Neurocomputing 459 (2021) 122–130

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Emotion-sensitive deep dyna-Q learning for task-completion dialogue policy learning

Rui Zhang, Zhenyu Wang *, Mengdan Zheng, Yangyang Zhao, Zhenhua Huang

Department of Software Engineering, South China University of Technology, Guangzhou, Guangdong, PR China

ARTICLE INFO

Article history:
Received 18 June 2020
Revised 7 June 2021
Accepted 25 June 2021
Available online 29 June 2021
Communicated by Zidong Wang

Keywords:
Dialogue policy learning
Deep reinforcement learning
Emotional intelligence
Neural networks

ABSTRACT

In recent years, task-oriented dialogue systems have received extensive attention from academia and industry. Training a dialogue agent through reinforcement learning is often costly because it requires many interactions with real users. Although the Deep Dyna-Q (DDQ) framework uses simulation experience to alleviate the cost of direct reinforcement learning, it still suffers from challenges such as delayed rewards and policy degradation. This paper proposes an Emotion-Sensitive Deep Dyna-Q (ES-DDQ) model which: (1) presents an emotional world model that considers emotion-related cues to improve the ability of the traditional DDQ framework to model and simulate users, and (2) designs two kinds of emotion-related immediate rewards to mitigate the delayed reward problem. Experimental results show that our proposed approach effectively simulates users' behaviors and is superior to the state-of-the-art benchmarks.

1. Introduction

The task-oriented dialogue system is designed to help users solve specific task through human-computer interaction [4,9,7]. The recent advancements in deep learning also stimulate the enthusiasm of academia and industry to explore human-computer dialogue, and dialogue-based intelligent virtual assistants (such as Apple Siri, Microsoft Cortana, and Amazon Echo) have begun to facilitate human life. As one of the key components of task-oriented dialogue systems, dialogue policy, which determines how dialogue agents respond and take actions, have been extensively studied in recent years.

Dialogue policy optimization is often regarded as a reinforcement learning (RL) task. However, training a dialogue policy model through direct RL [8,36] requires a lot of conversation experience with real users, which may cause expensive training costs. Although user simulators can be applied as a cheaper alternative [10,9], the discrepancies between real users and simulators always exist [35]. Recently, methods based on Dyna-Q learning [27,16,26,40,38] have effectively combined the advantages of direct RL and simulated-based RL, and have shown advanced performance in task-oriented dialogue policy learning.

Although the emergence of DDQ has effectively alleviated the training cost problem in dialogue policy optimization, this method still faces challenges such as delayed reward and policy degradation. On the one hand, the traditional DDQ model only provides a clear reward to the dialogue agent at the end of the conversation, which leads to the delayed reward problem. On the other hand, the existing world model for dialogue policy learning is usually designed as a simple multi-layer perceptron (MLP), which is insufficient to model the complex user behaviors in real conversation. Furthermore, the poor performance of the world model will cause the policy model to degenerate in the later training stage, as pointed out by Peng [16]. Although some researchers [26,34] attempted to filter out the low-quality simulated experience through extra discriminators, these methods are undoubtedly not able to take full advantage of the world model.

An intuitive way to improve the above problem is to consider emotional information in the dialogue policy learning model. In fact, the user's emotions in task-oriented dialogues can usually reflect the performance of the policy model (as shown in Table 1). The research by Martinovski [13] proved that many breakdowns in human-computer dialogues are avoidable if the dialogue agent can recognize the user's emotional state and present more appropriate responses. Prendinger [19] took a different perspective on emotional dialogue, and showed that a dialogue agent with empathetic ability can indeed contribute to a more positive perception of the human-machine conversation. Broekens [2] also found that, com-

* Corresponding author.
E-mail addresses: wangzy@scut.edu.cn (Z. Wang), msyyz@mail.scut.edu.cn (Y. Zhao).

https://doi.org/10.1016/j.neucom.2021.06.075
0925-2312/

Table 1
Sample conversations with negative user emotions. These dialogues are from the Frame dataset.

User:	Hi! I'm looking to go from Essen to San Juan between August 27 to September 17.
User:	Traveling alone.
Agent:	How does Sept 1–7 sound?
User:	Its for an internship, dates are non-negotiable. If thats not possible, can you check Santo Domingo for me?
Agent:	Do you have a budget? Those dates don't work for Santo Domingo.
User:	Try Kyoto.
Agent:	August 30-Sept 4?
User:	August 27 to September 17. [Displeased]
Agent:	August 27-September 1?
User:	I feel like I've myself abundantly clear. [Angry]
Agent:	August 30-September 4?
User:	August 27 to September 17. (The user terminated the session.)
User:	Yo I need a flight to Beijing, leaving from Rio de Janeiro on Sunday August 21st
Agent:	I will book you 8 days in Manaus leaving from Rio de Janeiro on August 24th for 469.25USD.
Agent:	Enjoy your stay.
User:	No I want Beijing, not Manaus. How much would that cost??? [Displeased]
Agent:	Sadly I have no trips available to Beijing that leave from Rio de Janeiro.
Agent:	I can find you another destination.
User:	You gotta get me to Beijing. [Angry]
Agent:	No.
User:	Check Porto Alegre to Beijing. [Displeased]
Agent:	I can book you from Porto Alegre to Rio de Janeiro.
User:	Check Puebla to Beijing. [Displeased]
Agent:	You cannot travel from Puebla to Beijing.
User:	I give up. (The user terminated the session.)

pared to a non-emotional agent, the learning of an agent with human's emotional feedback can be sped up.

To this end, we propose an Emotion-Sensitive Deep Dyna-Q framework (ES-DDQ), which improves the existing DDQ-based dialogue policy learning methods using emotional information from the following two aspects: (1) We propose an emotional world model that utilizes rich contextual dialogue and emotional information to improve its imitation ability, and (2) We designed two emotion-related immediate rewards, the dynamic subgoal completion reward and the static emotional state reward, to encourage the policy model to take actions that benefit the user's emotions. Experiments on two conversation datasets show that the proposed ES-DDQ framework can better model the real world (environment) and improve the efficiency of model convergence. The major contributions of this paper are summarized as follows:

- We propose an Emotional-Sensitive Deep Dyna-Q (ES-DDQ) model for dialogue policy learning. To the best of the authors' knowledge, this is the first work to apply emotional features to dialogue policy learning problems on the base of the Deep Dyna-Q framework.
- We present an emotional world model with contextual modeling capabilities to better simulate user behaviors, and design two emotion-related immediate rewards to encourage the policy model to produce more appropriate actions.
- Extensive experiments are performed on two conversation datasets. The experimental results demonstrate the efficacy and superiority of our approach.

