APSIPA Transactions on Signal and Information Processing, 2023, 12, e37

Overview Paper

Advances and Challenges in Multi-Domain Task-Oriented Dialogue Policy Optimization

Mahdin Rohmatillah[1] and Jen-Tzung Chien[2*]

[1]*EECS International Graduate Program, National Yang Ming Chiao Tung University, Taiwan*

[2]*Institute of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Taiwan*

ABSTRACT

Developing a successful dialogue policy for a multi-domain task-oriented dialogue (MDTD) system is a challenging task. Basically, a desirable dialogue policy acts as the decision-making agent who understands the user's intention to provide suitable responses within a short conversation. Furthermore, offering the precise answers to satisfy the user requirements makes the task even more challenging. This paper surveys recent advances in multi-domain task-oriented dialogue policy optimization and summarizes a number of solutions to policy learning. In particular, the case study on the task of travel assistance using the MDTD dataset based on MultiWOZ containing seven different domains is investigated. The dialogue policy optimization methods, categorized into dialogue act level and word level, are systematically presented. Moreover, this paper addresses a number of challenges and difficulties including the user simulator design and the dialogue policy evaluation which need to be resolved to further enhance the robustness and effectiveness in multi-domain dialogue policy representation.

*Corresponding author: Jen-Tzung Chien, jtchien@nycu.edu.tw.

Received 22 May 2023; Revised 20 July 2023
ISSN 2048-7703; DOI 10.1561/116.00000132

Keywords: Multi-domain task-oriented dialogue system, dialogue policy optimization, reinforcement learning, imitation learning, dialogue act prediction, word-level policy learning.

1 Introduction

Task-oriented dialogue system is a type of conversational system designed to fulfill user goals within specific domains in a limited number of conversation turns. In contrast to open-domain dialogue system, which functions as a chit-chat style via a chatbot without any pre-defined domains, task-oriented dialogue system operates within the pre-defined domains in which specific goals are engaged in a step-by-step conversation. In recent years, there has been significant progress in the development of task-oriented dialogue systems. From only handling one domain like a flight booking, the dialogue system has evolved to encompass multi-domain capability, allowing the system to handle various domains such as the bookings of hotel, restaurant and taxi. In order to satisfy the user goals for various domains, designing a multi-domain dialogue policy plays a vital role.

Basically, dialogue policy works as a decision-making component which is trained to determine the system response given the user input. In every conversation turn, dialogue policy should generate an appropriate response that aligns with the user input. Once the dialogue policy generates a wrong response, the conversation may potentially be disrupted. Sometimes, the problem is even more complicated in the case of multi-domain dialogue tasks where a single dialogue may contain a huge size of possible combinations of user intentions from different domains. For example, in a multi-domain dialogue scenario, a user may request an Italian restaurant reservation in the initial conversation turn, then ask about a 5-star hotel located nearby the Italian restaurant. In another case, a user may ask about a train schedule at the beginning, then ask about the ticket price of a specific attraction.

Due to the complexity in dialogue turns under multiple domains, while the recent well-known large language models (LLMs) have shown promising performance in the open-domain dialogue like chit-chat, LLMs still could not optimally cope with the multi-domain task-oriented dialogue if the models are simply trained according to a standard autoregressive objective, i.e. only predicting the next word. Different from the open-domain dialogue, task-oriented dialogue system aims to satisfy user specific goals within a small number of conversation turns. The answer for each user query is much more limited when compared with the open-domain dialogue task in which the dialogue policy may explore different responses to enhance user engagement. As a consequence, a delicate dialogue policy is required to handle the complicated dialogue flow in a task-oriented dialogue. In general, there are two levels of approaches to the task-oriented dialogue policy learning. First is the dialogue act (DA)-level [33, 81, 96] and the second is the word-level dialogue policy [16, 61, 102, 116].

In general, DA-level dialogue policy outputs the system dialogue act (e.g. 'Hotel-Inform':['area', 'centre']) in every conversation turn. The output of DA-level dialogue policy is then transformed by the natural language generation (NLG) component [62] to be a readable sentence. DA-level policy learning is commonly optimized by using reinforcement learning (RL) methods by leveraging the trajectories which are stored in the replay buffer in a dialogue task. The trajectories are defined as the conversation history between a dialogue agent and a simulated user with the corresponding reward in every turn. A simulated user can be built by using an agenda-based policy [84, 87] or a neural network model which is trained by using the provided dataset [75]. The replay buffer will be updated by following the first-in-first-out strategy. Another approach is to employ the offline RL which is an optimization method that totally relies on the dataset. There are two directions of offline RL. The first one is to estimate the dialogue policy by only using the dataset. The second one is to estimate the reward function in an inverse RL optimization, since the reward function in dialogue policy optimization is prone to be sparse.

Meanwhile, word-level dialogue policy is another kind of dialogue policy that conducts a sequence of actions by selecting a string of words as a readable sentence. Essentially, the word-level dialogue policy combines the components of dialogue policy and natural language generation. There are two distinct learning approaches within this dialogue policy framework. The first approach is to train an encoder-decoder model. The encoder aims to extract meaningful features from user input which are then concatenated with the belief state and information from the database to serve as the input for the decoder. The encoder can be built by using either recurrent neural network [83] or transformer-based model like BERT [24]. The second approach is to use the transformer-based decoder such as the GPT-2 [76] model. In this approach, the transformer-based decoder is actually optimized to represent all of the dialogue system components ranging from natural language understanding (NLU) to NLG. However, to some extent, this approach can be considered as the word-level dialogue policy if the belief state is estimated by another model or the ground-truth belief state is given during evaluation.[1]

[1] As shown on the MultiWOZ benchmark [4], the dialogue policy evaluation involves the transformer-based decoder conditioned on the ground-truth belief state.

Unfortunately, optimizing the dialogue policy performance is very challenging [15]. Either DA-level or word-level policy has its advantages and disadvantages. In DA-level policy, some recent methods [34, 81, 96] have shown a promising performance in the end-to-end system evaluation for multi-domain task-oriented dialogue. However, the sentences generated from the system using DA-level policy sometimes are not as natural as expected since such a policy relies on a template-based NLG to transform the dialogue acts into the corresponding sentences. Furthermore, it is challenging to design a proper simulated user that can mimic humans in the real implementation and exploit a reward function that can generate meaningful feedback in each conversation turn during RL optimization. On the other hand, the word-level policy, which is able to generate diverse responses [13] due to the utilization of LLMs such as the generative pre-trained transformer (GPT) [76], suffers from the issue of computational cost. Furthermore, several complicated pre-processing and post-processing stages must be carried out to pave an avenue to implement a task-oriented dialogue system based on LLMs [67].

Based on the aforementioned analysis, this paper surveys the recent advances in the multi-domain task-oriented dialogue (MDTD) policy optimization covering both DA-level policy and word-level policy. In contrast to the previous surveys on task-oriented dialogue management [23, 53] which either provide an overview of dialogue policy learning by RL methods or survey on the approaches to shortcomings of dialogue management models, this survey specifically focuses on the recent advances in dialogue policy learning within the context of MultiWOZ dataset [4] where both DA-level and word-level policy learning methods are systematically included in the evaluation. The MultiWOZ dataset is considered as the most challenging dataset among various datasets for MDTD task. Moreover, this survey encompasses a wide range of methodologies beyond RL-based approaches for dialogue policy. The surveyed methodologies in this paper are presented chronologically based on their publication year and categorized according to their optimization approaches. Additionally, this survey highlights the open problems and challenges that should be addressed in future research. These highlights include designing the simulated users, enhancing the robustness of multi-domain dialogue policies, and establishing the standardized evaluation settings to facilitate fair performance comparison across different models.

