Automatic Curriculum Learning With Over-repetition Penalty for Dialogue Policy Learning

Yangyang Zhao, Zhenyu Wang *, Zhenhua Huang
School of software engineering, South China University of Technology
msyyz@mail.scut.edu.cn, wangzy@scut.edu.cn, sezhhuangscut@mail.scut.edu.cn

Abstract

Dialogue policy learning based on reinforcement learning is difficult to be applied to real users to train dialogue agents from scratch because of the high cost. User simulators, which choose random user goals for the dialogue agent to train on, have been considered as an affordable substitute for real users. However, this random sampling method ignores the law of human learning, making the learned dialogue policy inefficient and unstable. We propose a novel framework, Automatic Curriculum Learning-based Deep Q-Network (ACL-DQN), which replaces the traditional random sampling method with a teacher policy model to realize the dialogue policy for automatic curriculum learning. The teacher model arranges a meaningful ordered curriculum and automatically adjusts it by monitoring the learning progress of the dialogue agent and the over-repetition penalty without any requirement of prior knowledge. The learning progress of the dialogue agent reflects the relationship between the dialogue agent's ability and the sampled goals' difficulty for sample efficiency. The over-repetition penalty guarantees the sampled diversity. Experiments show that the ACL-DQN significantly improves the effectiveness and stability of dialogue tasks with a statistically significant margin. Furthermore, the framework can be further improved by equipping with different curriculum schedules, which demonstrates that the framework has strong generalizability.

Introduction

Learning dialogue policies are typically formulated as a reinforcement learning (RL) problem (Sutton and Barto 1998; Young et al. 2013). However, dialogue policy learning via RL from scratch in real-world dialogue scenarios is expensive and time-consuming, because it requires real users to interact with and adjusts its policies online (Mnih et al. 2015; Silver et al. 2016; Dhingra et al. 2017; Su et al. 2016b; Li et al. 2017). A plausible strategy is to use user simulators as an inexpensive alternative for real users, which randomly sample a user goal from the user goal set for the dialogue agent training (Schatzmann et al. 2007; Su et al. 2016a; Li et al. 2017; Budzianowski et al. 2017; Peng et al. 2017; Liu and Lane 2017; Peng et al. 2018a). In task-oriented dialogue settings, the entire conversation revolves around the sampled user goal implicitly. Nevertheless, the dialogue agent's objective is to help the user to accomplish this goal even though the agent knows nothing about this sampled user goal (Schatzmann and Young 2009; Li et al. 2016), as shown in Figure 1a.

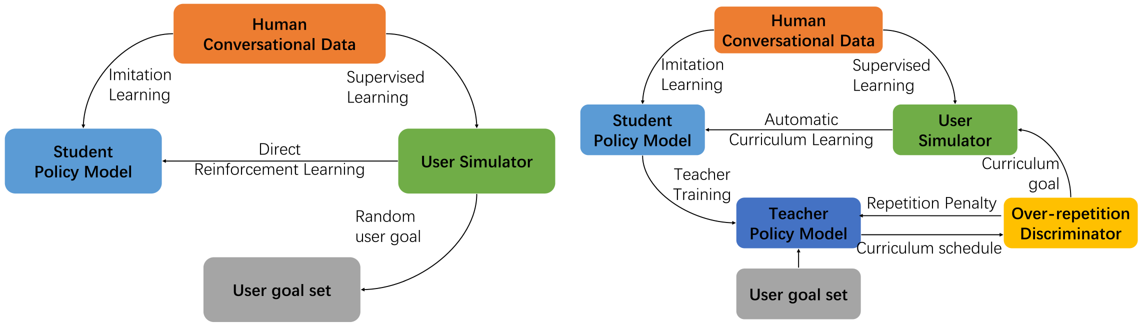
The randomly sampling-based user simulator neglects the fact that human learning supervision is often accompanied by a curriculum (Ren et al. 2018). For instance, when a human-teacher teaches students, the order of presented examples is not random but meaningful, from which students can benefit (Bengio et al. 2009). Therefore, this randomly sampling-based user simulators bring two issues:

- *efficiency* issue: since the ability of the dialogue agent does not match the difficulty of the sampled user goal, it takes a long time for the dialogue agent to learn the optimal strategy (or fail to learn). For example, in the early learning phase, it is possible that the random sampling method arranges the dialogue agent to learn more complex user goals first, and then learn simpler user goals.
- *stability* issue: using random user goals to collect experience online is not stable enough, making the learned dialogue policy unstable and difficult to reproduce. Since RL is highly sensitive to the dynamics of the training process, dialogue agents trained with stable experience can guide themselves more effectively and stably than dialogue agents trained with instability.

Most previous studies of dialogue policy have focused on the *efficiency* issue, such as reward shaping (Kulkarni et al. 2016; Lu, Zhang, and Chen 2019; Zhao et al. 2020), companion learning (Chen et al. 2017a,b), incorporate planning (Peng et al. 2018b; Su et al. 2018; Wu et al. 2019; Zhao et al. 2020), etc. However, *stability* is a pre-requisite for the method to work well in real-world scenarios. It is because, no matter how effective an algorithm is, an unstable online leaned policy may be ineffective when applied in the real dialogue environment. This can lead to bad user experience and thus fail to attract sufficient real users to continuously improve the policy. As far as we know, little work has been reported about the stability of dialogue policy. Therefore, it is essential to address the stability issue.

