify the effectiveness of our proposed curriculum learning framework, we compare with different RL policy learning agents in task-completion dialogue as baselines:

DQN: A reinforcement learning based task-oriented dialogue model that the agent learned the dialogue policy by standard DQN (Li et al., 2017).

DDQ: The state-of-the-art task-oriented dialogue policy learning approach, Deep Dyna-Q (DDQ) (Peng et al., 2018b). DDQ integrates planning for dialogue policy learning to make the best use of limited real user experiences.

DDQ-CR: A DDQ agent only applied the curiosity reward into the dialogue policy learning but without curriculum learning (Wesselmann et al., 2019).

DDQ-CL-rule: A DDQ agent incorporating the curriculum learning (CL) and the dialogue complexity evaluation is rule-based, where empirically evaluating the dialogue by its slots number.

DDQ-SDPL: A DDQ agent that incorporating our proposed framework.

WDQN: A warm-DQN agent pre-trained by supervised learning (Lee et al., 2019).

WDQN-CR: A warm-DQN agent that only incorporating the curiosity rewards into dialogue policy learning without curriculum learning (Wesselmann et al., 2019).

WDQN-CL-rule: A warm-DQN agent that incorporating the rule-based curriculum learning (CL), similar to **DDQ-CL-rule**.

WDQN-SDPL: A warm-DQN agent that incorporating our proposed framework.

4.4 Settings

The policy learning of the agent is trained by DQN and DDQ algorithm with the same set of hyper-parameters on Movie-Ticket Booking (Li et al., 2018). The batch size is 64, and the learning rate is 0.001. The state encoding network is an MLP layer with 80 node. The optimizer for neural networks is RMSProp. And we employ the WDQN with the same set of hyper-parameters (Lee et al., 2019). The batch size is 16, and the learning rate is 0.001. The state feature encoder is an MLP layer with 300 node. We use Adam as the optimization algorithm. Besides, dialogues are ordered by our complexity evaluation in the training process while during the testing process, dialogues are used as baselines with no order. The training epoch is 100, 200, 300 and each number is averaged over 5 runs, each run tested on 2000 dialogues.

5 Automatic Evaluation

In this section, we first show the experiment results of our proposed method on Movie-Ticket Booking and MultiWoz datasets to verify the effectiveness of our proposed framework. Then we make a comprehensive qualitative analysis to show the merits of automatic evaluation for dialogue complexity.

5.1 Main Results

Table 2 presents the main results at the training epoch on the Movie-Ticket Booking dataset. For each agent, we report its results in terms of success rate, average reward, and average turns. Compared to the original DDQ agent, DDQ-CR, DDQ-CL-rule and DDQ-SDPL respectively achieve 8%, 7% and 10% improvements in terms of success rate, which confirms curriculum learning can facilitate and stable the training process of RL agents.

Further, our proposed automatic evaluation method for dialogue complexity outperforms the empirical method by 3%. The result represents applying the heuristic attribute to model the dialogue complexity is insufficient and it is hard to make effective handcrafted rules, due to subjectivity and diversity of dialogues and the limit of user-generated training. Our proposed method models the dialogue complexity in an automatic method evaluating dialog state differential space of RL agents. For other metrics, average reward and average turns, our method also achieves better performance. Apart from overall performance gain, we observe our method converges fast to similar accuracy 76% at epoch 100, compared to 73% of DDQ agent at epoch 300. It proves a better evaluation method for dialogue complexity can bootstrap training steps of RL agents. Besides,

Agent	Epoch = 100			Epoch = 200			Epoch = 300		
	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns
DQN (Li et al., 2017)	0.42	-3.84	31.93	0.53	10.78	22.72	0.64	27.66	22.21
DDQ (Peng et al., 2018b)	0.60	20.35	26.65	0.71	36.76	19.55	0.73	39.97	18.99
DDQ-CR	0.62	24.82	23.78	0.78	46.52	14.05	0.81	50.16	13.27
DDQ-CL-rule	0.71	32.48	19.01	0.76	45.36	14.56	0.80	48.15	13.59
DDQ-SDPL	**0.76**	**44.75**	**14.90**	**0.80**	**51.15**	**13.29**	**0.83**	**54.63**	**12.73**