This paper is structured as follows. In Section 2, an overview of the multi-domain task-oriented dialogue system is addressed, including the problem definition and the optimization approaches. Sections 3 and 4 describes different policy learning strategies for multi-domain task-oriented dialogue which are categorized in DA-level policy and word-level policy, respectively. Next, a number of main challenges and difficulties are pointed out as mentioned in

Section 5. At last, Section 6 addresses the conclusions and future works drawn from this study.

2 Multi-Domain Task-Oriented Dialogue System

Different from open-domain dialogue system that focuses on sentence generation task, multi-domain task-oriented dialogue system is designed to understand and satisfy user goals across different domains, for example find the nearest restaurant from the hotel that has been recommended in the previous conversation turn. In order to achieve such user goals, the conversation between user and system should be driven in a more systematic way which is more like step-by-step conversation until achieving the final goal instead of free-flowing conversation like chit-chat. In the MDTD task, the dialogue system is not required to have broad knowledge capability. Instead, MDTD system is expected to excel in specific domains. The strength of MDTD system lies in its ability to handle those particular domains which have been included during training. Table 1 provides a summary of the differences between open-domain dialogue system and MDTD system.

Table 1: Summary of the differences between open-domain dialogue (ODD) system and multi-domain task-oriented (MDTD) system over different metrics.

	ODD	**MDTD**
domain	no specific domain	well-defined multiple domains
capability	broad knowledge but lack of domain-specific capability	capable only on the specific domains
dialogue flow	free-flowing dialogue	goal-driven dialogue
task	language generation	goal accomplishment

2.1 Performance Evaluation Tasks

In recent years, MDTD tasks have gained high attention due to the development of the datasets which sufficiently reflect the real-world scenarios. Among those publicly available datasets, MultiWOZ is the most popular dataset for MDTD task. MultiWOZ is specifically designed as the travel assistant task offering rich features. It comprises 7 different domains, 13 intents, 25 slot types, 10,483 dialog sessions, and a total of 71,544 conversation turns. This extensive dataset results in a large search space, encompassing all possible combinations of user intentions from different domains. The data collection process for MultiWOZ involved human-to-human interactions, ensuring a realistic dialogue setting. Although MultiWOZ is not the largest dataset in terms of data size, the popularity of MultiWOZ is assured due to its comprehension in label

Table 2: Summary of the differences among different MDTD datasets. H2H and M2M refer to human-to-human and machine-to-machine, respectively. English (En) and Chinese (Zh) are included. Different virtual assistants and labels are considered.

Name	Task-Language	Method	Size	Label Information
MultiWOZ [4]	Specific to Travel assistant-En	H2H	- 7 domains - 13.68 turns/dialogue - 13.18 tokens/turn	- Dialogue states - System dialogue acts - User dialogue acts (ConvLab) - Database - User goals
Taskmaster [5]	General Virtual assistant-En	H2H	- 6 domains - 22.9 turns/dialogue - 8.1 tokens/turn	- API calls and arguments
SGD [78]	General Virtual assistant-En	M2M	- 20 domains - 20.44 turns/dialogue - 9.75 tokens/turn	- Schema-guided dialog states - User dialogue acts - System dialogue acts - Services
CoSQL [111]	Specific to database retrieval assistant-En	H2H	- 138 domains - 22.9 turns/dialogue - 8.1 tokens/turn	- SQL queries - User dialogue acts - Database - Query goals
CrossWOZ [117]	Specific to Travel assistant-Zh	H2H	- 5 domains - 16.9 turns/dialogue - 16.3 tokens/turn	- Dialogue states - System dialogue acts - User dialogue acts (ConvLab) - Database - User goals

information. This dataset offers the detailed domain-specific labels, intents, and slot types, which are crucial to train a model for MDTD task. The availability of such label information allows the model to be specialized to become an expert in specific domains. Table 2 shows a comparison over different MDTD datasets in terms of task, language, method, size and label information.

In general, a multi-domain task-oriented dialogue system is built with different components ranging from natural language understanding, dialogue state tracking (DST), dialogue policy and natural language generation. Figure 1

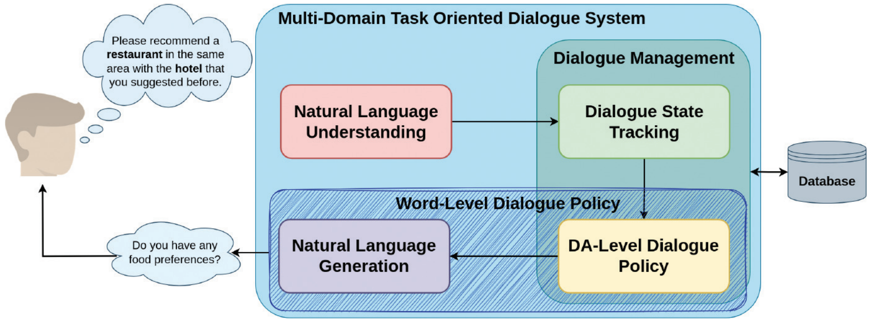

Please recommend a **restaurant** in the same area with the **hotel** that you suggested before.
Multi-Domain Task Oriented Dialogue System
Dialogue Management
Natural Language Understanding
Dialogue State Tracking
Database
Word-Level Dialogue Policy
Natural Language Generation
DA-Level Dialogue Policy
Do you have any food preferences?

Figure 1: An overview of multi-domain task-oriented dialogue system. A user may mention more than one domain in the conversation.

shows an illustration of MDTD system consisting of these components. This general setting is seen as a pipeline dialogue system in which each component can be optimized individually or jointly by involving two different components like dialogue policy and NLG. In this pipeline setting, each dialogue component has its own functionality. The NLU aims to predict current user dialogue acts conditioned on the conversation history or context. The model called bidirectional encoder representations from transformer (also known as the BERT) [24] is mostly used to act as the NLU component. Next, DST is a component which determines the user's intention in every conversation turn. The state of the conversation is expressed as a collection of pairs that associate a slot with a value. Typically, DST is optimized by using the classification objective where the model is trained to predict the slot-value pairs which are given in a pre-defined ontology. BERT-based models have shown convincing performance in the DST optimization [51, 112]. Meanwhile, the dialogue policy serves as the brain of the system which determines the system action given the user utterances. Dialogue policy basically produces the system dialogue act (DA) that is subsequently transformed as a readable sentence by the NLG component. Such a setting is designed to implement the DA-level policy which is commonly optimized by using RL algorithms [29, 49, 68, 86]. For the NLG component, the current popular methods are implemented as a kind of template-based NLG where the system dialogue act was expressed by a natural sentence through the pre-defined rules. Another approaches are the semantically conditioned long short-term memory (LSTM) [60] and GPT [73] which were exploited for sequence labeling and sentence generation, respectively.

2.2 Performance Comparison with ChatGPT

Instead of training dialogue policy and NLG components independently, alternative approaches have been developed to focus on training them jointly. This kind of dialogue policy is called the word-level dialogue policy. In this case, the dialogue policy directly generates a coherent sentence, making the action space of the policy correspond to the vocabulary size rather than a set of system dialogue acts. Autoregressive optimization is the most common method to train this word-level dialogue policy. At the beginning, most of the approaches were built through the recurrent neural networks [83]. However, due to the emerging transformer model [100] that has shown remarkable performance in various natural language processing (NLP) tasks, the research direction to train word-level dialogue policy is shifted to utilize either transformer encoder such as BERT [24] or transformer decoder such as GPT-2 [76]. Even though most GPT-2-based solutions were intended to represent all components in the task-oriented dialogue system by a single model, these approaches were still categorized as a kind of word-level dialogue policy. For example, in the

Table 3: Performance comparison between traditional learning methods and prompt learning methods by using ChatGPT or GPT-3.5 in the MultiWOZ dataset.