In this paper, we propose a novel policy learning framework that combines curriculum learning and deep reinforce-

*Corresponding authors.



(a) Policy learning with user simulators.

(b) Policy learning with proposed ACL-DQN framework.

Figure 1: Two strategies of user simulator sampling for learning task-oriented dialogue policies via RL.

ment learning, namely Automatic Curriculum Learning-based Deep Q-Network (ACL-DQN). As shown in Figure 1b, this framework replaces the traditional random sampling method in the user simulator with a teacher policy model that arranges a meaningful ordered curriculum and dynamically adjusts it to help dialogue agent (also referred to student agent in this paper) for automatic curriculum learning. As a scheduling controller for student agents, the teacher policy model arranges students to learn different user goals in different learning stages without any requirement of prior knowledge. Sampling the user goals that match the ability of student agents regarding different difficulty of each user goal, can not only increases the feedback of the environment to the student agent but also makes the learning of the student agent more stable.

There are two criteria for evaluating the sampling order of each user goal: the learning progress of the student agent and the over-repetition penalty. The learning progress of the student agent emphasizes the efficiency of each user goal, encouraging the teacher policy model to choose the user goals that match the ability of the student agent to maximize the learning efficiency of the student agent. The over-repetition penalty emphasizes the sampled diversity, preventing the teacher policy model from *cheating*[1]. The incorporation of the learning progress of the student agent and the over-repetition penalty reflects both sampled efficiency and sampled diversity to improve efficiency as well as stability of ACL-DQN.

Additionally, the proposed ACL-DQN framework can equip with different curriculum schedules. Hence, in order to verify the generalization of the proposed framework, we propose three curriculum schedule standards for the framework for experimentation: i) *Curriculum schedule A*: there is no standard, only a single teacher model; ii) *Curriculum schedule B*: user goals are sampled from easiness to hardness in proportion; iii) *Curriculum schedule C*: ensure that the student agents have mastered simpler goals before learning more complex goals.

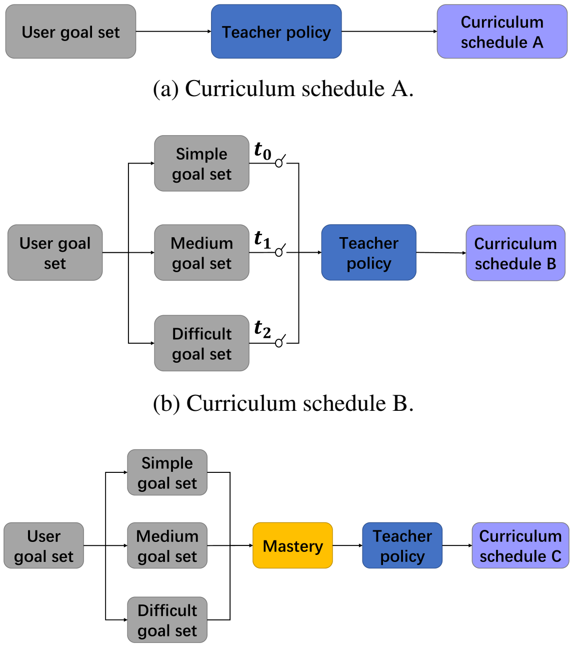
Experiments have demonstrated that the ACL-DQN significantly improves the dialogue policy through automatic curriculum learning and achieves better and more stable performance than DQN. Moreover, the ACL-DQN equipped with the curriculum schedules can be further improved. Among the three curriculum schedules we provided, the ACL-DQN under curriculum schedule C with the strength of supervision and controllability, can better follow up on the learning progress of students and performs best. In summary, our contributions are as follows:

- We propose Automatic Curriculum Learning-based Deep Q-Network (ACL-DQN). As far as we know, this is the first work that applies curriculum learning ideas to help the dialogue policy for automatic curriculum learning.
- We introduce a new user goal sampling method (i.e., teacher policy model) to arrange a meaningful ordered curriculum and automatically adjusts it by minoring the learning progress of the student agent and the over-repetition penalty.
- We validate the superior performance of ACL-DQN by building dialogue agents for the movie-ticket booking task. The efficiency and stability of ACL-DQN are verified by simulation and human evaluations. Moreover, ACL-DQN can be further improved by equipping curriculum schedules, which demonstrates that the framework has strong generalizability.

Proposed Framework

The proposed framework is illustrated in Figure 1a, the ACL-DQN agent training consists of four processes: (1) *curriculum schedule* includes three strategies (Figure 2), which are arranged by the teacher policy model based on three standards we provided and automatically adjusted according to the learning process of the student agent and the over-repetition penalty. (2) *over-repetition penalty*, which punishes the *cheating* behaviors of the teacher policy model to guarantee the sampled diversity. (3) *automatic curriculum learning*, where the student agent interacts with a user simulator revolving around curriculum goal specified by the teacher policy model, collects experience, improves the student dialogue policy, and feeds its performances back to the teacher policy model for adjusting. (4) *teacher reinforcement learning*, where the teacher policy model is leaned and refined through a separate teacher experience replay buffer.