The rest of paper is arranged as follow. In Section 2 we review the background and related work. We present details of our propose method in Section 3. We outline the experiments and ablation analysis in Section 4. We close with our conclusions and a discussion of future work in Section 5.

2. Related work

2.1. Emotions in human-computer dialogue

The understanding and application of user emotions in human-machine dialogues have been studied for decades. The first psychomedical chatting robot ELIZA [31] was developed by MIT in 1966, in which emotion plays an important role. In the following half-century, human-computer dialogue technology has undergone several generations of development, and the emotions in conversation have gradually attracted more and more researchers' attention.

The first-generation emotional human-computer dialogue systems [20,1,24,18] mainly determined the system response mode based on artificially constructed rules to produce reasonable responses to different user emotional states. The advantage of these approaches is that the internal logic is transparent, which is easy to analyze and debug, but it highly depends on the manual intervention of experts and lacks flexibility and scalability.

With the development of big data technology, the second-generation dialogue systems based on statistical methods have emerged. The most representative one is the statistical dialogue system based on the Partially Observable Markov Decision Process (POMDP) [36]. Bui et al. [3] and Ren et al. [20,21] further proposed human-machine dialogue systems considering emotional characteristics, which combined dialogue state and emotional state by Factored POMDP [33].

In recent years, the emergence of neural networks has greatly promoted the dialogue policy learning methods based on deep reinforcement learning. Many researchers attempted to learn a dialogue agent in a data-driven manner [25,9,32,16]. However, most of these studies can only provide a clear reward to the dialogue agent at the end of the conversation, which leads to the delayed reward problem. Since it is difficult for the agent to obtain the immediate reward of the current action during the dialogue, this makes the model convergence slower. Although some immediate rewards such as expert-based reward [6], reward based on multi-modal emotion features [23] and interaction quality reward [29] have been designed, research on emotions in human-machine dialogue is still scarce.

In addition, research [18,37] also shows that a human-machine dialogue system with emotional intelligence can effectively improve the user experience. Emotional information helps human users interact more actively with the conversation system, thereby reducing low-quality communication [13,17]. Researchers also explored potential cues that can reflect the user's emotions in the conversation [12].

2.2. Deep dyna-Q learning

Task-oriented dialogue policies learning is often regarded as a reinforcement learning (RL) problem [8,36]. However, RL may incur huge costs in the real world, since it requires a large amount of interaction with real users. A common solution is to build user simulators based on real conversation data, and convert human-machine dialogue into a simulation problem [22,9]. In this way, dialogue agents can learn strategies by interacting with the user simulator without any actual costs. Then, policy models trained in this manner can be deployed online to interact with real users to further improve model performance. In recent years, most studies in this field have adopted this training strategy [25,39,32].

Unfortunately, user simulators are often unable to imitate the complex scenes in real human-machine dialogues, and the training of dialogue agents is inevitably affected by biases in the design of the user simulator. In order to solve this problem, Peng et al. [16]

proposed a Deep Dyna-Q (DDQ) learning approach based on the Dyna-Q framework [27], and adapts to the huge state space of task-oriented dialogue through the neural network. As illustrated in Fig. 1, the DDQ framework generally contains four components: Policy Model, World, World Model, and Evaluation Function. Among them, the Policy Model is a function formed by a set of reactions, which receives as input a description of the current state of the world, and produces as output an action to respond. The World can be regarded as an environment composed of real users and their goals, it receives actions yielded by Policy Model and produces the next state. The World Model is intended to mimic the one-step input-output behavior of the World, by interacting with the Policy Model through a process called planning to generate experiences that can be used for policy learning. The Evaluation Function is used to quickly map the state to reward value. The DDQ framework integrates the advantages of pre-planning and real-time learning by switching between the World and the World Model.

Nevertheless, the standard DDQ model is difficult to guarantee the quality of simulation experience on complex dialogue tasks. In the early stage of dialogue policy optimization, it is helpful to perform planning with simulated experiences regardless of their quality, but such low-quality simulated experiences are usually detrimental to the performance of the dialogue agent when the policy model has been significantly improved, as pointed out in [16]. Although recent studies such as D3Q [26] and Switch-based Active DDQ [34] attempted to filter out low-quality simulated experiences through extra discriminators, the performance of the world model has not been essentially improved and these methods cannot fully utilize the advantages of Dyna-Q learning.

3. Proposed method

This paper proposes to use emotional information to improve the delayed reward problem and the performance degradation problem in the DDQ-based model. Our method optimizes the policy learning process from two aspects: (1) We design emotion-based immediate rewards to improve the existing reward function; and (2) We introduce emotion-related cues as observation information in the world model, and apply the user emotion prediction as a joint task to improve the world model's ability to model the environment.

As illustrated in Fig. 2, the training of the ES-DDQ model includes four processes: (1) *Direct Reinforcement Learning*: where the agent interacts with real users directly, and improve the dialogue policy based on real experiences. (2) *Reward Calculating*: the evaluation function calculates immediate rewards based on the dialogue state and user's emotional state. (3) *World Model Learning*: the emotional world model updates its parameters using real experience. (4) *Planning*: where the agent interacts with the world model and improves the policy using simulated experiences.

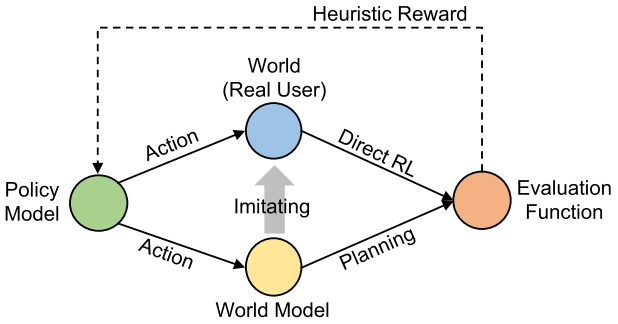


Fig. 1. Basic idea of the DDQ framework.