Model	Inform	Success	BLEU	Combined
Traditional Learning				
DAMD [115]	57.90	47.60	16.40	84.80
AuGPT [52]	76.60	60.50	16.80	85.40
SOLOIST [71]	81.70	67.10	13.60	88.00
UBAR [109]	83.40	70.30	17.60	94.40
GALAXY [39]	85.40	75.70	19.64	100.20
Prompt Learning				
ChatGPT	66.70	54.70	6.96	66.66
GPT-3.5 [70]	82.00	72.50	9.22	86.47

MultiWOZ evaluation,[2] when the model is given by the ground-truth belief state, this model is learned to act as a dialogue policy.

Amid the hype surroundings of ChatGPT or GPT-3.5 [70] due to their breakthrough performance in solving various NLP tasks by only using *prompt tuning or learning* [7, 63, 108], several works have sought to assess their capability in the MultiWOZ dataset. The first study was done in [2] which evaluated the quality of the response generation using ChatGPT in two different ways. The first approach involved the few-shot prompts with the access to the oracle system dialogue acts, while the second approach utilized the in-context learning. Subsequent work [114] aimed to enhance the performance of ChatGPT by designing multiple prompt functions such as DST and policy prompter. However, as shown in Table 3, the performance of using both ChatGPT and GPT-3.5 is worse than that of using traditional learning methods. The performance of each method was evaluated in three different metrics and a combined metric. Inform score measures whether the system provides an appropriate entity, success rate measures whether the system answers all the requested attributes, and BLEU score measures the quality of the generation of responses in accordance to the ground-truth responses. Two main reasons for suboptimal performance using ChatGPT and GPT-3.5 are the so-called hallucination and reasoning problems which cause the unrelated sentence generation. These problems reflect the fact that the challenges in the MDTD task are considerable and the unique solutions to these challenges are required.

Given the characteristics and challenges of the MDTD task, this paper provides a comprehensive survey of recent approaches in task-oriented dialogue policy optimization, specifically in the context of MultiWOZ dataset. The focus of this paper is on the learning of task-oriented dialogue policies, since it serves as the decision maker of the system which is responsible for generating

[2] https://github.com/budzianowski/multiwoz.

appropriate responses to user queries. To provide an overview, Figure 2 depicts the taxonomy of the recent approaches to the task-oriented dialogue policy learning.

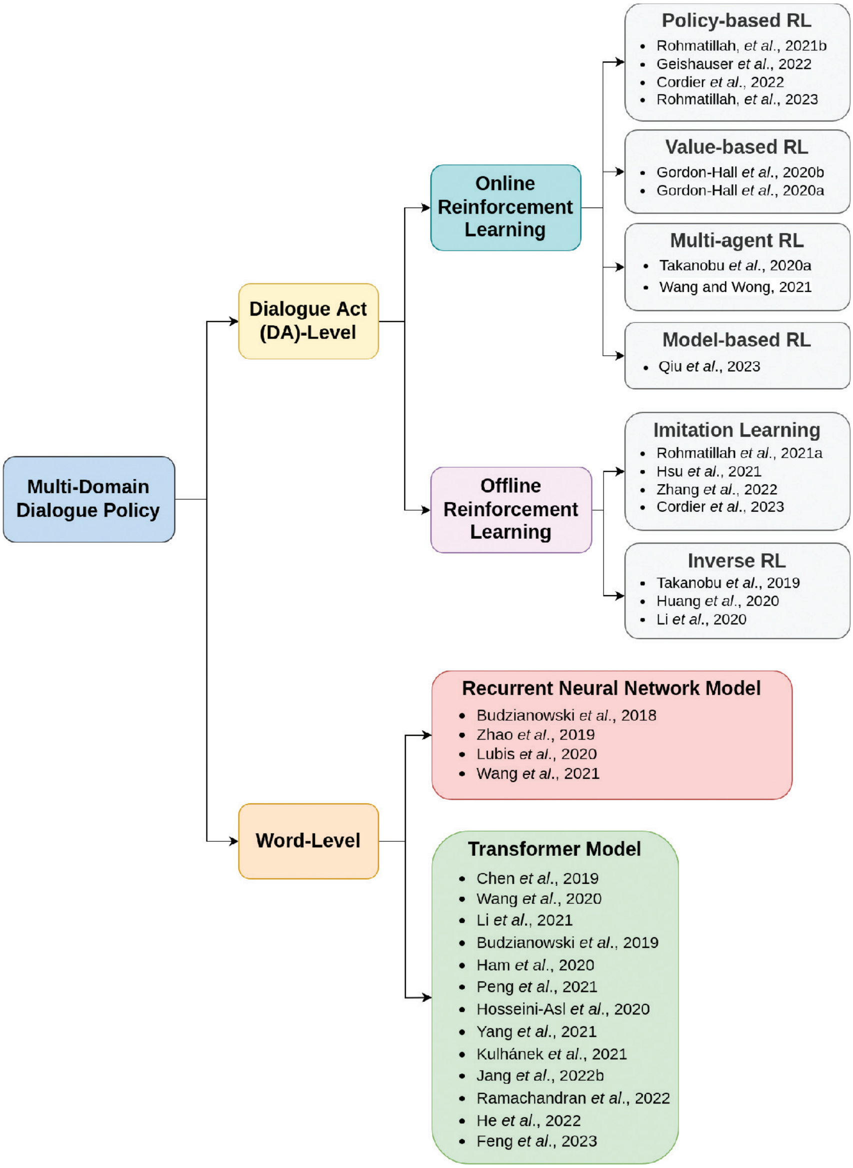


Figure 2: Taxonomy of the multi-domain task-oriented dialogue policy optimization approaches.

3 Policy Optimization in Dialogue Act Level

In this section, the advances in multi-domain task-oriented policy learning through utilization of dialogue act is described. All of the previous methods were either designed with online RL or offline RL paradigms.

3.1 Online Reinforcement Learning for Policy Optimization

Reinforcement learning (RL) has received significant attention as a method to optimize dialogue policies since the learning criterion is directly built through the interaction between the dialogue policy or agent and a simulated user to reflect real-world scenarios. Basically, the interaction is modeled as a partially observable Markov decision process (POMDP) which means that the dialogue agent only receives an observation $\mathbf{o}$ instead of the complete state information $\mathbf{s} = \{g, \mathbf{o}\}$ consisting of user goal g and observation $\mathbf{o}$ where $\mathbf{o}$ is generated by the DST component. The learning objective is to find a policy π^* that maximizes the discounted accumulation of a reward function R via

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t R\left(g, \mathbf{o}_t, \mathbf{a}_t\right) \right] \tag{1}$$

where $\mathbf{a}_t$ denotes the action at time t, and γ and T denote the discount factor and the length of trajectory, respectively.

