Dynamic Reward-Based Dueling Deep Dyna-Q: Robust Policy Learning in Noisy Environments

Yangyang Zhao, Zhenyu Wang,[*] Kai Yin, Rui Zhang, Zhenhua Huang, Pei Wang
School of Software, South China University of Technology
{msyyz, sekaiyin, sewangpei}@mail.scut.edu.cn, wangzy@scut.edu.cn,
zhang1rui4@outlook.com, zhhuangscut@gmail.com

Abstract

Task-oriented dialogue systems provide a convenient interface to help users complete tasks. An important consideration for task-oriented dialogue systems is the ability to against the noise commonly existed in the real-world conversation. Both rule-based strategies and statistical modeling techniques can solve noise problems, but they are costly. In this paper, we propose a new approach, called Dynamic Reward-based Dueling Deep Dyna-Q (DR-D3Q). The DR-D3Q can learn policies in noise robustly, and it is easy to implement by combining *dynamic reward* and the Dueling Deep Q-Network (Dueling DQN) into Deep Dyna-Q (DDQ) framework. The Dueling DQN can mitigate the negative impact of noise on learning policies, but it is inapplicable to dialogue domain due to different reward mechanisms. Unlike typical dialogue reward function, we integrate *dynamic reward* that provides reward in real-time for agent to make Dueling DQN adapt to dialogue domain. For the purpose of supplementing the limited amount of real user experiences, we take the DDQ framework as the basic framework. Experiments using simulation and human evaluation show that the DR-D3Q significantly improve the performance of policy learning tasks in noisy environments.[1]

1 Introduction

Task-oriented dialogue systems aim at assisting users to solve a task with fewer turns and have been used in a variety of applications (Dhingra et al. 2017; Li et al. 2017; Gao, Galley, and Li 2019). Dialogue policy, which can select appropriate dialogue actions to respond and steer the conversation, is the key component of task-oriented dialogue systems. The noise in real-world conversation include automatic speech recognition (ASR) or natural language understanding (NLU) errors and ambiguous user utterances. This noise adds the complexity of real dialogue tasks, increasing the challenge of designing dialogue policies that can robustly handle noise. A dialogue system should be able to carry on a conversation without the luxury of the accurate ASR, NLU, or precise user utterances (Paek and Horvitz 2000). Therefore, designing a robust dialogue policy is crucial for the application of task-oriented in real-world environments.

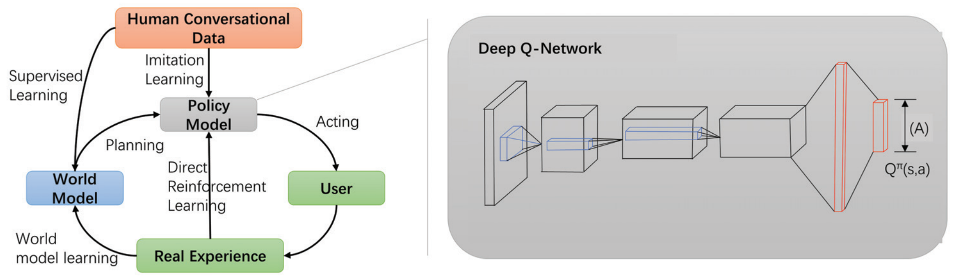
The idea of dialogue policy learning for noise is not new (Schatzmann et al. 2006). It has been suggested that both the rule-based strategies and statistical modeling techniques offer a natural framework for modeling noise and support policies which are robust to their effects (Young et al. 2013; Bohus and Rudnicky 2009). Some statistical-modeling techniques generally model dialogue policies as a reinforcement learning (RL) problem which requires a huge amount of interactions between the dialogue system and real users (Fazel-Zarandi et al. 2017; Guo et al. 2019). Although user simulators provide an inexpensive alternative (Li et al. 2016; Williams, Asadi, and Zweig 2017; **?**), there always exists discrepancies between real users and user simulators (Young et al. 2016).

To alleviate the biases in the design of user simulators, the Deep Dyna-Q (DDQ) has been proposed recently which integrates planning into RL for dialogue policy learning (Peng et al. 2018). As illustrated in Figure 1a, the policy of the DDQ agent can be improved through both real user experiences via direct RL and simulated experiences via planning. However, the effectiveness of DDQ depends upon the quality of simulated experiences. Although some DDQ variants further incorporate discriminators (Su et al. 2018), active learning (Wu et al. 2019) and Budget-Conscious Scheduling (BCS) (Zhang et al. 2019) to obtain high-quality simulated experiences, they are only suitable in simulated environment without noise (Li et al. 2017). Noise in the real-world environments hurts their performance badly. Moreover, in practice the noise problems are exacerbated in domains where multi-step planning is used (Su et al. 2017).

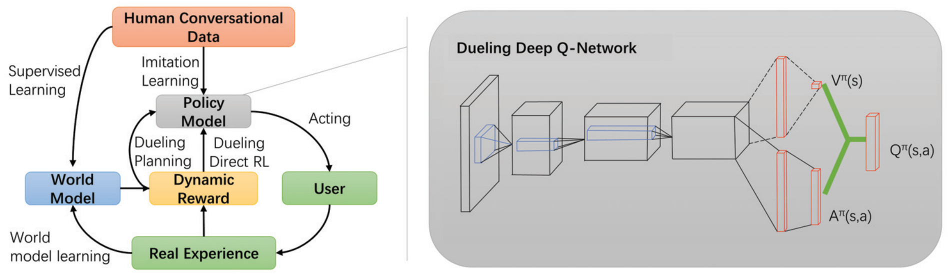
Existing research on Dueling Deep Q-Network (Dueling DQN) has shown that, in Atari 2600 tasks, an RL agent can lead to better policy evaluation in noise conditions from many similar-valued actions by automatically producing separate estimates of the state value function $V(s)$ and the state-dependent action advantage function $A(s, a)$ (Wang et al. 2016). In this paper, we attempt to utilize Dueling DQN method to mitigate the negative impact of noise in real dialogue environments.