[1] The teacher policy model repeatedly selects user goals that the student agent has mastered to obtain positive rewards.



(a) Curriculum schedule A.

(b) Curriculum schedule B.

(c) Curriculum schedule C.

Figure 2: Three curriculum schedules were arranged and adjusted by the DQN-based teacher model based on three standards by monitoring student's training process and the over-repetition penalty (the feedback is shown in Figure 1a).

Curriculum Schedule

In this section, we introduce a DQN-based teacher model and three curriculum schedules, which are later used in the (2), (3), and (4) processes mentioned above.

DQN-based Teacher Model The goal of the teacher model is to help the student agent learn a series of user goals sequentially. We can formalize the teacher goal as a Markov decision process (MDP) problem, which is well-suitable for reinforcement learning to solve:

- The state s_t consists of five components: 1) the state provided by the environment; 2) ID of the current user goal; 3) ID of last user goal; 4) a scalar representation of student policy network's parameters under the current user goal; 5) a scalar representation of student policy network parameters under the last user goal.
- The action a_t corresponds user goal chosen g_t by teacher policy model.
- The reward r consists of two parts, one is the reward r_t^{or} from the *Over-repetition Discriminator*, and the other r_t^c is the change in episode total reward acquired by the student for the user goal g_t, formulated as:

$$r_t = r_t^{or} + r_t^c = r_t^{or} + x_t^{g_t} - x_{t'}^{g_t} \tag{1}$$

where $x_{t'}^{g_t}$ is the previous episode total reward when the same user goal g_t was trained on.

In this article, we user the deep Q-network (DQN) (Mnih et al. 2015) to improve the teacher policy based on teacher experience. In each step, the teacher agent takes the state s_t as input and chooses the action g_t to execute. The sampled user goal g_t is handed over to the *Over-repetition Discriminator* to judge whether it is over-sampling. if not, it will be passed to the user simulator as a goal to interact with the student agent, otherwise it will give the teacher agent a penalty. The more times the user goal has been selected, the greater penalty gave, the less the probability of being selected in the next step. During training, we use ϵ-greedy exploration that selects a random action with probability ϵ or otherwise follows the greedy policy $g_t = \arg\max_{g_t'} Q(s_t, g_t'; \theta^T)$. $Q(s_t, g_t; \theta^T)$ is the approximated value function, implemented as a Multi-Layer Perceptron (MLP) parameterized by θ^T. When the dialogue terminates, the teacher agent then receives reward r_t, and updates the state to s_{t+1}. At each simulation epoch, we simulate N $(N = 1)$ [1] dialogues and store the experience (s_t, g_t, r_t, s_{t+1}) in the teacher experience replay buffer D^T for teacher reinforcement learning. This cycle continues until the num_episodes is reached.

```
Algorithm 1 ACL-DQN with Curriculum schedule A
1: the DQN-based teacher model with probability $\epsilon$ select
   a random action $g_i$ in the user goal $G$;
2: otherwise the DQN-based teacher model select $g_i =$
   $\arg\max_{g'} Q(s^t, g'; \theta^T)$ in the user goal $G$;
```

Curriculum Schedule A As shown in Figure 2a, in order to evaluate the effect of a single DQN-based teacher model clearly, we replace the traditional sample method in user simulators with a single DQN-based teacher model that directly selects a user goal from the user goal set and dynamically adjust it according to the learning progress of the student agent and the over-repetition penalty using a ϵ-greedy exploration (Algorithm 1).

Curriculum Schedule B In our curriculum schedule B, we make the learning process of the student agents similar to the education process of human students, which is that students usually learn many easier curriculums before they start to learn more complex curriculums (Ren et al. 2018). Accordingly, we integrate user goal ranking in Curriculum schedule A, which allows student agents under the guidance of Curriculum schedule B to achieve progressive learning from easiness to hardness in proportion (Figure 2b).

We take the total number of inform_slot and request_slot n in the user goals as a measure of the difficulty of each user goal. According to this measure, user goals are divided into three groups from easiness to hardness: simple user goal set G_{simple}, medium user goal set G_{medium}, and difficult user goal set $G_{difficult}$. In the learning process of the student agents, we set the three user goal sets (from easiness to hardness) as the action set of the teacher agent sequentially to guarantee that the stu-

[1] Considering the user cost in real dialogue scenarios, we set up only 1 simulation epoch for experience storage, $N = 1$, to better reflect the performance of the proposed method on real dialogue tasks.

Algorithm 2 ACL-DQN with Curriculum schedule B

```
Get the total number of inform_slot $n_i$ and the number of request_slot $n_r$ of each user goal, $n = n_i + n_r$;
Sort user goal set $G$ based on $n$ and divide it into three groups, simple user goal set $G_{simple}$ (30), medium user goal set $G_{medium}$ (72), and difficult user goal set $G_{difficult}$ (26);
Initialize curriculum_phase = 'simple'
if $len(G_{curriculum\_phase})/len(G) * epoch\_size$ have been reached then
    curriculum_phase = next_difficult_stage();
else
    curriculum_phase = stay_current_stage();
    the DQN-based teacher model with probability $\epsilon$ select a random action $g_i$ in $G_{curriculum\_phase}$;
    otherwise the DQN-based teacher model select $g_i = \arg\max_{g'} Q(s^t, g'; \theta^T)$ in $G_{curriculum\_phase}$;
end if
```

dent agents learn the user goals of each stage in an orderly manner (Algorithm 2).