A number of online reinforcement learning (RL) approaches to dialogue policy have been proposed under different learning settings. An overview of these approaches, categorized from different views or perspectives, is presented in Table 4. The first perspective focuses on the main approach, categorized based on the RL configuration. The standard settings that utilize the policy-based and value-based approaches are surveyed. The next views are the RL optimization method, the simulated user, and the reward definition employed in individual works. Lastly, a brief description of the key idea of each work is provided. This section will address the details of these works. In general, Convlab-2 [118] is popularly adopted in these works as a framework with the reward function to train the dialogue policy π or equivalently build an agent. During the interactions, the dialogue agent receives the reward -1 in every conversation it made, $+5$ if the current domain is satisfied, and $+40$ if the dialogue agent successfully satisfies the user goal. The simulated user is designed according to the agenda-based user simulation [84, 87]. In ConvLab-2, the observation $\mathbf{o}$ is defined as a vector consisting of six different components including user action, system last action, belief state, book information, database pointer and termination. Belief state vector is a vectorized version of user belief, for example [‘hotel’-‘price’-‘expensive’]. Book information is a one-hot vector that indicates whether the system makes a booking in the

Table 4: Summary of online reinforcement learning for MDTD policies.

Reference	RL Type	Optimization Method	Simulated User	Reward	Key Idea
[80]	Policy-based	PPO	Agenda-based	ConvLab-2	Propose human-in-the-loop learning strategy to provide additional feedback
[31]	Policy-based	VTRACE	Agenda-based	ConvLab-2	Reformulate MDTD problem as a continual learning problem
[22]	Policy-based	ACER+IL	Agenda-based	ConvLab-2	Design domain-specific policies by using GNN architecture
[81]	Policy-based	PPO	Agenda-based	ConvLab-2	Combine hierarchical RL with human-in-the-loop learning strategy
[34]	Value-based	DQN	Agenda-based	ConvLab-2	Propose multiple auxiliary loss using expert trajectories
[33]	Value-based	DQN	Agenda-based	ConvLab-2	Design adaptive auxiliary losses based on the quality of trajectories
[95]	Multi-agent	A2C	Learnable User	Role-aware	Propose hybrid value network to alculate role-aware value function
[101]	Multi-agent	DQN	Agenda-based	ConvLab-2	Action space factorization using multiple dialogue agents
[75]	Model-based	DQN	Agenda-based	ConvLab-2	Modify deep dyna-Q with scheduled reward knowledge distillation

specific domain. Database pointer is a vector that represents the results from the entities retrieved from the MultiWOZ database conditioned on user belief. Termination is an indicator of whether the ongoing dialogue is finished or not. Meanwhile, the action **a** is defined as a vector which reveals the system dialogue acts. The interaction between dialogue agent and environment is depicted in Figure 3 where the environment involving a simulated user is configured in the interaction. The dimensions of **a** and **o** are shown.

3.1.1 Policy-Based Method

Policy-based method in dialogue act level is commonly implemented by training a policy π_θ with parameter θ which is estimated according to a policy gradient method by using the following gradient

$$\begin{aligned}\nabla_\theta J(\theta) &= \nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[R(\tau)\right] \\ &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\nabla_\theta \log \pi_\theta(\tau) R(\tau)\right]\end{aligned} \tag{2}$$

where the trajectory τ is obtained by running the policy π_θ in RL environment which involves a simulated user. τ consists of observation **o**, action **a** and reward value r. Advantage actor critic (A2C) [64] modifies the cumulative reward to be an advantage function that can be calculated as follow

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\nabla_\theta \log \pi_\theta(\tau) \hat{A}(\tau)\right] \tag{3}$$

where $\hat{A}(\tau) = R(\tau) - b(\tau)$ with $b(\tau)$ which is a learned baseline function that is commonly represented by a Q or V function.

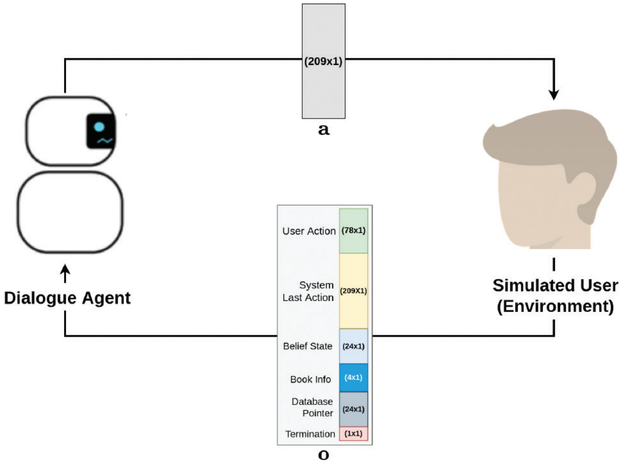


Figure 3: An interaction between dialogue agent and simulated user (or RL environment) through the system action **a** and observation **o** which are defined in Convlab-2 [118].

Among various policy-based optimization methods, the proximal policy optimization (PPO) with a clipped surrogate objective function [86], which is an actor-critic-based method, is seen as a representative approach to train a multi-domain policy. Such an approach has obtained the desirable performance in various RL tasks with the discrete action space [99]. PPO clipping and learning are performed for policy optimization by

$$\theta_{j+1} = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{\mathbf{o},\mathbf{a}\sim\pi_{\theta_j}} \left[L(\mathbf{o},\mathbf{a},\theta_j,\theta)\right] \tag{4}$$

where $L(\mathbf{o},\mathbf{a},\theta_j,\theta)$ denotes the clipped surrogate objective function which is constructed by considering the ratio $\rho(\theta) = \frac{\pi_\theta(\mathbf{a}|\mathbf{o})}{\pi_{\theta_j}(\mathbf{a}|\mathbf{o})}$ between new policy π_θ and old policy π_{θ_j}. The learning objective is constructed in accordance with the estimated advantage function $\hat{A}^{\pi_{\theta_j}}$ and the clipped advantage function $\text{clip}(\rho(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^{\pi_{\theta_j}}(s,a)$ with a clipping threshold ϵ in a form of

$$L(\mathbf{o},\mathbf{a},\theta_j,\theta) = \min\Big(\rho(\theta)\hat{A}^{\pi_{\theta_j}}(\mathbf{o},\mathbf{a}),\ \text{clip}(\rho(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^{\pi_{\theta_j}}(\mathbf{o},\mathbf{a})\Big). \tag{5}$$

The advantage function $\hat{A}^{\pi_{\theta_j}}$ is estimated by using the generalized advantage estimation (GAE) [85] via

$$\hat{A}^{\pi_{\theta_j}}(\mathbf{o}_t,\mathbf{a}_t) = \delta^V(\mathbf{o}_t) + \gamma\lambda\hat{A}^{\pi_{\theta_j}}(\mathbf{o}_{t+1},\mathbf{a}_{t+1}) \tag{6}$$

where $\delta^V(\mathbf{o}_t) = r_t + \gamma\hat{V}_{\phi_j^-}(\mathbf{o}_{t+1}) - \hat{V}_{\phi_j^-}(\mathbf{o}_t)$. Here, r_t is the reward, γ is the discount factor, and λ is a GAE factor for adjusting the bias-variance tradeoff in model construction. Meanwhile, PPO critic parameter ϕ is updated by minimizing the mean squared error between the predicted value function $\hat{V}_{\phi_j}(\mathbf{o}_t)$ and the target value function $y_t = r_t + \gamma \hat{V}_{\phi_j^-}(\mathbf{o}_{t+1})$ where state $\mathbf{o}_{t+1}$ and reward r_t are sampled from the current replay buffer $\mathcal{D}_j$. ϕ_j and ϕ_j^- denote the recently updated and the delayed critic weights in current epoch, respectively. The critic parameter ϕ_{j+1} is updated by minimizing the temporal difference (TD) error [12] of value function $\hat{V}_{\phi_j}(\mathbf{o})$

$$\phi_{j+1} = \underset{\phi_j}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{o}\sim\mathcal{D}_j}\left[\left(y - \hat{V}_{\phi_j}(\mathbf{o})\right)^2\right]. \tag{7}$$

Due to the situation of sparse reward in real-world application, the other related works [80, 81] applied the human-in-the-loop (HITL) learning paradigm [1] to provide additional feedback from the expert in every conversation turn. The expert was represented by a pre-defined rule-based policy. Therefore, two different kinds of feedback were proposed. One was for action correction, and the other was for action evaluation. Action correction aimed to intervene the dialogue agent action by providing the correct response to the simulated user in order to make the conversation continue until reaching the end of user goal. This feedback is crucial as once the dialogue agent produces the wrong answer, then this agent may be trapped in the same conversation topic until reaching the maximum number of conversation turns. An example of this scenario is shown in Table 5. Meanwhile, the action evaluation feedback was designed to provide additional reward in every conversation turn by calculating the behavior cloning loss [74] between the dialogue agent action and the expert action given the obtained $\mathbf{o}$ in every turn.