[*]Corresponding author.

[1]Source code is at https://github.com/zhaoyangyangHH/DR-D3Q.



(a) DDQ framework



(b) Proposed DR-D3Q framework

Figure 1: Designs of RL agents for dialogue policy learning in task-completion dilaogue systems

Although it may seem intuitive to simply apply Dueling DQN to dialogue domain, there are still problems needing solution. The vanilla Dueling DQN method is not directly applicable to dialogue domain, for the reason that the typical dialogue reward function is such that the agent only receives a positive reward for success ,and otherwise it gets negative rewards for failure or per turn taken to encourage shorter dialogs dialogues (whereas the game agent used in Atari 2600 tasks gets positive rewards for each subgoal). Taking the typical dialogue reward function, the immediate rewards obtained often are the same using different actions in the same state, resulting in the inability to learn $V(s)$ and $A(s, a)$ effectively. Therefore, the typical dialogue reward function limits the applicability of the Dueling DQN in dialogue tasks.

To solve the above problems, we propose Dynamic Reward-based Dueling Deep Dyna-Q (DR-D3Q), providing an effective and robust mechanism to against noise. As illustrated in Figure 1b, we incorporate *dynamic reward* to provide rewards in real-time for agent, which helps the vanilla Dueling DQN adapt to the dialogue domain. The Dueling DQN based on dynamic reward, called DR-Dueling DQN, is characterized by giving a dynamic reward according to the complexity of the subgoals completed by the current dialogue segment to encourage the dialogue agent to achieve more challenging goals (the more complex subgoals, the higher the corresponding rewards). Moreover, our approach is based on the DDQ framework where the DR-Dueling DQN is applied to replace native DQN in DDQ framework to supplement the limited amount of real user experiences. The policy of the DR-D3Q agent can be improved through either *dueling indirect RL* or *dueling planning*. Experiments show that our method is robust and effective in the face of noise and achieve better performance than DDQ in noise-free environments. In summary, our main contribution in this work are as follows:

- We propose Dynamic Reward-based Dueling Deep Dyna-Q. As far as we know, this is the first work that applies the Dueling DQN idea to the problem of dialogue policy learning in noisy environments.
- We introduce a new reward function (i.e., dynamic reward) to measure the quality of each dialogue action in real-time.
- We conduct extensive experiments on the movie-ticket booking task for different noise level. The results show that our model outperforms the state-of-the-art methods.

2 Model Architecture

As illustrated in Figure 2, the DR-D3Q framework consists of six modules: (1) a LSTM-based NLU module (Hakkani-Tür et al. 2016) for converting user's raw utterance to the semantic form of dialogue acts; (2) a state tracker (Mrksic et al. 2017) for tracking the dialogue states; (3) a dialogue policy relies on the current state provided by the state tracker to select an action; (4) a model-based natural language generation (NLG) module (Wen et al. 2015) for converting dialogue actions into a natural language response; (5) a world model for generating simulated user actions and simulated rewards; and (6) dynamic reward module provides dynamic rewards for dialogue agent to rate the quality of each dialogue action in real-time during the conversation.

Figure 1b illustrates the training of the DR-D3Q agent comprises four stages: (1) *dynamic reward obtaining*: the

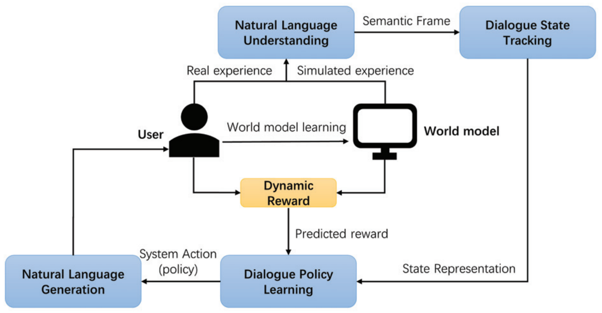


Figure 2: Illustration of the proposed DR-D3Q dialogue system framework

dynamic rewards are obtained by judging whether the dialogue segment from the real and simulator experiences accomplishes the user subgoals in real-time. (2) *Dueling Direct Reinforcement Learning*: the agent interacts with real users directly, where the generated real experiences are used to improve the dialogue policy; (3) *Dueling planning*: the agent interacts with the world model and improves the policy using the simulator experiences; (4) *world model learning*: the world model updates itself using real experience. Each stage is detailed in the subsections below.

2.1 Dynamic Reward

To obtain dynamic rewards requires three steps: *Definition of Dialogue Subgoal*: we introduce the concept of dialogue subgoal; *Dynamic Segmentation Algorithm*: we present a dynamic segmentation algorithm that efficiently segment *valid* subgoals; *Dynamic Reward Obtaining*: calculate dynamic predicted rewards based on the complexity of valid subgoals.