Curriculum Schedule C The curriculum schedule B may slow down the student agent learning. The reason is that even if the student agent has quickly mastered the goals of the current difficulty, it still needs to continue learning the remainder of this current difficulty. Accordingly, we design the curriculum schedule C, which is integrated "mastery" in curriculum schedule B, as shown in Figure 2c. The curriculum schedule C supports the student agent to directly enter the user goal of the next stage without learning the remainder of the current difficulty if it has mastered the goals of the current difficulty.

It is considered that the student agent has mastered the user goals of this difficulty, if and only if the success rate of sampled user goals in the current difficulty exceeds the mastering threshold $\alpha(\alpha = 0.5)$[2] within a continuous-time T (T=5). The success rate of the sampled user goal in the current difficulty is $p_{success} = n_{success}/N_{sampled}$, where $n_{success}$ is the number of user goals completed by the student agent in the current difficulty, $N_{sampled}$ is the number of user goals sampled at the current difficulty (Algorithm 3).

Over-repetition Penalty

Under the three curriculum schedules mentioned above, the teacher policy model may *cheat* to obtain positive rewards, which is repeatedly selecting user goals that the student agent has mastered. Besides, it is clear that the limited size of replay memory makes overtraining even worse (De Bruin et al. 2015). Therefore, if the student agent is only restricted to some user goals already mastered, it will cause student agent learning to stagnate. For the sake of generalization of the proposed ACL-DQN method, we take into account guarantee the diversity of sampled user goals and integrate the over-repetition penalty mechanism in the framework.

Algorithm 3 ACL-DQN with Curriculum schedule C

```
Initialize curriculum_phase = 'simple', a mastery threshold $\alpha$, a list $L$ for storing the success rate of the sampled user goal in the current difficulty;
$p_{success} = n_{success}/N_{sampled}$;
$L.append(p_{success})$
if $episode \geq T$ then
    $L.remove(0)$
end if
for i in len(L) do
    if $L[i] \geq \alpha$ then
        $n = n + 1$
    end if
end for
if $n \geq$ T then
    curriculum_phase = next_difficult_stage();
else
    curriculum_phase = stay_current_stage();
end if
the DQN-based teacher model with probability $\epsilon$ select a random action $g_i$ in $G_{curriculum\_phase}$;
otherwise the DQN-based teacher model select $g_i = \arg\max_{g'} Q(s^t, g'; \theta^T)$ in $G_{curriculum\_phase}$;
```

Similar to the coverage mechanism in neural machine translation (Tu et al. 2016), we introduced an over-repetition vector $[og_1, og_2, ..., og_n]$ to the teacher experience replay buffer D^T for recording the sample times of each user goal. In the beginning, we initialize it as a zero vector with dimension $[1 * n]$, where n is the number of user goals in the current user goal set. In each simulation training step, if a user goal g_i is sampled, the corresponding variable over-repetition number og_i is update by $og_i = og_i + 1$. The more times the user goal has been selected, the greater the over-repetition penalty gave by the over-repetition discriminator, the less the probability of being selected in the next step. Thus, an over-repetition penalty function $ORP(og)$ satisfies the following requirements:

- $ORP(og) \to [-L, 0]$.
- $ORP(og)$ is a monotonically decreasing function of og.

where $L(L = 40)$ is the maximum length of a simulated dialogue.

Automatic Curriculum Learning

The goal of student agents is to achieve a specific user goal through a sequence of actions with a user simulator, which can be considered as an MDP. In this stage, we use the DQN method to learn the student dialogue policy based on experiences stored in the student experience replay buffer D^S :

- The state s_t consists of five components: 1) one-hot representations of the current user action and mentioned slots; 2) one-hot representations of last system action and mentioned slots; 3) the belief distribution of possible value for each slot; 4) both a scalar and one-hot representation of current turn number; and 5) a scalar representation indi-

[2] We verified it in the subsequent experiment, the ACL-DQN performs best when the mastery threshold is 0.5.

cating the number of results which can be found in the database according to current search constraints.

- The action a_t corresponds pre-defined action set, such as request, inform, confirm_question, confirm_answer, etc.
- The reward r: once a dialogue reaches the successful, the student agent receives a big bonus $2L$. Otherwise, it receives $-L$. In each turn, the student agent receives a fixed reward -1 to encourage shorter dialogues.

At each step, the student observes the dialogue s, and choose an action a, using an ϵ-greedy. The student agent then receives reward r,and updates the state to s'. Finally, we store the experience tuple (s, a, r, s') in the student experience replay buffer D^S. This cycle continues until the dialogue terminates.