Another approach is to combine the HITL paradigm with a hierarchical reinforcement learning (HRL) to simplify the multi-domain task complexity [81] by treating each dialogue session uniquely based on the predicted base domain. The base domain is defined as the domain that becomes the primary concern in each dialogue session and is always present at the beginning of the conversation. In [81], the high-level policy initiates a sub-policy in the low-level policy that matches the current dialogue's base domain. This happens only once at the beginning of the conversation. It is a sensible approach to create a task-oriented dialogue system that meets the user's objective in a very short time, typically fewer than 15 time steps. This is much faster than standard HRL tasks such as maze or robotic tasks that may require hundreds to thousands of time steps to complete [66]. By using HRL in this scenario, the task's complexity, which involves the extensive state and action spaces, can be reduced, resulting in more efficient training of the dialogue policy. In this setting, the high-level policy was optimized by using standard policy gradient method, called REINFORCE. Meanwhile, the sub-policies or low-level policies

Table 5: Example of showing three confounded states which trap agent for being stuck and their corresponding actions. The repetition of dialog acts from both user and system can be a clear indicator for agent to ask for human corrections.

Goal Id	User Dialog Act	User Sentence	System sentence
14	[['Request', 'Hotel', 'Addr', '?']]	Could I get their address?	Their address is Regent Street City Centre. Is there anything else I can help you with?
14	[['Request', 'Hotel', 'Addr', '?']]	I just need the address of the hotel.	Their address is Regent Street City Centre. Is there anything else I can help you with?
14	[['Request', 'Hotel', 'Addr', '?']]	No, I just need their address.	Their address is Regent Street City Centre. Is there anything else I can help you with?
44	[['Request', 'Hotel', 'Stars', '?']]	How many stars?	Did you need any further assistance today?
44	[['Request', 'Hotel', 'Stars', '?']]	Can you please tell me how many stars it has?	Is there anything else I can help you with?
44	[['Request', 'Hotel', 'Stars', '?']]	How many stars it is?	Is there anything else I can help you with?
85	[['Request', 'Hotel', 'Area', '?'], ['Request', 'Hotel', 'Phone', '?']]	What area is it in? What is their phone number?	The attraction phone number is 01223902088. Did you need any further assistance today?
85	[['Request', 'Hotel', 'Area', '?'], ['Request', 'Hotel', 'Phone', '?']]	Yeah, what area of town is it in? Could I get the phone number of the hotel?	The attraction phone number is 01223902088. Is there anything else I can help you with?
85	[['Request', 'Hotel', 'Area', '?'], ['Request', 'Hotel', 'Phone', '?']]	What's the area at first? Do you have their phone numbers?	The attraction phone number is 01223902088. You are welcome. Is there anything else I can help you with today?

were trained by using PPO. Figure 4 illustrates the procedure of implementing the policy optimization by using HITL-HRL method.

In addition to formulating the MDTD task as a sequence of different tasks based on the corresponding base domain using HRL, an alternative problem formulation was proposed in [31]. This paper considered MDTD task as a continual learning problem, aiming to enable the dialogue policy to quickly adapt to a new domain without forgetting the previously learned knowledge from previous domains. This approach addressed the challenges of retaining knowledge and adapting efficiently to new domain in a continual learning setting, and presented the so-called VTRACE [27] as the dialogue

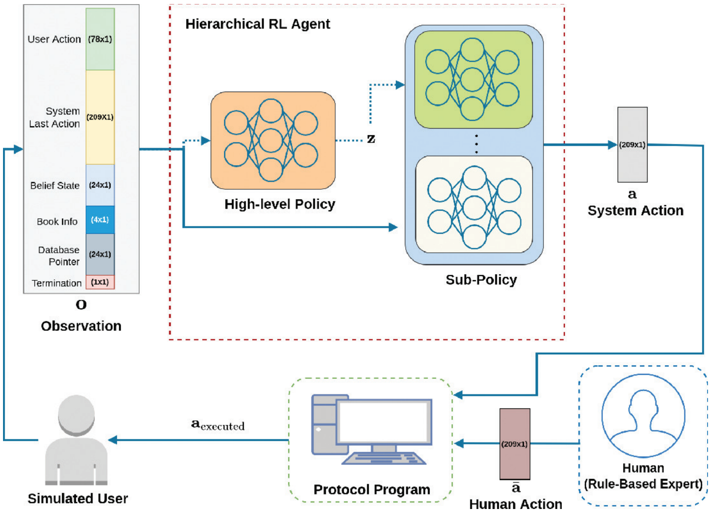


Figure 4: An overview of human-in-the-loop learning combined with the hierarchical reinforcement learning. $\mathbf{a}_{\text{executed}}$ denotes the action applied to the environment determined by the protocol program. $\mathbf{z}$ denotes the output of high-level policy which will activate one of the sub-policies.

policy. VTRACE is a variant of off-policy actor-critic algorithm with the advantage function. In order to mitigate the catastrophic forgetting and allow rapid adaptation, this method proposed the sampling strategy from [46] which sampled the non-recent experience from the replay buffer.

Owing to the importance of the relations among different features in multi-domain dialogue policy, an intuitive approach based on graph neural network (GNN) was proposed to handle various MDTD tasks. Recent approaches that developed GNN for multi-domain dialogue policy [22] adopted the domain independent parameterization [104] to characterize the inter-slot relations in the feature space. Furthermore, domain-specific modules were proposed to decompose multi-domain problems into several single-domain problems. Due to the introduction of domain-specific modules, GNN-based approaches were also considered as the hierarchical and multi-task learning methods. A domain-specific module was activated by the DST module following the user input domain. Each node and each directed edge in every domain-specific module represented different slot names and message passing, respectively. To train the dialogue policy, a combination of imitation learning (IL) and reinforcement learning based on standard actor critic with experience replay (ACER) [105] was exploited. Half of the stored trajectories were obtained from oracles, while the other half were collected from the interaction between dialogue agent and simulated user.