Definition of Dialogue Subgoal Task-oriented dialogue systems assist users to solve a task, where have an objective goal G. The goal G includes a set of constraints C and a set of requests $R : G = (C, R)$ (Schatzmann and Young 2009).

Consider a air-ticket booking domain. A user may ask about the *departure time* and *prices* of a *tomorrow*'s air-ticket from *Beijing* to *Guangzhou*,where the goal is in the form of:

$$\textbf{Goal} = \left(C = \begin{bmatrix} location\_from = Beijing \\ location\_to = Guangzhou \\ date = tomorrow \end{bmatrix}, \quad R = \begin{bmatrix} air-ticket\_price = \\ departure\_time = \end{bmatrix} \right) \tag{1}$$

Definition 1. ***Subgoal***[2] *Given $G = (C, R)$ and $G' = (C', R')$, we say $G'$is a subgoal of G, or $G' \sqsubset G$, if $C' \subset C$and $R' \subset R$, where $G' \neq \emptyset$.*

[2]Our definition of *subgoal* is provided by two complex HER methods (Lu, Zhang, and Chen 2019).

$$\textbf{Subgoal 1} = \left(C = \emptyset, R = \begin{bmatrix} air-ticket\_price = \\ departure\_time = \end{bmatrix} \right)$$
$$\textbf{Subgoal 2} = \left(C = \begin{bmatrix} location\_from = Beijing \\ location\_to = Guangzhou \end{bmatrix}, R = \begin{bmatrix} air-ticket\_price = \end{bmatrix} \right) \tag{2}$$

According to Definition 1, Equation 2 shows two example subgoals (out of many) in the air-ticket booking example. For instance, Subgoal 2 corresponds to the request of "*What is the price of the air-ticket from Beijing to Guangzhou ?*"

Typically, in a successful dialogue, the agent gets a $2L$ reward (L is the maximum length of a dialogue), otherwise it gets a $-L$ reward, which encourages the agent complete the entire user goal. Furthermore, in each turn, the agent receives a reward of -1. But if the user's goal is not completed, it is considered that the dialogue is failed, which lacks any explicit guidance for agents on how to drive the dialogue (Barlier, Laroche, and Pietquin 2018).

It should be noted that some subgoals do not make sense to users, but can be useful for dialogue learning. For instance, Subgoal 1 corresponds to a query about *air-ticket_price* and *departure_time* without any constraints. Real users do not have such goals, but an agent can still learn from the experience of achieving such subgoals.

Continuing the "air-ticket booking" example, if the agent has not achieved *sugoal 2*, the dialogues will be deemed unsuccessful, meaning that the agent cannot learn much from it, even though the agent has correctly achieved the other subgoals. In this work, we not only make use of the successful dialogues that the agent has achieved the whole goal, but also leverage such unsuccessful dialogues that the agent has achieved partial subgoals in the training process.

Dynamic Segmentation Algorithm Give dialogue D and D', we say D' is a segment of D, if D' includes a consecutive sequence of turns of D. We introduce an assessment function, $success(G, D)$ provided by dialogue simulators , that outputs *true* or *false* representing whether dialogue D accomplishes goal (or subgoal) G or not. Using the dialogue segments and the assessment function, we define the validity of subgoals:

Definition 2. ***Validity*** *of subgoals: Given dialogue D, user goal G, and dialogue segment D' (of D). If there exists a subgoal $G' \subset G$, and $success(G', D')$ is true, we say G' is a valid subgoal.*

Using Definition 1 and 2, we can obtain the valid subgoal G'. Due to the combinatorial explosion, there are many subgoals of the entire goal, making it infeasible to assess the validity of the subgoal using all subgoals. Formally, Given goal G, the number of subgoals N_g is shown in Equation 3. For instance, if $|C| = 5$ and $|R| = 5$, the number of subgoals N_g is 1598.

$$N_g = \sum\nolimits_{i=0}^{|C|} \binom{i}{|C|} \cdot \sum\nolimits_{j=0}^{|R|} \binom{j}{|R|} - 2 \tag{3}$$

Algorithm 1 Dynamic Dialogue Segmentation

```
Require: Dialogue $D$ from real and simulator experience; Entire user goal $G = (C, R)$; Assessment function $success(\cdot, \cdot)$;
Ensure: A collection of valid subgoal, $\Omega$;
 1: Intialize $\mathbb{G}' = \emptyset$, and $\mathbb{G} = C \cup R$;
 2: Intialize $\Omega = \emptyset$;
 3: while Dialogue $D$ in not ended do
 4:     Outcome flag $segment\_outcome = False$
 5:     for $q \in \mathbb{G} \cap q \notin \mathbb{G}'$ do
 6:         Construct a subgoal $G' = \mathbb{G}' \cup q$
 7:         if $success(G', D')$ then
 8:             $segment\_outcome = True$
 9:             $\mathbb{G}' \leftarrow \mathbb{G}' \cup q$
10:         end if
11:     end for
12:     if $segment\_outcome = True$ then
13:         $\Omega \leftarrow \Omega \cup \mathbb{G}'$
14:     end if
15: end while
```