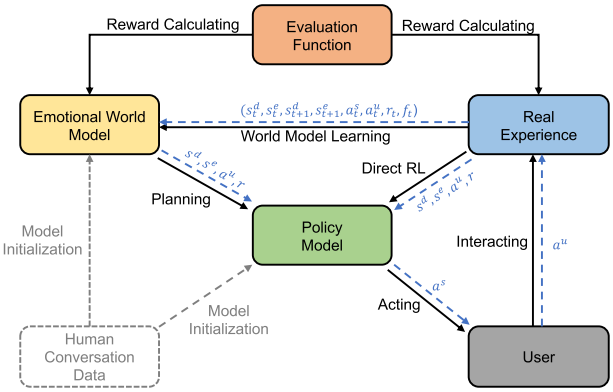


Fig. 2. Overall framework of our proposed ES-DDQ model. The black solid line indicates the dependency relationship between modules, and the blue dashed line indicates the data flow.

3.1. Direct reinforcement learning

Both dialogue and emotional states can be modeled as a Markov process [3,21,30], which is, the probability of each state depends only on the previous state. Thus, in this paper, we apply a DQN network [14] to learn the dialogue policy from real dialogue data.

At each turn t, the agent first executes a system action a_t^s according to the dialogue state s_t^d and user's emotional state s_t^e. Action a_t^s is chosen based on an ϵ-greedy policy which yields a random action with probability ϵ or according to the greedy policy $a_t^s = \arg\max_{a'} Q(s_t^d, s_t^e, a'; \theta_Q)$. Following [16] we utilize a Multi-Layer Perceptron (MLP) to approximate the value function $Q(s^d, s^e, a^s; \theta_Q)$. Afterwards, the agent receives a reward r_t, observes both the next user response a_t^u and user's emotional state s_{t+1}^e, then updates the dialogue state to s_{t+1}^d. Finally, we store the experience tuple $(s_t^d, s_t^e, a_t^s, r_t, a_t^u, s_{t+1}^d, s_{t+1}^e)$ into the replay buffer D^u. This cycle continues until the end of conversation.

The value function $Q(s^d, s^e, a^s; \theta_Q)$ is optimized by minimizing the mean-squared loss function, calculated as:

$$\mathscr{L}(\theta_Q) = \mathbb{E}_{(s_t^d, s_t^e, a_t^s, r_t, s_{t+1}^d, s_{t+1}^e) \sim D^u}\left[\left(y - Q(s_t^d, s_t^e, a_t^s; \theta_Q)\right)^2\right] \tag{1}$$

with y calculated as:

$$y = r + \gamma \max_{a'} Q'(s_t^d, s_t^e, a'; \theta_Q') \tag{2}$$

where $\gamma \in [0, 1]$ is a discount factor, and $Q'(\cdot)$ indicates the target value function which is only updated periodically. $Q(\cdot)$ can be optimized through $\nabla \theta_Q \mathscr{L}(\theta_Q)$ by back-propagation and mini-batch gradient descent.

3.2. Reward calculating

In order to solve the aforementioned delayed reward problem, and prevent the agent from producing repeated queries and irrelevant responses, we introduce a dynamic subgoal completion reward r_{subgoal} and a static emotion reward r_{emotion}. For each completed conversation, the agent will receive a positive reward of $2L$ for success or a negative reward of $-L$ for failure. In each turn, the agent receives a reward of $-1+r_{\text{subgoal}}+r_{\text{emotion}}$, where -1 and r_{subgoal} encourages shorter and more relevent conversations, respectively, and r_{emotion} rewards the agent based on user's emotional state. Details of r_{subgoal} and r_{emotion} are discussed as below.

3.2.1. Dynamic subgoal reward

A certain user goal can be divided into multiple sub-goals according to the user's constraints and requests, as suggested by [31,40]. When all subgoals are completed, it is equivalent to completing the corresponding user goal. Intuitively, by rewarding the completion of subgoals, the agent can be encouraged to produce task-related dialogue actions, and effectively improve the user experience.

Considering an user goal $G = (C, R)$ which contains a set of constraints C and a set of requests R, the goal G can be divided into multiple subgoals $G' = (C', R') \sqsubset G$, where $C' \subset C, R' \subset R$, and $G' \neq \varnothing$. Subgoals can be generated using a dynamic segmentation algorithm to avoid the combinatorial explosion problem. And then, the dynamic subgoals completion reward R_{subgoal} can be calculated as:

$$r_{\text{subgoal}} = \alpha_1 |C'| + \alpha_2 |R'|. \tag{3}$$

where C' and R' are the constraints and requests in a subgoal $G' = (C', R')$, with $|C'|$ and $|R'|$ represent the numbers of slots that have been identified, respectively. Details of the definition of subgoals and the dynamic segmentation algorithm are discussed in [40].

3.2.2. Static emotion reward

As can be seen in Table 1, repeated actions (e.g. queries) often result in negative user emotions. However, the user's goal may change during a task-oriented conversation, so it is necessary to cover some known dialogue slots under certain circumstances. Therefore, we may need to design very complex and difficult-to-understand logic to determine whether a repeated question is reasonable. Hence, we promote agents to avoid such unsuitable behaviors base on user's emotion, which is a more explicit signal. The proposed static emotion reward is calculated as $r_{\text{emotion}} = \alpha_3 R_e(\cdot)$, where $R_e(\cdot)$ is a mapping function:

$$R_e(s_e) = \begin{cases} -4 & \text{, if } s_e \text{ is “Very Bad”}, \\ -2 & \text{, if } s_e \text{ is “Bad”}, \\ -1 & \text{, if } s_e \text{ is “Neutral”}, \\ 1 & \text{, if } s_e \text{ is “Good”}, \\ 2 & \text{, if } s_e \text{ is “Very Good”}. \end{cases} \tag{4}$$

where s_e represents the user's emotional state. Since there is no unified emotion classification standard in the field of psychology research, we divide the user's emotion into five intensity levels {*Very Bad, Bad, Neutral, Good, Very Good*}. This simple assumption greatly reduces the complexity of our problem, and experimental results show that such an assumption does not reduce the model performance.

3.3. Emotional world model learning

To improve the world model's ability to imitate user behaviors, we propose an Emotional World Model (EWM), as illustrated in Fig. 3. As can be seen, our EWM consists of an RNN-based modeling network and an MLP-based inference network. The modeling network, which aims to extract relevant features from the emotion-related observations, is designed as a Gated Recurrent Unit (GRU). And the MLP-based inference network is used to jointly predict the user's action a_t^u, user's emotion s_t^e, reward r_t, and a symbol f_t indicating whether the dialogue is finished.