3.1.2 Value-Based Method

To implement the value-based policy for multi-domain dialogue, the dialogue agent is built by using the Q network in dialogue act level. The learning objective follows the theory of Q learning [107] in which the total cumulative discounted reward over a trajectory can be estimated according to a Q function which is expressed by

$$Q(\mathbf{o}, \mathbf{a}) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^{t} R\left(g, \mathbf{o}_{t}, \mathbf{a}_{t}\right)\right]. \tag{8}$$

T is the length of trajectory while t and γ are the current time step and the discounted factor, respectively. The action $\mathbf{a}$ of dialogue policy in every turn with observation $\mathbf{o}$ can be obtained through the greedy calculation on the predicted Q function

$$\pi^{*}(\mathbf{o}) = \underset{\mathbf{a}}{\operatorname{argmax}}\, Q(\mathbf{o}, \mathbf{a}). \tag{9}$$

Deep Q network (DQN) [65], which is the implementation of Q learning with the neural network architecture, is feasible to carry out the value-based policy in dialogue act level. In order to allow the dialogue agent to do exploration, ϵ greedy algorithm is employed so that the dialogue policy might produce a random action with probability ϵ. The learning objective with a replay buffer $\mathcal{D}$ is therefore defined by a regression loss given by

$$\phi^{*} = \underset{\phi}{\operatorname{argmin}}\, \mathbb{E}_{(\mathbf{o},\mathbf{a},r)\sim\mathcal{D}}\left[\left(r + Q_{\phi'}(\mathbf{o}_{t+1}, \mathbf{a}_{t+1}) - Q_{\phi}\left(\mathbf{o}_{t}, \mathbf{a}_{t}\right)\right)^{2}\right] \tag{10}$$

where ϕ' and ϕ denote the parameters of the delayed Q network and recently updated Q network, respectively. Due to the sparse reward problem, recent approaches have employed the deep Q learning from demonstration [40] that utilized the dialogue data $\mathcal{D}$ from the expert. In [34], a constant was added as the auxiliary loss if the output of dialogue policy was different from the expert. Meanwhile, an adaptive auxiliary loss was proposed to provide different penalties according to the quality of the stored trajectories [34].

3.1.3 Multi-Agent Method

In addition, the multi-domain dialogue system was implemented according to the multi-agent policy in dialogue act level. There are two approaches to fulfill the multi-agent policy optimization as shown in Figure 5. The first one is to train multiple dialogue agents which play two individual roles [95]. One role acts as the learnable user agent and the other role acts as a dialogue policy agent. The communication between these two agents is modeled as a kind of collaborative interaction so as to achieve the final goal. The main motivation

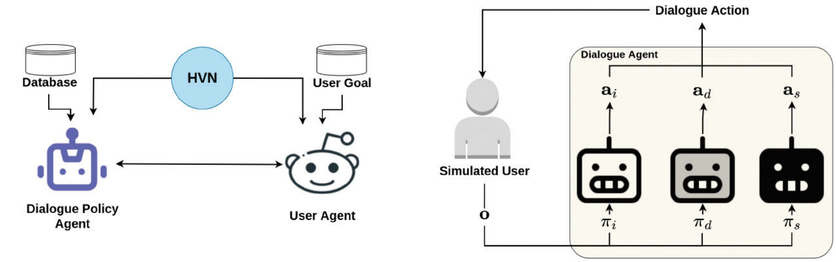


Figure 5: Approaches to multi-agent policy optimization. (Left) Multi-agent RL to jointly train dialogue policy agent and user agent. (Right) Multi-agent RL for action decomposition from intent policy π_i, domain policy π_d and slot policy π_s.

of this approach is to handle the difficulty of designing a simulated user that sufficiently represents human behaviors. However, designing a rule-based simulated user requires domain expertise, meanwhile training a data-driven simulated user [26, 35, 50] requires an abundance of manually labeled data. In order to ensure a stable learning between two different agents, the previous work [95] proposed a hybrid value network (HVN) for role-aware reward decomposition into the global reward and the role-specific rewards. There are two role-specific rewards including system reward and user reward. System reward is basically defined as the task success and the system response quality conditioned on the user requested slot. Meanwhile, the user reward is defined based on how naturally the user expresses the goal to the system. The global reward was identical to the common reward in the RL optimization. HVN was trained by considering the system reward, user reward, global reward, system observation and user observation so as to provide the critic value to optimize both dialogue policy agent and user agent.

The second multi-agent policy optimization is to model the intent, slot, and value in the dialogue act as the independent dialogue policy in order to reduce the complexity of action space [101]. A previous work proposed the joint optimization based on the independent experience replay buffers that were used to train each dialogue policy agent based on its role. The domain policy π_d and its corresponding action a_d reflect the current domain information in the prediction. The intent policy π_i selected an appropriate intent a_i given the predicted domain by π_d and $\mathbf{o}$. Lastly, the slot policy π_s produces the corresponding slot a_s conditioned on $\mathbf{o}$, a_d and a_i. Different from the previous approach that defined different reward function for each individual agent, in this approach, all dialogue agents receive identical reward from one reward function. Furthermore, all of them were optimized by using DQN and the learning objective was formed as the TD error which was minimized as expressed in (7) and (10).

3.1.4 Model-Based Method

Model-based RL [17, 49] has been developed and employed in policy optimization in dialogue act level where the dialogue policy is optimized to conduct a planning step. Such a step aims to predict what happens if the agent takes a specific action in the current time step. Furthermore, a planning step is performed to pursue sample efficiency in reinforcement learning. To implement the planning step, model-based RL is performed to build a predictive world model to represent the environment through RL optimization. For the case of a dialogue system, the world model is designed to mimic the user behavior in a dialogue conversation. Traditionally, the world model is represented by a sequence-to-sequence model based on the gated recurrent unit (GRU) [18] which captures the dynamics from real environment. GRU is trained according to the supervised learning through leveraging the human annotated dataset and the trajectories stored in the replay buffer. Given the current observation and action from the dialogue agent $(\mathbf{o}, \mathbf{a})$, the world model will output three predictions which are $\hat{\mathbf{a}}_{\text{user}}$, $\hat{r}$ and $\hat{T}$. This process is called as the planning step. $\hat{\mathbf{a}}_{\text{user}}$ is an estimated user action that will be employed in the simulated RL environment together with the estimated terminal condition $\hat{T}$ to obtain the estimated next time step observation $\hat{\mathbf{o}}'$. World model also outputs the predicted reward $\hat{r}$ that might differ from r which is obtained from the interaction with the simulated user. Figure 6 shows the interaction between the dialogue agent and the simulated user in the model-based RL.

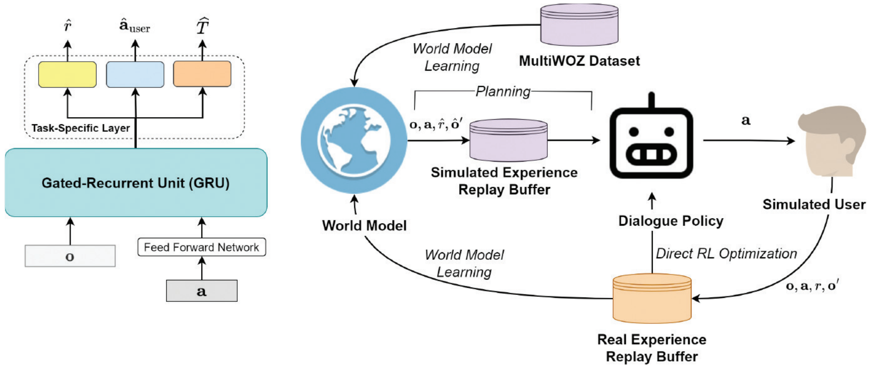


Figure 6: An illustration of world model (Left) and model-based RL for policy optimization in MultiWOZ (Right).