Based on previous research on dialogue policy learning (Schatzmann et al. 2007), we aim at using only the ones with the so-far-highest ”cardinality”, instead of assessing the validity of a dialogue segment using exhaustive subgoal. First, we initialize two collections $\mathbb{G}'$ and $\mathbb{G}$. $\mathbb{G}'$ is the subgoal set accomplished by dialogue segments, and $\mathbb{G}$ is the entire user goal set. Therefore, at the beginning of dialogues $\mathbb{G}'$ is an empyty set, and $\mathbb{G}$ stores all constrains and requests. Formally, $\mathbb{G}' = (C', R') = \emptyset$, and $\mathbb{G} = (C, R)$. After each dialogue turn, we will leverage the NLU to identify the slot q of the current dialogue turn. If the slot meets the condition that the slot belongs to the set $\mathbb{G}$ but is not one of the set $\mathbb{G}'$, we update sugoal set $\mathbb{G}'$ and continue until the dialogue is ended. Therefore, we get a subgoal set $\mathbb{G}'$ where all subgoals share the same (so far highest) cardinality. Formally, $\mathbb{G}' = \{G' \mid G' = \mathbb{G}' \cup q, \forall q \in \mathbb{G} \cap q \notin \mathbb{G}'\}$. If $G' \in \mathbb{G}$ is accomplished by the current dialogue segment, the corresponding q is add into $\mathbb{G}'$, which is used to generate the subgoal set for the next dialogue segment. So at each dialogue turn, only a small set of subgoals is used to assess a dialogue segment. More detailed procedure is shown in Algorithm 1. Compared to exhaustive subgoal identification that suffers form combinatorial explosion, our dynamic segment algorithm has $O(|C| + |R|)$ time complexity.

Dynamic Reward Obtaining Building on our dynamic segment algorithm, we utilize valid subgoals to generate dynamic rewards for dueling direct reinforcement learning and planning. We want to encourage the agent to accomplish more challenging subgoals using positive reward, while avoiding the agent sticking to accomplishing only the simple subgoals. Given R_{max} and R_{min} being the reward and penalty to successful and failed dialogues with users, we design the following dynamic reward function for agents:

$$R(D') = \alpha \cdot |G'|, ensuring \alpha \cdot |G'| < R_{max} \quad (4)$$

where α is a weight, G' is a subgoal, D' is a dialogue segment, and $|G'|$ is the number of slots of G' that have be identified. The agent receives a big positive reward to encourage accomplish entire user goal, receives a small penalty (-1 in our case) each turn to encourage shorter dialogues, and it will be punished when the dialogue fails.

2.2 Dueling Direct Reinforcement Learning and Dueling Planning

In this stage, we employ a variant of DQN method, Dueling DQN method to improve the dialogue policy. Both the dueling direct reinforcement learning and dueling planning are implemented by the Dueling DQN method, operating on real experience in B^u for dueling direct reinforcement learning and on simulator experience in B^s for dueling planning.

The Dueling DQN architecture is decomposed into two separate streams: one for the state value function V and one for the state-dependent action advantage function A. Feature learning is the same as DQN is carried out by a number of convolutional and pooling layers. The activations of the last of these layers are sent to both separate streams which contains a number of fully-connected layers. The final layer combines the output of the two streams, and the outputs of the network is a set of Q values, one for each action. The aggregator for the two outputs of the advantage and value streams is:

$$\begin{aligned} Q(s, a; \theta, \alpha, \beta) =& V(s; \theta, \beta) + (A(s, a; \theta, \alpha) \\ & - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha)) \end{aligned} \quad (5)$$

β refers to the parameters specific to the value network, the α refers to the parameters specific to the advantage network, and the θ refers to the parameters to the function $Q(\cdot)$ by a Multi-Layer Perceptron (MLP).

Typically, we consider task-oriented dialogue as a Markov Decision Process (MDP), where the agent interacts with a user or simulator through a sequence of actions to accomplish a user goal. In each step, the agent observes the state s and selects an action a to further accomplish user goal, using an $\epsilon - greedy$ policy, that selects a random action with probability ϵ or otherwise the action that maximizes the $Q(s, a; \theta, \alpha, \beta)$ function. Afterwards, the agent receives a reward r provided by *dynamic reward*, and a corresponding response, updates the dialogue state to s'. Finally, we store the experience (s, a, r, s') into the real experience buffer B^u or simulator experience buffer B^s respectively. This cycle continues until the dialogue terminates.

We adjust the parameter θ_Q by minimizing the mean-squared loss function to optimize the value function $Q(\cdot)$ as follows:

$$\begin{aligned} &\mathcal{L}_i(\theta_Q) = \\ &\qquad \mathbb{E}_{(s,a,r,s') \sim B^u \cup B^s}[(y_i - Q(s, a; \theta_Q, \alpha, \beta))^2] \quad (6) \\ &y_i = r + R(D') + \gamma \mathbb{E}_{a' \sim \pi(s')}[Q(s', a'; \theta_{Q'}, \alpha', \beta')] \end{aligned}$$

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 $\nabla_{\theta_Q} \mathcal{L}(\theta_Q)$ by back-propagation and mini-batch gradient descent.

2.3 World Model Learning

We utilize the same design of the world model in DDQ model (Peng et al. 2018). Specifically, the world model $M(s, a; \theta_M)$ is trained using a multi-task deep neutral network (Liu et al. 2015) to generate the simulated experiences that can be used to improve dialogue policy and enable the dueling planning.

In each turn of dialogue, the world model takes the current dialogue state s and the last system action a as the input, through an MLP generates the corresponding user response o, reward r, and a binary variable t which indicates whether the dialogue terminates. The MLP has a common sharing representation in the first layer (referred to as layer h). The computation for each term can be shown as below:

$$h = \tanh(W_h(s, a) + b_h), \tag{7}$$

$$r = W_r h + b_r, \tag{8}$$

$$o = \text{softmax}(W_a h + b_a), \tag{9}$$

$$t = \text{sigmoid}(W_t h + b_t) \tag{10}$$

3 Experiments

We evaluate the proposed DR-D3Q method on a movie-ticket booking task with both simulated users and real users in a noise-free environment and noisy environments, we focus on comparing the results of different methods in the noisy environments.