Table 2: Results of different task-completion dialogue agents including the DQN,DDQ on Movie-Ticket Booking (Li et al., 2018) and ablation experiments of applying different curriculum learning strategies. The training epoch = {100, 200, 300}. (Success: success rate)

Agent	Epoch = 100			Epoch = 200			Epoch = 300		
	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns
WDQN (Lee et al., 2019)	0.37	-8.93	15.33	0.46	3.26	13.94	0.47	5.33	13.07
WDQN-CR	0.36	-5.16	15.10	0.45	3.21	14.02	0.51	10.68	13.79
WDQN-CL-rule	0.40	-4.79	14.79	0.48	6.41	13.19	0.49	9.35	13.85
WDQN-SDPL	**0.45**	**2.51**	**13.49**	**0.52**	**11.04**	13.36	**0.57**	**12.22**	13.81

Table 3: Results of WDQN agent on MultiWOZ (Budzianowski et al., 2018) and ablation experiments of applying different curriculum learning strategies. The training epoch = {100, 200, 300}. (Success: success rate)

DDQ-CR only applied curiosity-driven exploration with no curriculum learning and it can improve the agent's performance. While based on the curiosity reward's improvement, there is another promotion after applying the curriculum learning. And this ablation study demonstrates the effectiveness of the curriculum learning.

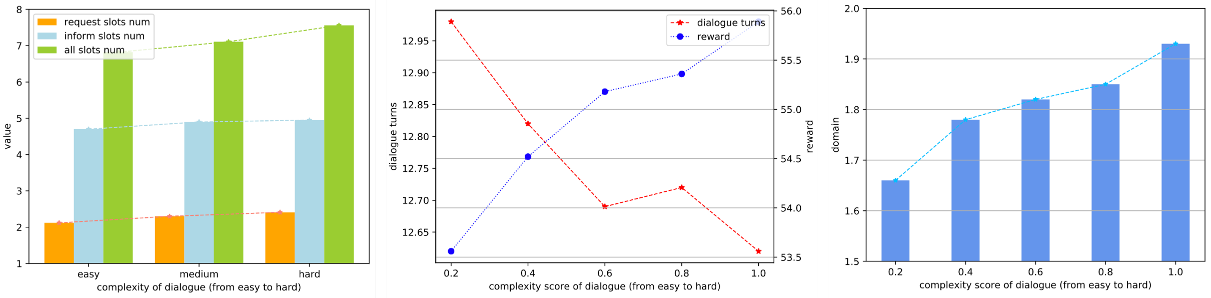
Table 3 shows main results at the training epoch on the MultiWOZ dataset. We aim to verify the effectiveness of our method on the more complicated dataset since MultiWOZ contains multi-domains and more dialog turns. Here we choose another RL agent, WDQN, as the baseline to show our proposed framework can be applied to other RL based models. Results show our method WDQN-SDPL outperforms the empirical rule WDQN-CL-rule by 6%, higher than the improvement of 3% on Movie dataset. The possible reason is that MultiWOZ is significantly complicated than Movie dataset and only the simple attribute of slot numbers evaluates dialog complexity insufficiently. For the Turns metric, our method achieves similar performance with the baselines. We assume the domain switching on MultiWOZ leads to increasing dialog turns to achieve a higher success rate. The comparison exhibits the effectiveness of our automatic evaluation method. Besides, the ablation study in WDQN-CR and WDQN-SDPL also demonstrates the effectiveness of the curriculum learning.

In summary, our proposed curriculum learning framework facilitates the training process of RL agents. Moreover, our automatic evaluation method for dialogue complexity consistently outperforms the rule-based empirical method.

5.2 Analysis

Comparison with Heuristic Complexity We analyze three empirical attributes to show the correlation between heuristics and proposed automatic dialogue complexity evaluation: slot number, dialog length, reward, and domain switching times. We show the correlation statistics results in Fig 3. Fig 3(a) and Fig 3(b) shows the statistics of slots number, dialogue length and reward in complexity ordered instances evaluating by our proposed dialog complexity evaluation on Movie-Ticket Booking dataset. And we perform the qualitative analysis on the correlation between dialogue complexity and the number of domain switching on the MultiWoz dataset in Fig 3(c). Our evaluation method for dialogue complexity reflects positive connections to heuristic methods in these dimensions, which to some extent explains the effectiveness of our method. On the other hand, our method considers multiple empirical dialog attributes which is more flexible than heuristics. We show more comparison analysis details in the Appendix.