In [75], the deep dyna-Q (DDQ) framework [72], which is an advanced version of dyna-Q [94], was proposed as a popular value-based approach to learn the model-based policy. In DDQ learning, the dialogue agent learned from two experience replay buffers. The first was identical to the standard

RL which used the stored trajectories gathered from the interaction with the simulated user. On the other hand, the second experience replay stored the trajectories gathered from the planning steps. In case of using MultiWOZ dataset in a sparse reward setting, an additional network [75] was introduced to estimate a reward through the scheduled knowledge distillation. Using this method, two different reward functions were designed. One was the teacher reward function and the other was the student reward function. The teacher reward function had an access to the user goals and was pre-trained by using the weakly supervised learning. Meanwhile, the student reward only had the access to observation $\mathbf{o}$. The knowledge distillation process was scheduled according to the divergence between teacher and student reward estimations.

3.2 Offline Reinforcement Learning for Policy Optimization

Offline learning is a machine learning technique that aims to learn a policy by leveraging the stored trajectories in the dataset. Learning from a collection of trajectories is comparable to fulfill the imitation learning. There are two main objectives designed for optimization in imitation learning. The first one is to directly learn an expert policy from the given dataset. Behavior cloning [74] is the simplest approach among the methods of offline RL because this approach can be simply done by only considering the standard classification loss without requiring any interaction with a simulated user. Another solution is the generative adversarial imitation learning (GAIL) [41], where the model is trained to deceive the discriminator model. Consequently, the discriminator becomes unable to differentiate between the trajectories generated by the expert and those generated by the learned dialogue policy. The second one is to estimate a learnable reward function [29, 68] or equivalently build a reward model to mitigate the sparse reward problem for policy learning in dialogue act level. The processes of reward model learning and dialogue policy learning can either be alternatively done or divided into two sequential learning stages. Traditionally, the learned reward model is accordingly implemented during RL training with a simulated user. However, in a recent work [45], the learned reward function can be used as a learning criterion in the supervised learning setting. Table 6 shows a summary of individual methods which utilize offline RL paradigm in MDTD task. These methods with different perspectives will be detailed in what follows.

3.2.1 Imitation Learning Method

In general, naively applying imitation learning to directly mimic expert actions can lead to two main challenges which are causal confusion [36] and distributional shift [6]. Causal confusion often arises in behavior cloning (BC) or

Table 6: Summary of offline reinforcement learning for MDTD policies.

Reference	Offline RL Type	Optimization Method	Simulated User	Key Idea
[80]	Imitation L.	BC	–	Auxiliary loss function to mitigate causal confusion problem
[44]	Imitation L.	GAIL	Agenda-based	Define MDTD task as a multi-task problem by combining HRL with GAIL
[113]	Imitation L.	BC	–	Decompose multi-label classification by using model-based imitation learning
[21]	Imitation L.	BC	–	Imitation learning with the combination of GNN and HRL
[96]	Inverse RL	AIRL	Agenda-based	Transform session-level reward to observation-action level reward estimation using AIRL
[45]	Inverse RL	IRL	Optional	Observation-action level reward estimation representing the RL environment dynamics
[58]	Inverse RL	AIRL	Agenda-based	Design AIRL method without adversarial learning in the loop

standard imitation learning, where the agent struggles to determine the true cause behind the expert actions. This issue is obvious when the model input consists of diverse features, such as in the MDTD task where $\mathbf{o}$ comprises six distinct features. Inspired by the solution to mitigate causal confusion in a self-driving car task [36], the auxiliary tasks [79] were proposed as the learning regularization techniques to address this problem. These regularization terms compelled the dialogue agent to focus on specific features within $\mathbf{o}$ that were likely to have a significant influence on the expert actions. Accordingly, the optimization of imitation learning of policy parameters with auxiliary tasks is formulated by

$$\theta^* = \operatorname*{argmin}_{\theta} \mathbb{E}_{(\mathbf{o},\mathbf{a},\mathbf{y}^{\text{aux}})\sim\mathcal{D}} \left[\mathcal{L}\left(\pi\left(\mathbf{o};\theta\right),\mathbf{a}\right) + \mathcal{L}\left(\pi\left(\mathbf{o};\theta\right),\mathbf{y}^{\text{aux}}\right)\right] \tag{11}$$

where $\mathcal{D}$ denotes the stored trajectories and $\mathbf{y}^{\text{aux}}$ denotes the label vector in auxiliary tasks, for example the labels for predicting the current belief state and the user action. Another approach [21] proposed the structured policies built upon the GNN model to improve input representation for the dialogue agent. The fully connected and directed GNN was suitable to handle multiple features from $\mathbf{o}$ to allow effective knowledge sharing among different inputs from different domains. In order to further improve the learning efficiency, a hierarchical network structure was proposed in [21]. Accordingly, the dialogue agent could adapt faster in a few-shot setting.

On the other hand, the issue of distributional shift happens due to the dissimilarity between the trajectories of the expert and the learned agent, which creates the bias in the dataset. This bias arises because the trajectories or state-action pairs in the dataset do not encompass all of the scenarios present in actual demonstrations. Consequently, in a diverse environment, the agent may generate an unsuccessful trajectory when an incorrect action is taken simply because the agent encounters an unfamiliar situation. To overcome this issue, the process of data augmentation [19] can be employed to introduce more data that can better generalize the phenomenon in an unseen environment. The amount of training data was increased. Meanwhile, the distribution shift is likely caused when the agent misidentifies the genuine factors behind an expert's action in a specific state during the training stage. This difficulty is further intensified by stochastic gradient descent learning, which generally presumes that all pairs of data are independent and identically distributed.

In order to cope with the distributional shift, the previous work further combined the hierarchical RL and GAIL [41] to carry out the so-called multi-task generative adversarial imitation learning (MGAIL) [44] to deal with the multi-domain dialogue representation. MGAIL defined seven different experts that represented individual domains in the MultiWOZ dataset. The dialogue representation was enriched. Furthermore, due to the implementation of hierarchical RL, the multi-domain problem was simplified as the multiple single-domain problems which facilitate the dialogue policy learning. In addition, MGAIL works similarly to the generative adversarial network (GAN) [32]. MGAIL contains a discriminator that distinguishes whether the input trajectories come from the expert π_e or from the generator which is the learned dialogue policy π. Therefore, the objective of MGAIL is formed by a minimax optimization problem

$$\begin{aligned}\min_{\pi}\max_{D}\,&\mathbb{E}_{(\mathbf{o},\mathbf{a},d)\sim\pi}[\log D(\mathbf{o},\mathbf{a},d)]\\&+\mathbb{E}_{(\mathbf{o},\mathbf{a},d)\sim\pi_e}[\log(1-D(\mathbf{o},\mathbf{a},d))]-\lambda H(\pi)\end{aligned} \tag{12}$$

where the entropy of a policy $H(\pi)\triangleq\mathbb{E}_{\pi}[-\log\pi(\mathbf{a}|\mathbf{o})]$ is seen as a regularizer with parameter λ, D denotes the discriminator and d denotes the domain information. Dialogue policy is therefore learned according to a two-player game theory which implements an adversarial optimization over policy π and discriminator D. Another work [113] proposed a model-based imitation learning to decompose a multi-label classification problem into multiple single-label classification problems by using multiple planning steps to avoid a distributional shift problem. The main motivation was to reformulate the problem as the multi-label classification which was considered as multiple turns in the single-label classification. To provide an accurate action prediction, this work proposed a K-planning path and then used the ensemble prediction to aggregate all K planning paths to identify the final dialogue agent action.