3.1 Dataset

In the experiment, we use a movie-ticket booking dataset which contains raw conversational data collected via Amazon Mechanical Turk. The dataset has been manually labeled based on a schema defined by domain experts, as shown in Table 1, consisting of 11 intents and 16 slots. We simulate the noisy environments by setting the value of *slot_error* in dataset. In total, the dataset contains 280 annotated dialogues, the average length of which is approximately 11 turns.

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

Table 1: The data annotation schema

3.2 Baselines

To evaluate the effectiveness of the DR-D3Q agent, we have developed different baselines to compare with:

- The **DQN** agents are implemented with only direct reinforcement learning in each epoch.

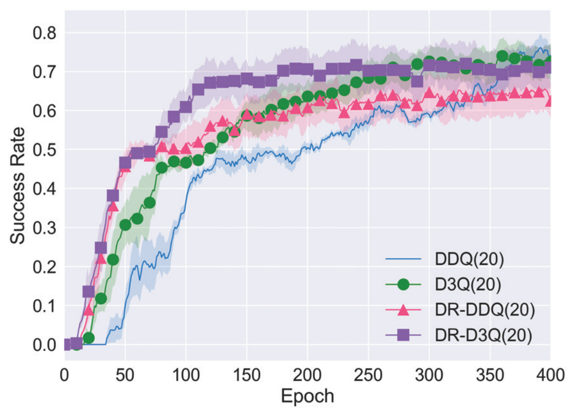


Figure 3: The learning curves of DDQ(K) without noise, where $(K - 1)$ denotes the number of planning steps. The DQN agent is identical to a DDQ(K) agent with $K = 0$.

- The **DQN**(K) has $(K - 1)$ times more real experiences that the DQN agent. The performance of DQN(K) can be viewed as the upper bound of DDQ(K), with the same number of planning steps $(K - 1)$, as they have the same training settings and the same amount of training samples during the entire learning process.
- The **DDQ**(K) agents are trained using an initial world model pre-trained on human conversational data and a DQN network for direct RL and planning, with $(K - 1)$ planning steps.
- The proposed **DR-D3Q**(K) agents are trained by utilizing a Dueling DQN network based dynamic reward.

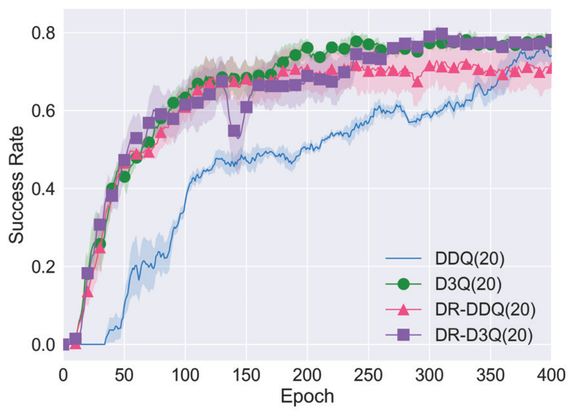
3.3 Implementation Details

Agent and Hyper-parameters Settings For all the models the world models (DQN, DDQ, and DR-D3Q) and their variants, we use MLPs to parameterize the value networks $Q(\cdot)$ with one hidden layer of size 80 and ReLU activation. We simulate the noise level of the environment s by setting the value of the *slot_error_prob*. ϵ-greedy is always applied for exploration. We set the discount factor $\gamma = 0.9$. The buffer size of B^u and B^s is set to 2000 and $2000 \times K$ *planning steps*, 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 30 turns ($L = 30$). 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. 2018) at the beginning stage to build a naive but occasionally successful rule-based agent based on 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.

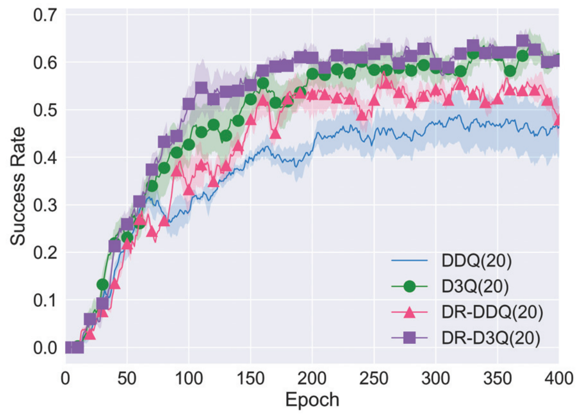
World Model For all the models the world models (DDQ and DR-D3Q) and their variants, the $M(\cdot)$ are MLPs with