Learning Efficiency Fig 4 shows the learning curves comparing different agents, DDQ, DDQ-CL-rule and DDQ-SDPL, to investigate the relative contribution of the automatic complexity evaluation and the curriculum learning strategy. The



(a) *Slot numbers* (b) *dialog length & reward* (c) *domain switching times*

Figure 3: The correlation statistics between heuristics and our proposed automatic dialogue complexity evaluation. We count three heuristic attributes: slot numbers(request slot, inform, all), dialog turns and reward, domain switching times based on ordered training samples evaluating by our proposed automatic complexity evaluation. And we divide obtained dialog complexity scores into different complexity intervals(easy, medium, hard) for statistics.

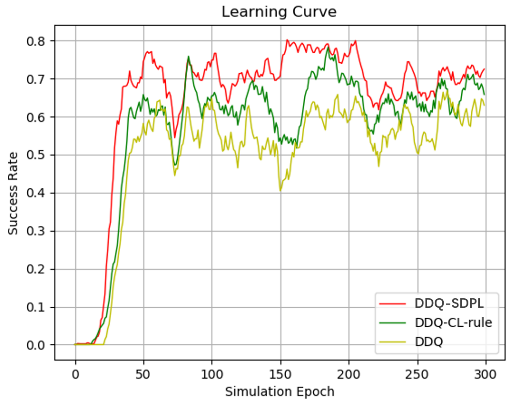


Figure 4: Learning curves of the DDQ, DDQ-CL-rule, DDQ-SDPL. The DDQ-SDPL outperforms baselines and converges fast to achieve higher accuracy.

DDQ-SDPL outperforms the baseline DDQ and DDQ-CL-rule and at each training step and converges fast to higher accuracy.

6 Human Evaluation

For human evaluation, we hire human experts to compare pairwise between DDQ-SDPL and baselines. Given a certain user goal, each expert is asked to read two simulated dialog sessions around this user goal, one from DDQ-SDPL and another from the other baseline. We randomly sample 100 goals for each baseline. For each goal, 3 experts are asked to judge which dialog is better (win, draw or lose) according to different subjective assessments: quality and task success. The quality metric evaluates whether the agent policy provides the user with the required information efficiently.

Table 4 shows the results of the human preference by majority voting. DDQ-SDPL outperforms other baselines significantly in all aspects (sign test, p-value < 0.01). Note that the difference between DDQ-CL-rule and DDQ-SDPL is only in the dialog complexity evaluation. This demonstrates again the advantage of the automatic complexity evaluation in DDQ-SDPL over the heuristic method. The human preferences agree well with the results of the automatic evaluation, which also indicates these experimental metrics are reliable to reflect user satisfaction to some extent. Besides, we show some sampled cases in the Appendix to demonstrate the effectiveness of our proposed learning framework.

VS.	Quality			Success		
	W	D	L	W	D	L
DDQ	46	24	30	58	25	17
DDQ-CL-rule	41	28	31	49	26	25

Table 4: Human preference on dialog session pairs that DDQ-SDPL wins (W), draws with (D) or loses to (L) baselines on quality and success by majority voting.

7 Conclusion

In this paper, we propose a novel curriculum learning framework to improve dialog policy learning by scheduling its learning process from easy to complex. We further propose an automatic dialog complexity evaluation for curriculum scheduling. The effectiveness validation of SDPL is conducted on two dialogue datasets and the state-of-the-art dialog model demonstrates that our proposed learning framework is able to boost the performance of existing dialogue policy learning. Furthermore, we believe that this automatic curriculum learning framework can be applied to improve other types of reinforcement learning based NLP tasks.

Acknowledgments

This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.

Hengyi Cai, Hongshen Chen, Cheng Zhang, Yonghao Song, Xiaofang Zhao, Yangxi Li, Dongsheng Duan, and Dawei Yin. 2020. Learning from easy to complex: Adaptive multi-curricula learning for neural dialogue generation. *CoRR*, abs/2003.00639.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards end-to-end reinforcement learning of dialogue agents for information access. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 484–495. Association for Computational Linguistics.

Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2019. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. *CoRR*, abs/1911.08717.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2016. VIME: variational information maximizing exploration. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1109–1117.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Xiang Li, Yaoqin Zhang, Zheng Zhang, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019. Convlab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 72–79. IEEE.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Çelikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 733–743. Asian Federation of Natural Language Processing.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Julia Kiseleva, Maarten de Rijke, Shahin Shayandeh, and Jianfeng Gao. 2020. Guided dialogue policy learning without adversarial learning in the loop. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2308–2317. Association for Computational Linguistics.

Pierre Lison and Serge Bibauw. 2017. Not all dialogues are created equal: Instance weighting for neural conversational models. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, pages 384–394. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum learning for natural answer generation. In *IJCAI*, pages 4223–4229.

Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. 2012. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems*

2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, pages 206–214.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017a. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2778–2787. PMLR.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017b. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Yun-Nung Chen, and Kam-Fai Wong. 2018a. Adversarial advantage actor-critic model for task-completion dialogue policy learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6149–6153. IEEE.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018b. Deep Dyna-Q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192, Melbourne, Australia. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M. Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *CoRR*, abs/1903.09848.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463.

Jürgen Schmidhuber. 1991. Adaptive confidence and adaptive curiosity. In *Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2*. Citeseer.

Jonathan Sorg, Satinder P. Singh, and Richard L. Lewis. 2010. Variance-based rewards for approximate bayesian reinforcement learning. In *UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010*, pages 564–571. AUAI Press.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. 2015. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.

Susanne Still and Doina Precup. 2012. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory Biosci.*, 131(3):139–148.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative deep Dyna-Q: Robust planning for dialogue policy learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3813–3823, Brussels, Belgium. Association for Computational Linguistics.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition.

Christopher Tegho, Pawel Budzianowski, and Milica Gasic. 2018. Benchmarking uncertainty estimates with deep reinforcement learning for dialogue policy optimisation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 6069–6073. IEEE.

Paula Wesselmann, Yen-Chen Wu, and Milica Gašić. 2019. Curiosity-driven reinforcement learning for dialogue management. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7210–7214. IEEE.

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

A Dataset

Table 5 shows the full statistics of Movie-Ticket Booking dataset. And the statistics of MultiWOZ dataset are shown in Table 6. And Table 7 demonstrates the full statistics of comparison that between the Movie-Ticket Booking and MultiWOZ dataset.

Annotations	
Intent	request, inform, deny, confirm_question, confirm_answer, greeting, closing, not_sure, multiple_choice, thanks, welcome
Slot	city, closing, date, distanceconstraints, greeting, moviename, numberofpeople, price, starttime, state, taskcomplete, theater, theater_chain, ticket, video_format, zip

Table 5: The data annotation schema of Movie-Ticket Booking dataset.

	Hotel	**Train**	**Attraction**	**Restaurant**	**Taxi**
Slots	price, type, parking, stay, day, people, area, stars, internet, name	destination, departure, day, arrive by, leave at, people	area, name, type	food, price, area, name, time, day, people	destination, departure, arrive by, leave by

Table 6: The dataset information of MultiWOZ.

B User goal

In the task-completion dialogue setting, the entire conversation is around a user goal $G = (C, R)$ implicitly, where C denotes the constraint and R is the requests. And the agent knows nothing about the user goal explicitly and its objective is to help the user to accomplish this goal. Every time a dialog is launched, the user goal is initialized by the user simulator at the beginning of a dialog session, by randomly sampling the *inform slots* and *requests slots* from the user goal. Generally, the definition of user goal contains two parts: *inform_slots* contain a number of slot-value pairs which serve as constraints from the user. *request_slots* contain a set of slots that user has no information about the values, but wants to get the values by interact with the agent.

A set of 1,000 user goals in MultiWOZ are used for automatic evaluation as shown in Table 8.

C Case study

To further analyze the effectiveness of our automatic complexity measurement qualitatively, we randomly choose three examples of dialogues with different normalized complexity scores. As shown in Table 9, it is clear to see that, the instance with more slots and domains obtains a higher complexity score in our automatic measurement, which is consistent with heuristics. Besides, Table 11 shows two dialog sessions with the same user goal in WDQN and WDQN-SDPL, respectively. The user goal consists of *train* and *restaurant* domains. The system is required to answer all the information in *Request* based on the constraints in *Inform*. When the user asks for some information about the restaurant, WDQN-SDPL manages to provide all the requested information, while original WDQN only answers part of the information(red line). Additionally, WDQN-SDPL answers the postcode of the restaurant (blue line) and the dialog proceeds successfully, however, WDQN fails.