We first define four kinds of emotion-related cues that can be obtained from the dialogue:

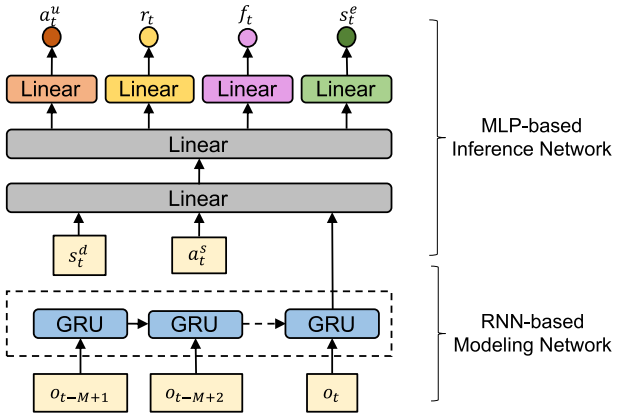


Fig. 3. The Emotional World Model architecture.

- **Task Completion Rate (TCR)**: which represents the progress of the user's goal. We obtain this value by calculating the proportion of completed subgoals.
- **Action Relevance (AR)**: which reflects how relevant the current system action a is to the user 's goal. We obtain this value by calculating the increment of TCR in this turn.
- **Subgoal/Turn Rate(STR)**: which indicates the average number of turns required by the dialogue system to complete a subgoal.
- **Last Emotion (LE)**: which is the emotional state s_e of user at last turn, as we model the emotional state as a Markov process.

At each time step t, the aforementioned emotion-related cues are regarded as observation o_t. Since the user's emotion change process is usually continuous, in this paper we apply the GRU to model the past M turns of observations into a vector o'_t, where M is the size of the context window. A GRU unit consists of the following components:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{5}$$
$$c_t = \sigma(W_c \cdot [h_{t-1}, x_t]) \tag{6}$$
$$\tilde{h}_t = \tanh(W \cdot [c_t * h_{t-1}, x_t]) \tag{7}$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{8}$$

where $\tilde{h}_t$ denotes the intermediate state which computes the candidate activation; h_t represents the activation of GRU at time t; c_t is a reset gate, which controls the effect of the previous activation h_{t-1} on the current candidate activation state $\tilde{h}_t$; and z_t is the update gate that controls the update process of the current activation based on the previous activation h_{t-1} and the candidate activation $\tilde{h}_t$.

Then the MLP-based inference network is implemented as a multi-task neural network which contains three classification objectives and a regression objective. The action inference network takes the current dialogue state s_t^d, the last system action a_t^s and the historical observation o'_t as input, and jointly generates user's action a_t^u, reward r_t, and the binary variable f_t, calculated as:

$$h^* = \tanh(W_h \cdot [s_t^d; a_t^s; o'_t] + b_h) \tag{9}$$
$$r_t = W_r \cdot h^* + b_r \tag{10}$$
$$a_t^u = \text{softmax}(W_a \cdot h^* + b_a) \tag{11}$$
$$f_t = \text{sigmoid}(W_t \cdot h^* + b_t) \tag{12}$$
$$s_t^e = \text{softmax}(W_e \cdot h^* + b_e) \tag{13}$$

where $[s_t^d; a_t^s; o'_t]$ is the concatenation of s_t^d, a_t^s and o'_t, and all W and b are weight matrices and bias terms, respectively.

3.4. Planning

The planning process is similar to direct reinforcement learning. Nevertheless, in this stage, the experience is generated through the interaction between the agent and the emotional world model. Therefore, we use another replay buffer D^s to store these experiences.

4. Experiment

4.1. Datasets

We conducted experiments on two public conversation datasets, including:

- The **Movie-ticket Booking** dataset provided by [9], which contains raw conversational data collected via Amazon Mechanical Turk. The dataset has been manually annotated according to a schema defined by domain experts, which consists of 11 intents and 16 slots, As illustrated in Table 4 in the Appendix. In total, the dataset contains 280 labeled conversations, and the average length of which is about 11 turns.
- The **Frame** dataset provided by [5], which consists of 1,369 goal-oriented human-human dialogues. The dataset contains 19,986 turns, with 20 intents (dialogue acts) and 43 unique slots, as shown in Appendix Table 5. The Frame dataset is marked with user emotion labels, which indicates whether the user is satisfied with the entire conversation.

In order to initialize the policy model and the emotional world model, for each dataset, we randomly selected 100 dialogues to perform manual emotion annotation.

4.2. Baselines

To evaluate the effectiveness of our proposed method, we compare the following approaches:

- The **DQN** agent that trained by the standard deep q-learning network, implemented with direct reinforcement learning in each epoch, as introduced in [9].
- The **DQN(K)** agent which is learned by DQN, but with K times more real experiences than the standard DQN agent. Assuming that the world model can effectively simulate the behavior of real users, the performance of DQN(K) can be regarded as the upper bound of DDQ(K),
- The **DDQ(K)** agent [16] that is trained using a pre-trained standard world model and a DQN network for direct reinforcement learning and planning, with $(K-1)$ planning steps. We set K to 5, 10, and 20 to compare the performance of the model under different planning steps.
- The **DR-D3Q** agent [40] which combines Dueling DQN network and dynamic subgoal rewards. The steps K of the DR-D3Q model is also set to 20.
- Our proposed **ES-DDQ** agent which utilizes a emotional world model with dynamic reward $R_{subgoal}$ and static reward $R_{emotion}$.