Agent	noise	Epoch = 100			Epoch = 200			Epoch = 300			Epoch = 400		
		Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns
DQN	0	0.3439	-11.19	26.27	0.4998	4.28	23.39	0.5909	13.37	21.64	0.7385	28.27	18.39
DQN(20)		*0.6288*	*17.56*	*20.05*	*0.7590*	*30.63*	*17.37*	*0.7672*	*31.96*	*16.17*	*0.7810*	*33.73*	*15.12*
DDQ(5)		0.4699	1.49	23.60	0.6284	17.53	20.04	0.7174	26.72	17.70	0.7257	27.75	17.12
DDQ(10)		0.5090	5.26	23.10	0.6019	14.81	20.72	0.6417	19.21	19.08	0.6319	18.47	18.82
DDQ(20)		0.6072	15.41	20.47	**0.7043**	25.06	18.66	0.7080	25.79	17.86	0.7126	26.66	16.94
DR-D3Q(20)		**0.6183**	15.94	21.42	0.6812	22.67	19.27	**0.7899**	33.75	16.68	**0.7705**	32.38	15.93
DQN	0.1	0.2814	-16.86	26.46	0.4128	-3.44	23.19	0.4678	1.72	22.76	0.4609	1.30	22.36
DQN(20)		*0.4265*	*-1.97*	*22.71*	*0.5749*	*13.02*	*19.43*	*0.55794*	*13.37*	*19.55*	*0.6162*	*17.20*	*18.50*
DDQ(20)		0.3382	-11.15	25.17	0.5235	7.87	20.48	0.5283	8.57	19.94	0.4742	3.27	20.82
DR-D3Q(20)		**0.5052**	5.86	21.21	**0.6169**	17.46	18.11	**0.6134**	17.40	17.60	**0.6028**	16.38	17.75
DQN	0.2	0.1570	-28.51	22.28	0.2533	-18.23	22.05	0.3956	-5.04	23.28	0.3991	-4.44	22.72
DQN(20)		*0.3552*	*-9.22*	*24.39*	*0.4921*	*5.09*	*20.41*	*0.5158*	*7.65*	*19.55*	*0.5215*	*8.14*	*19.56*
DDQ(20)		0.1628	-28.11	27.51	0.4192	-2.42	22.29	0.4653	2.31	21.12	0.4478	0.77	21.08
DR-D3Q(20)		**0.4654**	1.97	21.83	**0.5440**	10.26	19.40	**0.5769**	13.25	19.34	**0.5565**	11.62	18.92
DQN	0.3	0.1050	-34.27	29.45	0.2703	-17.63	25.94	0.2233	-22.28	26.77	0.2371	-20.91	26.50
DQN(20)		*0.1092*	*-33.83*	*29.32*	*0.3364*	*-10.46*	*23.47*	*0.3290*	*-10.88*	*22.99*	*0.3725*	*-6.72*	*22.49*
DDQ(20)		0.15026	-29.52	28.09	0.2860	-15.49	24.47	0.3124	-12.33	22.90	0.3169	-11.43	21.92
DR-D3Q(20)		**0.3033**	-14.54	25.67	**0.4113**	-2.96	21.96	**0.4466**	0.52	21.36	**0.4120**	-2.65	21.47

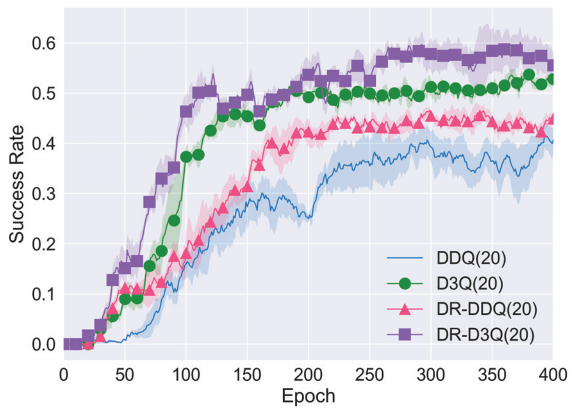
Table 2: Result of different agents at training $epoch = \{100, 200, 300, 400\}$. Each number is averaged over 4 turns, each run tested on 50 dialogues. Success: Evaluated at the same epoch (except one groups: at epoch 200, DDQ(20)), DR-D3Q outperforms DQN and DDQ variants in mean, especially the environment with louder noise, where DQN(20) serves as the upper bound. Best scores are labeled in blue.



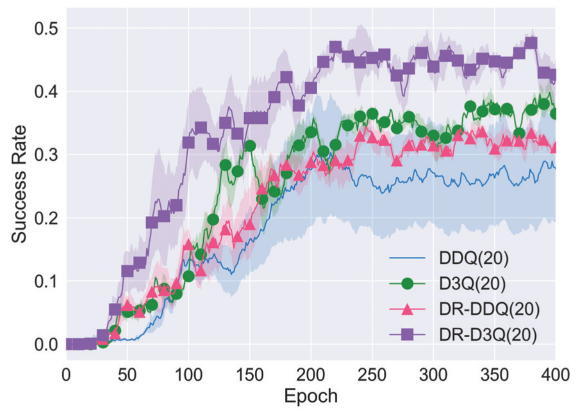
(a) The setting of noise is 0.1



(b) The setting of noise is 0.2



(c) The setting of noise is 0.3



(d) The setting of noise is 0.4

Figure 4: The learning curves of DQN, DQN(20), DDQ(20) and DR-D3Q(20) under different noise settings.

one shared hidden layer of size 160, hyperbolic-tangent activation, and one encoding layer of hidden size 80 for each state and action input.

3.4 Simulation Evaluation

The dialogue agents are trained by interacting with the user simulators instead of real users. In simple terms, the world model is trained to mimic user simulators. The simulation setting allows us to perform a detailed analysis of models without much cost and to reproduce the experimental results easily, in spite of the discrepancy between simulators and real users.