We improve the value function $Q(s, a, \theta^S)$ by adjusting θ^S to minimize the mean-squared loss function as follows:

$$\begin{aligned} &\mathcal{L}(\theta^S) = \mathbb{E}_{(s,a,r,s')\sim D^S}[(y_i - Q(s, a; \theta^S))^2] \\ &y_i = r + \gamma \max_{a'} Q'(s', a'; \theta^{S'}) \end{aligned} \tag{2}$$

where $\gamma \in [0, 1]$ is a discount factor, and $Q'(\cdot)$ is the target value function that is only updated periodically. $Q(\cdot)$ can be optimized through $\nabla_{\theta^S}\mathcal{L}(\theta^S)$ by back-propagation and mini-batch gradient descent.

Teacher Reinforcement Learning

The teacher's function $Q(\cdot)$ can be improved using experiences stored in the teacher experience replay buffer D^T. In the implementation, we optimize the parameter Q^T w.r.t. the mean-squared loss:

$$\begin{aligned} &\mathcal{L}(\theta^T) = \mathbb{E}_{(s,g,r,s')\sim D^T}[(y_i - Q(s, g; \theta^T))^2] \\ &y_i = r_t^{or} + r_t^{change} + \gamma \max_{g'} Q'(s', g'; \theta^{T'}) \end{aligned} \tag{3}$$

where $Q'(\cdot)$ is a copy of the previous version of $Q(\cdot)$ and is only updated periodically and $\gamma \in [0, 1]$ is a discount factor. In each iteration, we improve $Q(\cdot)$ through $\nabla_{\theta^T}\mathcal{L}(\theta^T)$ by back-propagation and mini-batch gradient descent.

Experiments

Experiments have been conducted to evaluate the key hypothesis of ACL-DQN being able to improve the efficiency and stability of DQN-based dialogue policies, in two settings: simulation and human evaluation.

Dataset

Our ACL-DQN was evaluated on movie-booking tasks in both simulation and human-in-the-loop settings. Raw conversational data in the movie-ticket booking task was collected via Amazon Mechanical Turk with annotations provided by domain experts. The annotated data consists of 11 dialogue acts and 29 slots. In total, the dataset contains 280 annotated dialogues, the average length of which is approximately 11 turns.

Baselines

To verify the efficiency and stability of ACL-DQN, we developed different version of task-oriented dialogue agents as baselines to compare with.

- The **DQN** agent takes the user goal randomly sampled by the user simulator for leaning (Peng et al. 2018b).
- The proposed **ACL-DQN**(A) agent takes the curriculum goal specified by the teacher model equipped with *curriculum schedule A* for automatic curriculum learning (Alforithm 1).
- The proposed **ACL-DQN**(B) agent takes the curriculum goal specified by the teacher model equipped with *curriculum schedule B* for automatic curriculum learning (Alforithm 2).
- The proposed **ACL-DQN**(C) agent takes the curriculum goal specified by the teacher model equipped with *curriculum schedule C* for automatic curriculum learning (Alforithm 3).

Implementation

For all the models, we use MLPs to parameterize the value networks $Q(\cdot)$ with one hidden layer of size 80 and $tanh$ activation. ϵ-greedy is always applied for exploration. We set the discount factor $\gamma = 0.9$. The buffer size of D^T and D^S is set to 2000 and 5000, respectively. The batch size is 16, and the learning rate is 0.001. We applied gradient clipping on all the model parameters with a maximum norm of 1 to prevent gradient explosion. The target network is updated at the beginning of each training episode. The maximum length of a simulated dialogue is 40 turns. The dialogues are counted as failed, if exceeding the maximum length of turns. For training the agents more efficiently, we utilized a variant of imitation learning, called Reply Buffer Spiking (RBS) (Lipton et al. 2016) at the beginning stage to build a naive but occasionally successful rule-based agent based on the human conversational dataset. We also pre-filled the real experience replay buffer B^u with 100 dialogues before training for all the variants of agents.

Simulation Evaluation

Main result The main simulation results are depicted in Table 1, Figure 3, and 4. The results show that all the ACL-DQN agents under three curriculum schedules significantly outperforms the baselines DQN with a statistically significant margin. Among them, ACL-DQN(C) shows the best performance, and ACL-DQN(B) shows the worst performance. The important reason is that, regardless of the mastering progress of the student agent and only let the student agent learning from easiness to hardness will slow down the learning of the student agent. As shown in Figure 3, ACL-DQN(B) does not show significant advantages until after epoch 320, while ACL-DQN(C) consistently outperform DQN by integrating the mastery module that monitors the learning progress of student agent and adjusts it in real-time. Figure 4 is a boxplot of DQN and ACL-DQN under three curriculum schedules about the success rate at 500 epoch. It

Agent	Epoch = 100			Epoch = 200			Epoch = 300			Epoch = 400		
	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns
DQN	0.4012	-6.48	31.24	0.5242	10.36	27.08	0.6448	26.17	24.40	0.6598	28.73	22.88
ACL-DQN(A)	0.4309	-2.92	31.25	0.6159	22.99	23.84	0.7064	35.23	21.06	0.7419	40.19	19.66
ACL-DQN(B)	0.4202	-3.97	30.78	0.5678	16.29	25.69	0.6673	30.12	21.92	0.7073	35.81	20.11
ACL-DQN(C)	**0.5717**	15.92	27.36	**0.7253**	37.39	21.30	**0.7573**	45.28	18.57	**0.8055**	49.05	17.22

Table 1: Result of different agents at $epoch = \{100, 200, 300, 400\}$. Each number is averaged over 5 turns, each run tested on 50 dialogues. Success: Evaluated at the same epoch (except one group: at epoch 100, ACL-DQN(B)), ACL-DQN(A), ACL-DQN(B), and ACL-DQN(C) all outperform DQN, where ACL-DQN(C) has the best performance and ACL-DQN(B) has the worst performance in three curriculum schedule. The best scores are labeled in bold.