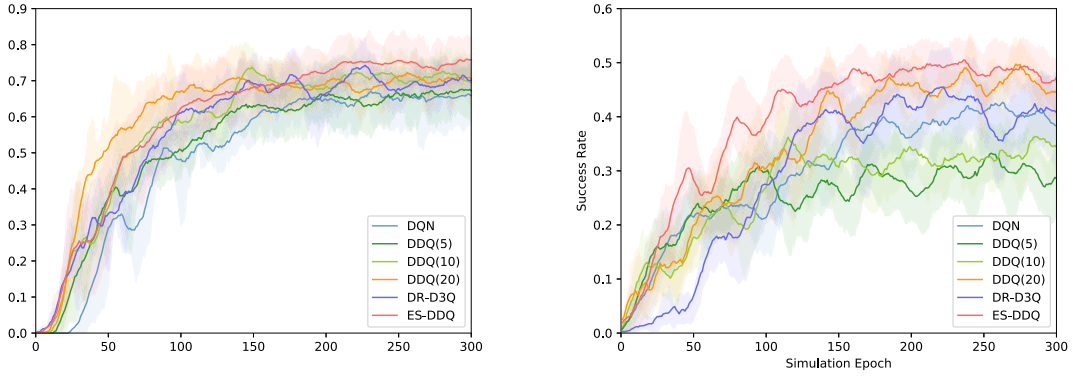
In addition, we adopt different model configurations for ablation studies.

- **ES-DDQ w/o EWM**. An ES-DDQ variant which uses a standard world model [16] without considering any emotional features.
- **ES-DDQ w/o** $r_{emotion}$. An ES-DDQ variant without static reward $r_{emotion}$, where the reward for unterminated conversation is $-1 + r_{subgoal}$ at each turn.
- **ES-DDQ w/o** $r_{subgoal}$. An ES-DDQ variant without dynamic reward $r_{subgoal}$, where the reward for a single unterminated turn is set to $-1 + r_{emotion}$.

4.3. Implementation details

For all the aforementioned approaches, the buffer size of B^u and B^s are both set to 10000. The batch size is set to 16, and the learning rate is $1e^{-3}$. And we set the discount factor $\gamma = 0.9$. For each method, we apply ϵ-greedy for exploration, with $\epsilon = 0.05$. The weight parameters α_1, α_2 and α_3 for proposed immediate rewards are set to 0.04, 0.04, 0.1, respectively. To prevent gradient explosion, we applied gradient clipping on all the model parameters with the maximum norm set to 1. The maximum length of a simulated dialogue is set to 40 turns ($L = 40$), which is, the dialogue is counted as failed if its length of turns exceeds L. In order to produce initial experiences that improve model training efficiency, we also utilize a variant of imitation learning, namely Reply Buffer Spiking (RBS) [11], by building a naive but occasionally successful rule-based agent according to real human conversation dataset at the beginning of training.

For DDQ(K) and DR-D3Q, each world model consists of 2 shared fully-connected layer and 3 task-specific hidden layers, with the size of each layer is set to 80. For ES-DDQ, the hidden size of GRU and that of the MLP-based inference network is also set to 80. The context window M is set to 10.



(a) The Movie-ticket Booking dataset. (b) The Frame dataset.

Fig. 4. Learning curves of DQN, DDQ(K), DR-D3Q and ES-DDQ on two datasets.

Table 2
Results of different agents on the Movie-ticket Booking dataset at training epoch={100, 200, 300}. Each number is averaged on 10 runs, each run tests on 128 dialogues. On epoch 200 and 300, ES-DDQ has a higher success rate than other benchmarks. Except for (epoch 100, ES-DDQ w/o $r_{subgoal}$), ES-DDQ is superior to its variants.

Agent	Epoch = 100			Epoch = 200			Epoch = 300		
	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns
DQN(20) *[upper bound]*	0.8246	53.46	12.98	0.8253	53.12	12.69	0.8248	53.25	13.46
DQN	0.4823	5.67	26.41	0.6483	27.07	23.46	0.6587	29.66	20.76
DDQ(5)	0.5081	8.70	26.52	0.6573	29.02	21.72	0.6727	31.71	20.02
DDQ(10)	0.5833	18.50	25.08	0.6847	33.21	19.90	0.6963	35.15	18.81
DDQ(20)	0.6717	30.67	21.86	0.6927	34.33	19.58	0.7063	36.16	19.20
DR-D3Q	0.6101	22.79	23.43	0.6656	30.23	21.82	0.7010	35.47	19.79
ES-DDQ (w/o EWM)	0.4427	−3.65	29.67	0.6673	26.92	23.45	0.7403	38.78	17.82
ES-DDQ (w/o $r_{emotion}$)	0.5608	16.33	23.93	0.6664	30.07	21.79	0.7327	38.69	20.47
ES-DDQ (w/o $r_{subgoal}$)	0.6837	32.16	22.07	0.7004	34.71	20.95	0.7284	38.53	20.02
ES-DDQ	0.6243	22.74	22.78	0.7232	35.76	20.78	0.7581	41.10	18.55

Table 3
Results of different agents on the Frame dataset at training epoch={100, 200, 300}. Each number is averaged on 10 runs, each run tests on 128 dialogues. On each epoch, ES-DDQ outperforms all the baseline methods.

Agent	Epoch = 100			Epoch = 200			Epoch = 300		
	Success	Reward	Turns	Success	Reward	Turns	Success	Reward	Turns
DQN(20) *[upper bound]*	0.5397	11.99	20.56	0.5561	16.63	19.74	0.5641	18.55	19.47
DQN	0.2341	−26.91	32.02	0.3785	−7.02	26.88	0.3971	−4.38	26.08
DDQ(5)	0.2922	−17.79	27.70	0.2689	−21.68	29.89	0.2894	−19.36	30.18
DDQ(10)	0.2741	−20.95	29.68	0.3341	−12.95	28.08	0.3546	−10.31	27.72
DDQ(20)	0.3179	−15.89	30.09	0.4603	5.29	21.92	0.4472	3.26	22.81
DR-D3Q	0.2657	−22.59	31.22	0.4282	0.72	23.63	0.4072	−2.41	24.82
ES-DDQ (w/o EWM)	0.2392	−26.97	27.73	0.4075	−3.60	22.46	0.4654	3.25	22.68
ES-DDQ (w/o $r_{emotion}$)	0.2847	−19.11	28.28	0.4377	2.11	22.60	0.4195	−0.67	23.77
ES-DDQ (w/o $r_{subgoal}$)	0.2782	−22.02	27.97	0.4648	3.33	22.59	0.4643	2.61	24.08
ES-DDQ	0.4001	−3.78	22.85	0.4886	8.59	19.58	0.4763	6.36	21.06

4.4. Simulation evaluation

The conversational agent is optimized by interacting with a user simulator instead of real users in this setting. Therefore, the world model is to imitate the behaviors of the user simulator. Although the difference between the simulator and real users might result in a sub-optimal agent [16], such a simulation setting allows us to analyze and reproduce the experimental results without much cost.