User Simulator We adapted a publicly available task-oriented user simulator (Li et al. 2016) in our simulated evaluation. A dialogue is considered successful if and only if a move ticket is booked successfully and the information provided by the agent satisfies all the constraint slots in the sampled user goal. In our work, we define a dialogue with one (or more) *valid* subgoal is considered *semi-successful*, otherwise it is considered failed. At each completed dialogue, the agent will first receive a subgoal reward $R(D')$ for semi-success, and then receive a corresponding reward based on whether the dialogue is successful or not (a positive reward $2L$ for success, or a negative reward $-L$ for failure). Furthermore, in each turn, a reward -1 is provided to encourage shorter dialogues.

Main Results The main performance results including success rate, average reward and average number of turns over different of models reported in Table 2. As illustrated in Figure 3, a large number of planning steps means leveraging a large amount of simulated experience to train the agents. Hence, the performance of DDQ(K) agents are highly sensitive to parameter K and its improved for all values of K. Therefore, we only keep the best performing DDQ(20) as the baseline in the following figures. We report the main performance detail in Table 2, the results show that the agent of DR-D3Q(20) significantly outperforms the baselines with a higher success rates and a smaller number of interaction turns in different noise level of environments. Even in a noise-free environment, its results are comparable to the best results (DQN(20)).

Figure 4 show the learning curves of different agents trained using different noise setting. From the figures and table, When in the noise-free environment, the difference between DR-D3Q and DDQ may not be significant, where both the DDQ(20) and DR-D3Q(20) agents achieve about 0.68 success rate after 200 epoch and DR-D3Q(20) agents maintains a success rate of over 0.75 after 300 epochs. However, with increase of the noise level, the experimental setting makes the training environment more complicated and unstable than the previous noise-free one, the DDQ agents is degraded significantly, while the performance for DR-D3Q demonstrates a higher degree of robustness to the noise level of environment, where DDQ(20) agents suffer from 30% incorrect slot values and achieve about 0.3 success rate after 250 epochs, DR-D3Q(20) still achieve higher than 0.3 success rate after 100 epoch. The result show that DR-D3Q can be more effective and robust against noisy than DDQ. A potential reason might be that the dueling network architecture could detect the nuances of the action under noisy conditions by extracting the effects of action and state separately for better policy evaluation in such unstable and noisy environments.

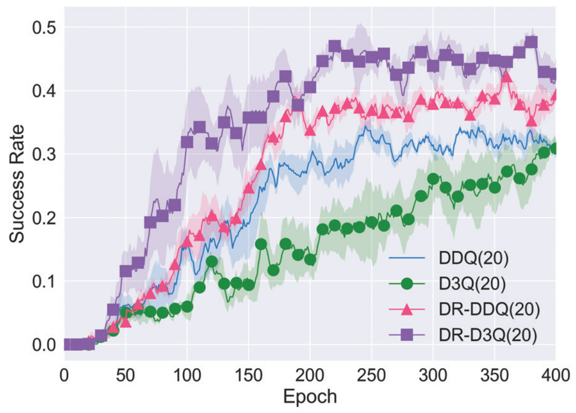


Figure 5: The learning curves of DDQ, D3Q, DR-DDQ and DR-D3Q, where DR-DDQ uses a uniform reward function with DR-D3Q, D3Q uses a original reward function with DDQ under 0.3 noise setting.

Ablation Test To further examine the effectiveness of the dynamic reward module, we conduct an ablation test by replacing original reward function of DDQ with the dynamic reward, referred to as DR-DDQ and updating with a Dueling DQN network with original reward function, referred to as D3Q. In order to observe the influence of dynamic reward module more clearly, we choose to compare models in the noisy environments and set the noise to 0.3. The result in Figure 5 demonstrate that DR-D3Q can consistently outperform DR-DDQ and D3Q, and the performance of the D3Q agent with dynamic reward improves more rapidly. This is due to the fact that there is a gap between the game and the dialogue tasks, where the game usually contains a small reward for each subgoals and the dialogue only receives positive reward for success, resulting in the D3Q agent is more sensible to the reward function in dialogue task.

3.5 Human Evaluation

We recruited real users to evaluate different systems by interacting with different systems without identifying which the agent system is, where the users can give some noise. At the beginning of each dialogue session, the user randomly selected one of the agents to converse using a randomly sampled user goal provided by the corpus and randomly extracts the 30% of slots in user goal for making an error answer deliberately. The user can terminate the dialogue at any time, ”if the user deems that the dialogue is too tedious or repetitive and it is unlikely that they'll complete their goal. In our experiments, such dialogue sessions are considered as failed.

Three agents (DQN, DDQ(20), DR-D3Q(20)) trained in noisy environments (Figure 4) at epoch 200 are selected for human evaluation.[3] As illustrated in Figure 6, the results of

[3]Epoch 200 is picked since we are testing the effectiveness of

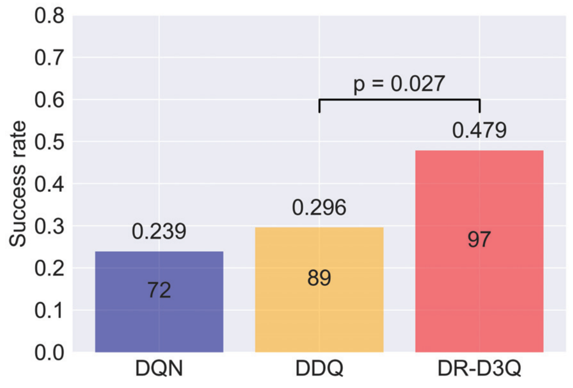


Figure 6: The human evaluation resuts of DQN, DDQ(20), adn DR-D3Q(20) in the noisy settings. The number of dialogues is indicated on each bar and the one-sided p-value is from a two-sample permulation test (difference in mean is significant with $p < 0.05$).

human evaluation confirm what we observed those in the simulation evaluations (Section 3.4). We find that DQN is abandoned more often as it takes so many turns to reach a promising result, DDQ is kept not good enough since they could not adapt the noisy environment and the proposed DR-D3Q outperforms all the other agents.