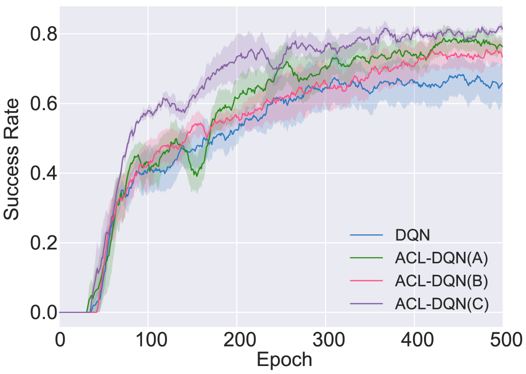


Figure 3: The learning curves of DQN, ACL-DQN(A), ACL-DQN(B), and ACL-DQN(C).

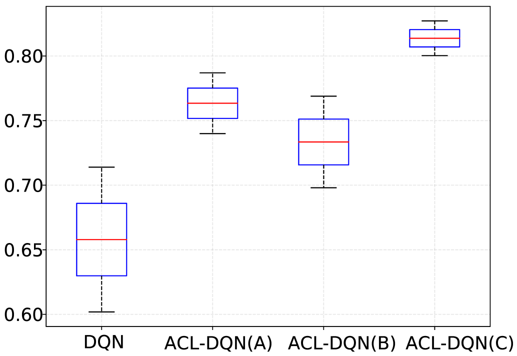


Figure 4: The stability of DQN, ACL-DQN(A), ACL-DQN(B), and ACL-DQN(C) about average success rate at 500 epoch.

is clearly observed that ACL-DQN(A), ACL-DQN(B), and DQN-ACL(C) are more stable than DQN, where the average success rate of ACL-DQN(C) has stabilized above 0.8 while the DQN still fluctuates substantially around 0.65. The result shows that ACL-DQN under the guidance of the teacher policy model shows a more effective and stable performance and the ACL-DQN(C) agent with the strength of supervision and controllability performs best and most stable.

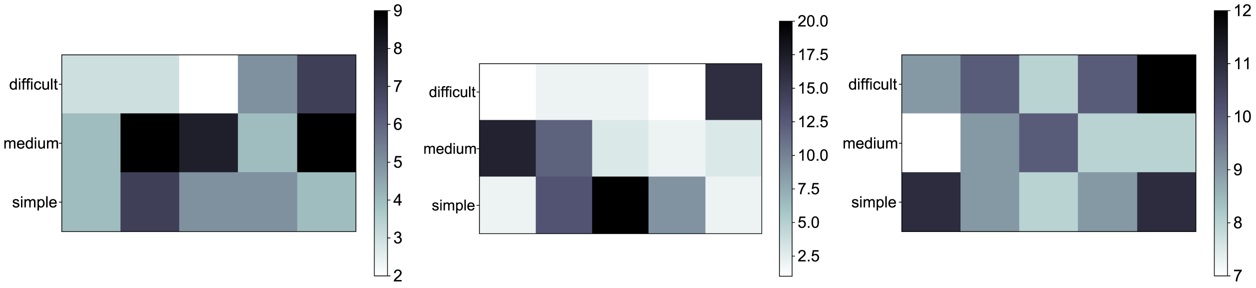
Mastery threshold of ACL-DQN(C) Choosing a new difficulty user goal set is allowed in ACL-DQN(C), if and only if the success rate of sampled user goals in the same difficulty has exceeded the ”mastery” threshold within a continuous-time T (details in Algorithm 3). Intuitively, if the threshold is too small, student agents will enter the learning of the harder goals before they mastered the simpler goals. The student agent is easy to collapse because it is difficult to learn positive training dialogues in time. If the threshold is too big, the student agent will continue to learn the remaining simple goals even if they have mastered the simple goals, slowing down the efficiency of student agent learning.

Figure 6 depicts the influences of different thresholds. As expected, when the threshold is too high or too small, it is difficult for student agents to lean a good strategy, and the learning rate of them is not as good as using a threshold within the range of $[0.5, 0.6]$. The result here can serve as a reference to ACL-DQN(C) practitioners.

Ablation Test To further examine the efficiency of the over-repetition penalty module, we conduct an ablation test by removing this module, referred to as ACL-DQN/-ORP. In order to observe the influence of the over-repetition penalty module more clearly, we take ACL-DQN(A) as an example to compare with ACL-DQN/-ORP and DQN with traditional randomly sampled. We choose five user goals from the C_{simple}, C_{medium} and $C_{difficult}$ and divide them into three groups according to their difficulty. The heat maps of three different methods (DQN, ACL-DQN(A)/-ORP, and ACL-DQN(A)) are displayed in Figure 5, where the color of grid reflects the number of the selected user goals. The darker the color, the more times the user goals have been selected. It is clear that the simple number in Figure 5a is almost the same. But a serious imbalance phenomenon appears in Figure 5b, which does ha

Human Evaluation

We recruited real users to evaluate different systems by interacting with different systems, without knowing which the agent is hidden from the users. At the beginning of each dialogue session, the user randomly picked one of the agents to converse using a random user goal. The user can terminate the dialogue at any time if the user deems that the dialogue is too procrastinated and it is almost impossible to achieve their goal. Such dialogue sessions are considered as failed.