4.4.1. User simulator

We build our user simulator based on a publicly available framework [9] according to our task. During training, the user simulator provides the agent with a simulated responding action in each conversation turn and a reward signal at the end of the dialogue.

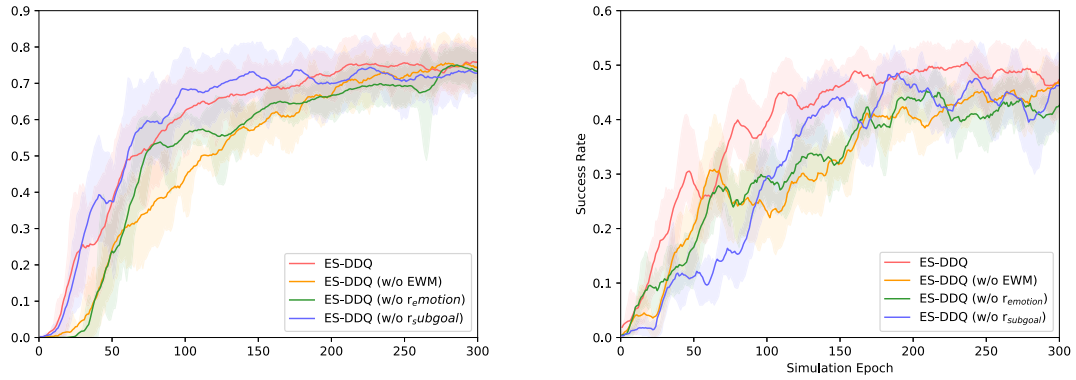
Unlike previous works [16,9,15], we design an additional "emotion" dimension in the user simulator inspired by [20,3]. When the dialogue agent produces some unexpected or unreasonable actions (such as asking the user for information that has already been informed or completely irrelevant to the current task), the user simulator will transfer its emotional state to a more negative state with a certain probability according to specific rules. Conversely, the emotion will shift to a positive state with a certain probability when the agent produces the desired action. When the user's emotion is in the "Bad" or "Very Bad" state, the simulator will randomly yield a "complaint" action with probability ρ. With this setting, we can mimic the complaining or aborting behavior in real human-machine conversation. Details of the user simulator are reported in Appendix A.

4.4.2. Simulation results

The experimental results performed on two datasets are reported in terms of success rate, average reward, and the average number of turns. Since the performance of the DDQ(K) agent is highly sensitive to parameter K and the best performance can be obtained when K is set to 20, we use K = 20 as a default setting in subsequent experiments.

This shows that our proposed ES-DDQ model can better use the emotional features as a cue to improve the policy model. thereby increasing user's willingness to participate in the dialogue and improving the success rate of conversation. Fig. 4(a) illustrates the success rates of the different methods change as the simulation epoch increases. It can be seen that the DDQ(20) agent achieves a leading position in the early stage and achieve a success rate of 0.5 at about 50 epochs. However, the performance of ES-DDQ surpasses that of DDQ(20) at about 180 epochs, and reached the optimal performance in subsequent simulations. We believe that this is because our proposed emotional world model contains a more complex recurrent network structure, which needs more training steps for finding the optimal parameters in the early stages of training. With the training steps increase, our model shows better performance. Meanwhile, the DR-D3Q agent which utilizes the dueling network for decision-making does not show superiority under this simulation scenario.

For the Frame dataset, ES-DDQ also achieves better performance than the benchmark models, as shown in Table 3. Since the Frame dataset contains more slots and dialogue actions, the upper bound (DQN(20)) is much lower. In this dataset, our model shows a stronger learning ability compared to the baseline. In each epoch, our ES-DDQ method increases the average success rate by 0.0822, 0.0283, and 0.0291, which shows that our model converges faster than the baseline models and achieves better performance.



(a) The Movie-ticket Booking dataset. (b) The Frame dataset.

Fig. 5. Learning curves of ES-DDQ and different control models on two datasets.

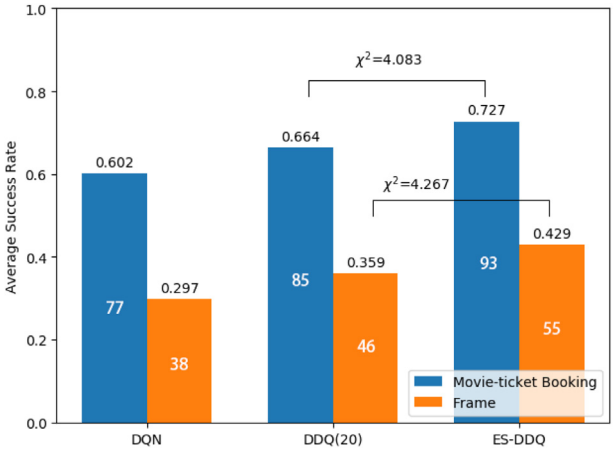


Fig. 6. The human evaluation results of DQN, DDQ(20) and ES-DDQ ($\chi^2_{0.05} = 3.841$).

4.4.3. Ablation studies

In order to further evaluate our proposed method, we conduct an ablation experiment and verify the impact of certain modules on model performance by removing them. Table 2 and 3 indicates the experimental results of different variant models. The results of the control model (ES-DDQ w/o EWM) show that, compared with standard world models, the EWM that considers emotional information can more effectively understand and simulate the environment, which is conducive to learning better dialogue policy. Moreover, the ablation experiments on r_{emotion} and r_{subgoal} show that these two reward mechanisms can improve the performance of agents to a certain extent, and make the model achieve the best performance when working collaboratively. Fig. 5 illustrates the training curves of EWM and its control models on two datasets.

4.5. Human evaluation

We recruit real users to evaluate model performance by interacting with different dialogue agents. At the beginning of each conversation, the user randomly selects a user goal from the dialogue corpus. Unlike [40], the user is required to record his/her own emotional state at each turn, and is allowed to generate complaints or terminate the conversation based on his current emotional state. In our experiments, conversations terminated early will be considered as failures.

As shown in Fig. 6, the success rate of all models in human evaluation is lower than the results of simulation experiments, which indicates the sub-optimality of simulation-based methods. We use McNemar's test to verify the significance of the performance difference between the ES-DDQ model and the DDQ model. The ES-DDQ is superior to other agents since ES-DDQ produces fewer repeated and irrelevant queries compared to the baseline models, which makes users more willing to collaborate.