	Movie-Ticket Booking	MultiWoz
Dialogues	2,890	8,438
Intents	11	13
Slots	29	25
Avg. turns per dialogue	7.5	13.68
Domains	1	7

Table 7: The statistics of two public datasets, Movie-Ticket Booking and MultiWoz.

Class	Attraction	Hospital	Hotel
Count	320	22	389
Police	Restaurant	Taxi	Train
22	457	164	421
Num.	Single	Two	Three
Count	328	549	123

Table 8: Domain distribution of user goals used in the automatic evaluation. A user goal with multiple domains is counted repeatedly for each domain.

Easy case with complexity score: 0.2		**Medium** case with complexity score: 0.6		**Hard** case with complexity score: 0.9	
Restaurant		**Train**		**Attraction**	
Inform	*Request*	*Inform*	*Request*	*Inform*	*Request*
food=european pricerange=expensive time=19:30	phone postcode address	destination=peterborough day=thursday departure=cambridge leaveAt=19:45	duration arriveBy	type=museum area=centre	entrance fee address postcode
		Restaurant		**Hotel**	
		Inform	*Request*	*Inform*	*Request*
		food=european pricerange=expensive	phone postcode address	price range=expensive stars=4 people=8 day=saturday	address area
				Taxi	
				Inform	*Request*
				arrive by=19:45	car type phone

Table 9: The case study of different training instances (Easy, Medium, Hard) with their different complexity scores. The instance with more slots and domains obtains a higher complexity score in our automatic measurement, which is consistent with heuristics.

Train		Restaurant	
Inform	Request	Inform	Request

destination=peterborough
day=sunday
departure=cambridge
leaveAt=20:45
	duration
	people= 7
		food=british
		price range=expensive
		area=centre
			postcode
			address

Table 10: User goal used in the case study.

WDQN	**WDQN-SDPL**
U: I need to book a train to peterborough. I will departing from cambridge. Yes, I need to leave sometimes after 20:45. I would like to leave on sunday.	U: I need to book a train to peterborough. I will departing from cambridge. I need to leave sometimes after 20:45. I would like to leave on sunday.
S: What day would you like your booking for?	S: What day would you like your booking for? What time would you like the reservation for?
U: Sunday is fine, what is the travel time on that?	U: That should work, what is the travel time on that?
S: What time would you like the reservation for?	S: The travel time is 51 minutes.
U: That should work, what is the travel time on that?	U: Yes, can you book that for 7.
S: What time would you like the reservation for?	S: Reference number is : 00000068.
U: can you book that for 7?	U: I want to find a expensive priced restaurant. I'm visiting Cambridge and would like some suggestions for an restaurant which serves british. What restaurants are located in the centre?
S: Reference number is : 00000068.	S: Do you have a preference for the price range? I have 3 options for you.
U: I want to find a expensive priced restaurant. I'm visiting Cambridge and would like some suggestions for an restaurant which serves british. What restaurants are located in the centre?	U: I am looking for a expensive restaurant.
S: Do you have a preference for the price range? I have 3 options for you.	S: All set . Your reference number is 00000045.
U: I am looking for a expensive restaurant.	U: May I have the address? What is the postal code?
S: All set. Your reference number is 00000045.	S: The address is 51 Trumpington Street City Centre.
U: What is the address? What is the postal code?	U: I will need the postcode though.
S: The address is 51 Trumpington Street City Centre.	S: The post code is cb21rg.
U: Thanks. I will need the postcode though.	U: I am all set. Have a nice day. Bye.
S: Your reference number is 00000045.	S: You are very welcome. Goodbye.
U: I am all set. Have a nice day. Bye.	
S: You are very welcome. Goodbye.	
Failure: User does not request the arrive time of the train, and the system does not book the restaurant for the user.	**Success**

Table 11: A sample dialog session comparison between WDQN and WDQN-SDPL in natural language.