(a) DQN.

(b) ACL-DQN(A) w/o -ORP.

(c) ACL-DQN(A).

Figure 5: Heat maps of the number of the selected user goal in three different methods: (a) DQN, (b) ACL-DQN(A)/-ORP, (c) ACL-DQN(A). The depth of color in each image represents the number of times the goals has been select.

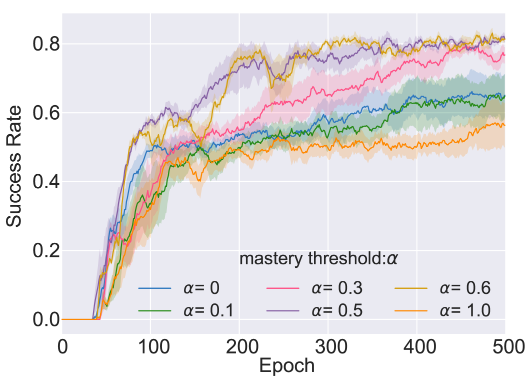


Figure 6: The *mastery* in ACL-DQN(C): mastery threshold α in $[0.5, 0.6]$ performs the best.

For the stability of different systems, each time the system was given a score (1-10), where the process was repeated 20 times. The greater the variance, the more unstable the system was.

Four agents (DQN, ACL-DQN(A), ACL-DQN(B), and ACL-DQN(C)) trained as previously described (Figure 3) at epoch 200 [3] are selected for human evaluation. As illustrated in Figure 7, the results of human evaluation confirm what we observed in the simulation evaluations. We find that DQN is abandoned more often due to its unstable performance, and it takes so many turns to reach a promising result in the face of more complex tasks (Figure 3), ACL-DQN(B) is kept not good enough since they could not adapt the harder goal quickly and the ACL-DQN(C) outperforms all the other agents. For the stability of different systems, the experimental results show that the variance of three ACL-DQN methods are all small than baselines, which means our methods are more stable, and ACL-DQN combined with the curriculum schedule C is the most stable one.

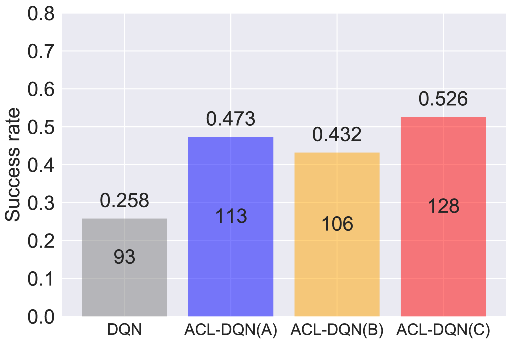


Figure 7: Human evaluation results of DQN, ACL-DQN(A), ACL-DQN(B), and ACL-DQN(C), the number of test dialogues indicated on each bar.

Conclusion

In this paper, we propose a novel framework, Automatic Curriculum Learning-based Deep Q-Network (ACL-DQN), to innovatively integrate curriculum learning and deep reinforcement learning in dialogue policy learning. We design a teacher model that automatically arranges and adjusts the sampling order of user goals without any requirement of prior knowledge to replace the traditional random sampling method in user simulators. Sampling the user goals that match the ability of student agents regarding the difficulty of each user goal, maximizes and stabilizes student agents learning progress. The learning progress of the student agent and the over-repetition penalty as the criteria of the sampling order of each user goal, guarantee both of the sampled efficiency and diversity. The experimental results demonstrate the efficiency and stability of the proposed ACL-DQN. Besides, the proposed method has strong generalizability, because it can be further improved by equipping with curriculum schedules. In the future, we plan to explore the factors in the curriculum schedules that have a pivotal impact on dialogue policy learning, and evaluate the efficiency and stability of our approach by adopting different types of curriculum schedules.

[3] Epoch 200 is picked since we are testing the efficiency of methods using a small number of real experiences.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback on the work, and we would like to acknowledge to volunteers from South China University of Technology for helping us with the human experiments. This work was supported by the Key-Area Research and Development Program of Guangdong Province,China (Grant No.2019B0101540042) and the Natural Science Foundation of Guangdong Province,China (Grant No.2019A1515011792).

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In Danyluk, A. P.; Bottou, L.; and Littman, M. L., eds., *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, 41–48. ACM.

Budzianowski, P.; Ultes, S.; Su, P.; Mrksic, N.; Wen, T.; Casanueva, I.; Rojas-Barahona, L. M.; and Gasic, M. 2017. Sub-domain Modelling for Dialogue Management with Hierarchical Reinforcement Learning. In Jokinen, K.; Stede, M.; DeVault, D.; and Louis, A., eds., *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 86–92. Association for Computational Linguistics.