5. Conclusion

In this paper, we propose an Emotion-Sensitive Deep Dyna-Q model for dialogue policy learning. The main contributions of our method include: (1) We propose an emotional world model to better model and imitate the behaviors of users in human-computer conversation scenario. (2) By regarding the user's emotional features as immediate rewards, we propose two kinds of emotion-related rewards, which effectively improves the problem of delayed reward for dialogue policy learning.

In the future, we will further consider the personalized difference of users to establish a dialogue policy model that can adapt to different types of users. In addition, we plan to combine our method with existing generative adversarial learning approaches, to further improve the performance of the dialogue policy model.

CRediT authorship contribution statement

Rui Zhang: Methodology, Writing - original draft. **Zhenyu Wang:** Writing - review & editing. **Mengdan Zheng:** Validation. **Yangyang Zhao:** Conceptualization, Writing - original draft. **Zhenhua Huang:** Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the editors and anonymous reviewers for their insightful feedback on this work. This paper is supported by the Natural Science Foundation of Guangdong (No.2019A1515011792), the Key-Area Research and Development Program of Guangdong Province (Grant No.2019B010154004), and the Industry-Academic Research Fund Project of Guangzhou (No.2019PT103).

Table 4
Annotation schema for the Movie-ticket Booking dataset.

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

Table 5
Annotation schema for the Frame dataset.

	Annotation
Intent	inform, offer, request, switch_frame, suggest, no_result, thankyou, sorry, greeting, affirm, negate, confirm, moreinfo, goodbye, heremore, request_alts, request_compare, canthelp, you_are_welcome, reject
Slot	price, duration, name, country, category, city, guest_rating, breakfast, parking, wifi, gym, spa, park, museum, beach, shopping, market, airport, university, mall, cathedral, downtown, palace, theatre, seat, duration_dep, departure_date_dep, departure_date_arr, departure_time_dep, arrival_time_dep, arrival_time_arr, duration_arr, price_max, price_min, destination_city, max_duration, num_adults, num_children, start_date, end_date, are_dates_flexible, origin_city

Appendix A. User simulator

In a task-oriented conversation, the objective of a dialogue agent is to help the user accomplish a certain goal, although the agent knows nothing about this goal. Generally speaking, the user's goal consists of two parts:

- inform_slots: contain a number of slot-value pairs which can be regarded as *constraints* for the user.
- request_slots contain a set of slots that user has no information about the values, but wants to get the value from the agent during the conversation.

In addition, user goals are divided into two groups: required slots and optional slots. For the two experimental datasets, we generate user goals using different mechanisms. For the Movie-ticket Booking dataset, we extract all the slots (known and unknown) from the first user turns in the data, since most existing dialogue policy learning approaches apply this extraction method on this dataset. For the Frame dataset, we extract all the slots from all user turns and record the order in which they appear. This is used to simulate the scenarios where user goals change during the conversation. These user goals are dumped into files as user goal database. Every time when running a dialogue, we randomly sample on user goal from this user goal database.

Furthermore, we adopt a finite state machine similar to [20] to simulate user emotions which are divided into five categories: {*Very Bad, Bad, Neutral, Good, Very Good*}. At each time step t, user's emotional state transitions from the previous state s_e to the current state s'_e, based on some handcrafted rules:

- When system action a^s is to inquiry a slot that has already been informed by the user or inform/inquiry information that is irrelevant to user's goal, the emotional state will shift to a more negative state (for example, from *Neutral* to *Bad*) with probability ρ, and maintain the current state with a probability of 1-ρ.
- When system action a^s is to inquiry/inform information that is relevant to user's goal, the emotional state will shift to a more positive state (for example, from *Neutral* to *Good*) with probability ρ, and maintain the current state with a probability of 1-ρ.

The probability ρ is calculated as $\rho = 0.2 - 0.01 * n_s$, where $n_s \in [0, 10]$ represents the number of turns where the user simulator continues to be in the current emotional state. Furthermore, an additional user action "complaint" is adopted in our user simulator: When the user's emotional state is *Bad*, the simulator will generate a complaint action with a probability of 0.1, and when the emotional state is *Very Bad*, the probability of this action is 0.5. The introduction of the "complaint" action will only cause a longer conversation without affecting the information of user's goal. We apply this setting to simulate the complaints and suspension behavior of real users during human-computer conversations.

Appendix B. Data annotation schema

Table 4 and 5 list all annotated dialogue acts and slots in details.