4 Conclusion

In this work, we developed a new approach Dynamic Reward-based Dueling Deep Dyna-Q (DR-D3Q) for task-oriented dialogue policy learning in noisy environments. Our method mainly solves three problems: (1) Utilizing the DDQ framework to alleviate the problem that the existing methods developed for noise are costly and the discrepancies existed between the real users and user simulators; (2) Utilizing the Dueling DQN to mitigate the negative impact of noise in real dialogue environments; (3) Utilizing Dynamic Reward module to solve the problem that the vanilla Dueling DQN cannot be directly applied to dialogue domain. Validating DR-D3Q on the movie-ticket booking task with simulation experiments and human evaluation, we show that the DR-D3Q agent significantly outperforms the agents trained by other state-of-the-art methods. Furthermore, DR-D3Q can be viewed as a generic model-based RL approach easily-extensible to other RL problems.

This is the first work that applies the Dueling DQN idea to the problem of dialogue policy learning in noisy environment. In the future, we plan to explore the impact of using different kinds of noise on dialogue policy learning and evaluate the robustness of our approach with different types of noise. Furthermore, we will recover the errors from ASR/NLU through other RL algorithm, or investigate other dimensions of dialogue to improve dialogue policy learning.

methods using a small number of real experiences.

Acknowledgement

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 Science and Technology Program of Guangzhou under Grant No. 201802010025, the Industry-University Research Fund Project of Guangzhou under Grant No. 2019PT103, and Guangdong Natural Science Foundation No. 2019A1515011792.

References

Barlier, M.; Laroche, R.; and Pietquin, O. 2018. Training dialogue systems with human advice. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 999–1007.

Bohus, D., and Rudnicky, A. I. 2009. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language* 23(3):332–361.

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 *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.

Fazel-Zarandi, M.; Li, S.-W.; Cao, J.; Casale, J.; Henderson, P.; Whitney, D.; and Geramifard, A. 2017. Learning robust dialog policies in noisy environments. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, ConversationalAI Workshop, 4-9 December 2017, Long Beach, CA, USA*.

Gao, J.; Galley, M.; and Li, L. 2019. Neural approaches to conversational AI. *Foundations and Trends in Information Retrieval* 13(2-3):127–298.

Guo, Y.; Zheng, Y.; Tan, M.; Chen, Q.; Chen, J.; Zhao, P.; and Huang, J. 2019. Nat: Neural architecture transformer for accurate and compact architectures. In *Advances in Neural Information Processing Systems 2019, NIPS*.

Hakkani-Tür, D.; Tür, G.; Çelikyilmaz, A.; Chen, Y.; Gao, J.; Deng, L.; and Wang, Y. 2016. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 715–719.

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.

Li, X.; Chen, Y.; Li, L.; Gao, J.; and Çelikyilmaz, A. 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*, 733–743.

Lipton, Z. C.; Li, X.; Gao, J.; Li, L.; Ahmed, F.; and Deng, L. 2018. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 5237–5244.

Liu, X.; Gao, J.; He, X.; Deng, L.; Duh, K.; and Wang, Y. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, 912–921.

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 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*, 2596–2603.

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

Paek, T., and Horvitz, E. 2000. Conversation as action under uncertainty. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, 455–464.

Peng, B.; Li, X.; Gao, J.; Liu, J.; and Wong, K. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *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.

Schatzmann, J., and Young, S. J. 2009. The hidden agenda user simulation model. *IEEE Trans. Audio, Speech & Language Processing* 17(4):733–747.

Schatzmann, J.; Weilhammer, K.; Stuttle, M. N.; and Young, S. J. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Eng. Review* 21(2):97–126.

Schatzmann, J.; Thomson, B.; Weilhammer, K.; Ye, H.; and Young, S. J. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *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.

Su, P.; Budzianowski, P.; Ultes, S.; Gasic, M.; and Young, S. J. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, August 15-17, 2017*, 147–157.

Su, S.; Li, X.; Gao, J.; Liu, J.; and Chen, Y. 2018. Discriminative deep dyna-q: Robust planning for dialogue policy learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 3813–3823.

Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; and de Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 1995–2003.

Wen, T.; Gasic, M.; Mrksic, N.; Su, P.; Vandyke, D.; and Young, S. J. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 1711–1721.

Williams, J. D.; Asadi, K.; and Zweig, G. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. 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*, 665–677.

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.

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.

Young, S.; Breslin, C.; Gašić, M.; Henderson, M.; Kim, D.; Szummer, M.; Thomson, B.; Tsiakoulis, P.; and Hancock, E. T. 2016. Evaluation of statistical pomdp-based dialogue systems in noisy environments. In *Situated Dialog in Speech-Based Human-Computer Interaction*. 3–14.

Zhang, Z.; Li, X.; Gao, J.; and Chen, E. 2019. Budgeted policy learning for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, 3742–3751.