Chen, L.; Yang, R.; Chang, C.; Ye, Z.; Zhou, X.; and Yu, K. 2017a. On-line Dialogue Policy Learning with Companion Teaching. In Lapata, M.; Blunsom, P.; and Koller, A., eds., *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, 198–204. Association for Computational Linguistics.

Chen, L.; Zhou, X.; Chang, C.; Yang, R.; and Yu, K. 2017b. Agent-Aware Dropout DQN for Safe and Efficient On-line Dialogue Policy Learning. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2454–2464. Association for Computational Linguistics.

De Bruin, T.; Kober, J.; Tuyls, K.; and Babuška, R. 2015. The importance of experience replay database composition in deep reinforcement learning. In *Deep reinforcement learning workshop, NIPS*.

Dhingra, B.; Li, L.; Li, X.; Gao, J.; Chen, Y.; Ahmed, F.; and Deng, L. 2017. Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 484–495. Association for Computational Linguistics.

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J. 2016. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 3675–3683.

Li, X.; Chen, Y.; Li, L.; Gao, J.; and Çelikyilmaz, A. 2017. End-to-End Task-Completion Neural Dialogue Systems. In Kondrak, G.; and Watanabe, T., eds., *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, 733–743. Asian Federation of Natural Language Processing.

Li, X.; Lipton, Z. C.; Dhingra, B.; Li, L.; Gao, J.; and Chen, Y. 2016. A User Simulator for Task-Completion Dialogues. *CoRR* abs/1612.05688.

Lipton, Z. C.; Gao, J.; Li, L.; Li, X.; Ahmed, F.; and Deng, L. 2016. Efficient Exploration for Dialog Policy Learning with Deep BBQ Networks \& Replay Buffer Spiking. *CoRR* abs/1608.05081.

Liu, B.; and Lane, I. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*, 482–489. IEEE.

Lu, K.; Zhang, S.; and Chen, X. 2019. Goal-Oriented Dialogue Policy Learning from Failures. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 2596–2603. AAAI Press.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nat.* 518(7540): 529–533.

Peng, B.; Li, X.; Gao, J.; Liu, J.; Chen, Y.; and Wong, K. 2018a. Adversarial Advantage Actor-Critic Model for Task-Completion Dialogue Policy Learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 6149–6153. IEEE.

Peng, B.; Li, X.; Gao, J.; Liu, J.; and Wong, K. 2018b. Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2182–2192. Association for Computational Linguistics.

Peng, B.; Li, X.; Li, L.; Gao, J.; Çelikyilmaz, A.; Lee, S.; and Wong, K. 2017. Composite Task-Completion Dialogue Policy Learning via Hierarchical Deep Reinforcement Learning. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2231–2240. Association for Computational Linguistics.

Ren, Z.; Dong, D.; Li, H.; and Chen, C. 2018. Self-Paced Prioritized Curriculum Learning With Coverage Penalty in Deep Reinforcement Learning. *IEEE Trans. Neural Networks Learn. Syst.* 29(6): 2216–2226.

Schatzmann, J.; Thomson, B.; Weilhammer, K.; Ye, H.; and Young, S. J. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In Sidner, C. L.; Schultz, T.; Stone, M.; and Zhai, C., eds., *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, 149–152. The Association for Computational Linguistics.

Schatzmann, J.; and Young, S. J. 2009. The Hidden Agenda User Simulation Model. *IEEE Trans. Speech Audio Process.* 17(4): 733–747.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T. P.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the game of Go with deep neural networks and tree search. *Nat.* 529(7587): 484–489.

Su, P.; Gasic, M.; Mrksic, N.; Rojas-Barahona, L. M.; Ultes, S.; Vandyke, D.; Wen, T.; and Young, S. J. 2016a. Continuously Learning Neural Dialogue Management. *CoRR* abs/1606.02689.

Su, P.; Gasic, M.; Mrksic, N.; Rojas-Barahona, L. M.; Ultes, S.; Vandyke, D.; Wen, T.; and Young, S. J. 2016b. On-line Active Reward Learning for Policy Optimisation in Spoken Dialogue Systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Su, S.; Li, X.; Gao, J.; Liu, J.; and Chen, Y. 2018. Discriminative Deep Dyna-Q: Robust Planning for Dialogue Policy Learning. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 3813–3823. Association for Computational Linguistics.

Sutton, R. S.; and Barto, A. G. 1998. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Networks* 9(5): 1054–1054.

Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Wu, Y.; Li, X.; Liu, J.; Gao, J.; and Yang, Y. 2019. Switch-Based Active Deep Dyna-Q: Efficient Adaptive Planning for Task-Completion Dialogue Policy Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 7289–7296. AAAI Press.

Young, S. J.; Gasic, M.; Thomson, B.; and Williams, J. D. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE* 101(5): 1160–1179.

Zhao, Y.; Wang, Z.; Yin, K.; Zhang, R.; Huang, Z.; and Wang, P. 2020. Dynamic Reward-Based Dueling Deep Dyna-Q: Robust Policy Learning in Noisy Environments. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI, New York, NY, USA, February 7-12, 2020*, 9676–9684. AAAI Press.