References

[1] E. Andre, M. Rehm, W. Minker, D. Bühler, Endowing spoken language dialogue systems with emotional intelligence, Tutorial and Research Workshop on Affective Dialogue Systems, Springer. (2004) 178–187.
[2] J. Broekens, Emotion and reinforcement: affective facial expressions facilitate robot learning, Artifical Intelligence for Human Computing. Springer (2007) 113–132.
[3] T.H. Bui, J. Zwiers, M. Poel, A. Nijholt, Affective dialogue management using factored pomdps, Interactive Collaborative Information Systems. Springer (2010) 207–236.
[4] B. Dhingra, L. Li, X. Li, J. Gao, Y.N. Chen, F. Ahmad, L. Deng, 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 (Volume 1: Long Papers), 2017, pp. 484–495.
[5] L. El Asri, H. Schulz, S.K. Sarma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, K. Suleman, Frames: a corpus for adding memory to goal-oriented dialogue systems, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 207–219.
[6] E. Ferreira, F. Lefèvre, Expert-based reward shaping and exploration scheme for boosting policy learning of dialogue management, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE, 2013, pp. 108–113.
[7] J. Gao, M. Galley, L. Li, Neural approaches to conversational ai, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, 2018, pp. 2–7.
[8] E. Levin, R. Pieraccini, W. Eckert, Learning dialogue strategies within the markov decision process framework, in: 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings, IEEE, 1997, pp. 72–79..
[9] X. Li, Y.N. Chen, L. Li, J. Gao, A. Celikyilmaz, End-to-end task-completion neural dialogue systems, in: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2017, pp. 733–743.
[10] X. Li, Z.C. Lipton, B. Dhingra, L. Li, J. Gao, Y.N. Chen, A user simulator for task-completion dialogues, 2016. arXiv preprint arXiv:1612.05688..
[11] Z. Lipton, X. Li, J. Gao, L. Li, F. Ahmed, L. Deng, Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018..
[12] J. Liscombe, G. Riccardi, D. Hakkani-Tür, Using context to improve emotion detection in spoken dialog systems, in: Ninth European Conference on Speech Communication and Technology, 2005..
[13] B. Martinovski, D. Traum, Breakdown in human-machine interaction: the error is the clue, in: Proceedings of the ISCA tutorial and research workshop on Error handling in dialogue systems, 2003, pp. 11–16..
[14] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.
[15] B. Peng, X. Li, J. Gao, J. Liu, Y.N. Chen, K.F. Wong, Adversarial advantage actor-critic model for task-completion dialogue policy learning, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 6149–6153.
[16] B. Peng, X. Li, J. Gao, J. Liu, K.F. Wong, 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), pp. 2182–2192.
[17] T.S. Polzin, A. Waibel, Emotion-sensitive human-computer interfaces, in: ISCA tutorial and research workshop (ITRW) on speech and emotion, 2000..
[18] H. Prendinger, M. Ishizuka, The empathic companion: a character-based interface that addresses users'affective states, Appl. Artif. Intell. 19 (2005) 267–285.
[19] H. Prendinger, S. Mayer, J. Mori, M. Ishizuka, Using bio-signals to measure and reflect the impact of character-based interfaces, in: Proceedings Fourth International Working Conference on Intelligent Virtual Agents, 2003.
[20] F. Ren, Y. Wang, C. Quan, Tfsm-based dialogue management model framework for affective dialogue systems, IEEJ Trans. Electr. Electron. Eng. 10 (2015) 404–410.
[21] F. Ren, Y. Wang, C. Quan, A novel factored pomdp model for affective dialogue management, J. Intell. Fuzzy Syst. 31 (2016) 127–136.
[22] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, S. Young, Agenda-based user simulation for bootstrapping a pomdp dialogue system, in: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, Association for Computational Linguistics, 2007, pp. 149–152.
[23] W. Shi, Z. Yu, Sentiment adaptive end-to-end dialog systems, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1509–1519..
[24] M. Skowron, Affect listeners: Acquisition of affective states by means of conversational systems, in: Development of Multimodal Interfaces: Active Listening and Synchrony, Springer, 2010, pp. 169–181..
[25] P.H. Su, M. Gasic, N. Mrksic, L. Rojas-Barahona, S. Ultes, D. Vandyke, T.H. Wen, S. Young, Continuously learning neural dialogue management, 2016. arXiv preprint arXiv:1606.02689..
[26] S.Y. Su, X. Li, J. Gao, J. Liu, Y.N. Chen, Discriminative deep dyna-q: Robust planning for dialogue policy learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3813–3823.
[27] R.S. Sutton, Integrated architectures for learning, planning, and reacting based on approximating dynamic programming, in: Machine learning proceedings 1990, Elsevier, 1990, pp. 216–224.
[28] D. Tang, X. Li, J. Gao, C. Wang, L. Li, T. Jebara, Subgoal discovery for hierarchical dialogue policy learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2298–2309.
[29] S. Ultes, P. Budzianowski, I. Casanueva, N. Mrkšić, L. Rojas-Barahona, P. Su, T. Wen, M. Gašić, S. Young, Domain-independent user satisfaction reward estimation for dialogue policy learning, in: Proceedings of the Annual Conference of the International Speech Communication Association INTERSPEECH, 2017, pp. 1721–1725.
[30] Y. Wang, F. Ren, C. Quan, A new factored pomdp model framework for affective tutoring systems, IEEJ Trans. Electr. Electron. Eng. 13 (2018) 1603–1611.
[31] J. Weizenbaum, Eliza–a computer program for the study of natural language communication between man and machine, Commun. ACM 9 (1966) 36–45.
[32] J.D. Williams, K.A. Atui, G. Zweig, 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 (Volume 1: Long Papers), 2017, pp. 665–677.
[33] J.D. Williams, P. Poupart, S. Young, Factored partially observable markov decision processes for dialogue management, in: Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, 2005, pp. 76–82.
[34] Y. Wu, X. Li, J. Liu, J. Gao, Y. Yang, Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 7289–7296..
[35] S. Young, C. Breslin, M. Gašić, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, E.T. Hancock, Evaluation of statistical pomdp-based dialogue systems in noisy environments, in: Situated Dialog in Speech-Based Human-Computer Interaction, Springer, 2016, pp. 3–14.
[36] S. Young, M. Gašić, B. Thomson, J.D. Williams, Pomdp-based statistical spoken dialog systems: a review, Proc. IEEE 101 (2013) 1160–1179.
[37] Z. Yu, A. Papangelis, A. Rudnicky, Ticktock: a non-goal-oriented multimodal dialog system with engagement awareness, in: 2015 AAAI Spring symposium series, 2015.
[38] Z. Zhang, X. Li, J. Gao, E. Chen, Budgeted policy learning for task-oriented dialogue systems, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3742–3751.
[39] T. Zhao, M. Eskenazi, Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning, in: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, pp. 1–10.
[40] Y. Zhao, Z. Wang, K. Yin, R. Zhang, Z. Huang, P. Wang, Dynamic reward-based dueling deep dyna-q: Robust policy learning in noisy environments, in: Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020..

Rui Zhang received the B.S. degree in School of Software Engineering from South China University of Technology, Guangzhou, China in 2015. He has been a Ph.D. candidate in School of Software Engineering at South China University of Technology since 2015. His research interests include natural language generation, text mining and sentiment analysis.

Zhenyu Wang received the Ph.D. degree from department of computer science, Harbin Institute of Technology in 1993. He is the dean of the School of Software, South China University of Technology, and the director of the Guangdong Provincial Social Media Processing and Engineering Center. His research interests include natural language processing, text mining, and social network analysis.

Mengdan Zheng received the B.S. degree from the School of Software Engineering, South China University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the master degree. His research interest includes natural language generation.

Yangyang Zhao born in 1995, she is currently a Ph.D. candidate in the School of Software Engineering at South China University of Technology, Guangzhou. Her main research interests include dialogue systems and deep reinforcement learning.

Zhenhua Huang received the B.S. degree in School of Software Engineering from South China University of Technology in 2014. He is currently a Ph.D candidate in the School of Software Engineering at South China University of Technology, Guangzhou. His research interests include social computing, sentiment analysis, and deep learning.