3.2.2 Inverse Reinforcement Learning Method

An alternative approach to implementing offline RL in dialogue policy learning involves learning a reward model within the framework of inverse RL (IRL) [68]. The guided dialogue policy learning (GDPL) [96] method was the first to leverage IRL to learn a reward model in the MultiWOZ dataset. Specifically, GDPL utilized adversarial IRL (AIRL) [29] to learn the reward model. The concept of adversarial learning in AIRL is similar to that in GAIL, as depicted in (12). However, instead of using the discriminator to directly improve the dialogue policy, AIRL utilized it to construct a reward model for reinforcement learning. Since using the trajectories as discriminator input, like in GAIL, can be computationally inefficient, GDPL proposed a reward estimation in observation-action level. This approach allowed for the immediate evaluation of the dialogue agent's output. The reward modeling and dialogue policy learning were performed alternately, with the dialogue policy being optimized by using the proximal policy optimization (PPO).

However, the alternative update between the reward model and dialogue policy may lead to suboptimal solutions due to the mode collapse in adversarial optimization. Previous approaches [45, 58] have considered sequential updates between the reward model and dialogue policy. In sequential learning, the reward model was first trained and then frozen before being incorporated into the RL process to update the dialogue policy. However, addressing the issue of mode collapse during reward model learning in sequential stages was quite challenging. This is because relying solely on the collected dataset in an adversarial learning setting may not be sufficient to enable the discriminator to effectively distinguish between observation-action pairs from the expert and non-expert sources.

To tackle this challenge, a variational autoencoder (VAE) was trained to generate adversarial examples by sampling exclusively from the prior distribution. These adversarial examples were then used to train the discriminator model [58]. Additionally, the encoder component of the VAE projected the discrete representation of $\mathbf{o}$ into a continuous space, allowing for similar samples of $\mathbf{o}$ to be mapped to closely located latent representations. In particular, VAE objective is formed as an evidence lower bound (ELBO) of log likelihood

$$L_{\text{vae}}(\omega, \psi) = \mathbb{E}_{\mathbf{z} \sim q_{\omega}(\mathbf{z}|\mathbf{o})} \left[\log p_{\psi}\left(\mathbf{o}|\mathbf{z}\right)\right] - \text{KL}(q_{\omega}\left(\mathbf{z}|\mathbf{o}\right) \| p\left(\mathbf{z}\right)). \tag{13}$$

q_ω and p_ψ denote the variational posterior and generative likelihood corresponding to VAE encoder and VAE decoder with parameters ω and ψ, respectively. Meanwhile, $\mathbf{z}$ is a latent representation from $\mathbf{o}$ and is sampled from $q_\omega(\mathbf{z}|\mathbf{o})$, which is a Gaussian distribution with mean and variance values from encoder outputs. This Gaussian is regularized by getting close to a prior as standard Gaussian $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ via an objective in term of Kullback-Leibler (KL) divergence. The objective of generative likelihood $p_\psi(\mathbf{o}|\mathbf{z})$ by using the latent sample $\mathbf{z}$ is taken into account to implement data reconstruction for observation $\mathbf{o}$ in the decoder. The observation $\mathbf{o}$ after the VAE encoder, denoted by $\text{Enc}(\mathbf{o})$ is concatenated with the action $\mathbf{a}$ from the dataset to obtain the real state-action representation. Besides, the adversarial or simulated state-action vector is generated by sampling $\mathbf{z}$ from standard Gaussian $\mathcal{N}(0, \mathbf{I})$ and passing through two individual feedforward networks or generators for $\mathbf{o}$ and $\mathbf{a}$. A discriminator is introduced to distinguish whether the state-action representation is real or simulated. Therefore, the generator and discriminator networks are jointly trained for state-action representation for dialogue policy learning.

In addition to the standard variational autoencoder (VAE), the variational recurrent neural network (VRNN) [20, 47] can be utilized in this learning scenario, where the observations from previous time steps are taken into account when calculating the latent representation $\mathbf{z}$ at the current time step. This idea was proposed in [45] where a reward function was constructed to capture the dynamics within the RL environment. The ground-truth dynamics were represented by the expert trajectories stored in the dataset. In this setup, the reward model took both the observation and the embedding of the action as inputs. The inclusion of the action embedding was designed to enhance the model's generalization capabilities. Besides being used for RL training, the reward model was also employed as a learning criterion for optimizing the dialogue policy through supervised learning [45].

4 Policy Optimization in Word Level

On the other hand, the multi-domain dialogue policy can be implemented in word level where the dialogue policy and natural language generation (NLG) are combined into a single processing component. Different from the standard NLG methods that generate the responses conditioned only on the dialogue context, the word-level dialogue policy produces the responses not only conditioned on a dialogue context but also on the predicted dialogue act. Basically, the majority of word-level dialogue policies receive a sequence of word tokens and utilize a text encoder to provide supplementary characteristics for dialogue policy. The ground-truth belief state and database (DB) pointer are given in both training and test stages. This input format is the prominent difference between word-level and dialogue act-level policies in which the dialogue act-level policy only receives $\mathbf{o}$ generated by the DST component. Another difference is that the belief state and DB pointer in the word-level policy can be either a multi-hot vector like in the dialogue-act level policy or a sequence of word tokens. However, due to the remarkable performance of transformer-based decoder using GPT-2, numerous studies have been conducted for various generative tasks. Even though the learning criterion was designed to optimize different components in a dialogue system components by sequentially connecting the optimal generation from individual components, this kind of approach can still be considered as the dialogue policy optimization by bypassing the generation of belief state. In other words, belief state information can be obtained from either another model or a dataset with the ground-truth of belief state. This scenario is common in the MultiWOZ evaluation.

This section elaborates the recent approaches to word-level multi-domain dialogue policy, which can be categorized into two kinds of models. One is the approaches that are purely designed by using the recurrent neural network models. Another one is the approaches which introduce the transformer models. For the latter category, this survey initially describes the approaches which utilized the transformer encoder model such as BERT. Next, the description of transformer decoder-based approaches such as fine-tuning the GPT model is provided. A summary of individual methods under the catetory of word-level policy learning is shown in Table 7. Different methods will be detailed and compared in what follows.

4.1 Recurrent Neural Network Model

During the early stage of word-level dialogue policy optimization, the prevailing method was based on the sequence-to-sequence learning which relied on a feedforward neural network-based encoder to encode the dialogue context $\mathbf{c}$ and a recurrent neural network (RNN)-based decoder to generate the word sequence of dialogue response. Here, $\mathbf{c}$ contains a sequence of word tokens representing the conversation history between the user and the system. As shown in Figure 7, the encoder output is combined with belief state ($\mathbf{bs}$) and database ($\mathbf{db}$) pointer and then fed into a linear transformation to calculate latent representation $\mathbf{z}$ that represents the system dialogue act, for example ['hotel'-'inform'-'location']. This information is crucial for dialogue policy optimization which is definitely different from the optimization of a pure natural language generation where the condition on dialogue act is missing. The RNN decoder based on long short-term memory (LSTM) [42, 90–92] is implemented to decode the dialogue response word by word. The teacher-forcing scheme is usually applied to predict the next word by forcing the input from the previous word.

To learn such a word-level dialogue policy, the simplest approach is to use the supervised learning objectives. However, this approach may suffer from an exposure bias issue due to the teacher-forcing implementation in the training stage [38] which could not sufficiently generalize to test environment. An alternative solution is to carry out a reinforcement learning (RL) method to optimize the word-level model. One challenge with this approach is that the dialogue policy action is defined as the selection of individual generated word, which results in a very large action space because of the size of the dictionary. Additionally, defining a useful reward function which is able to represent the

Table 7: Summary of word-level policy optimization in MDTD task. Notations E, D and RM stand for encoder, decoder and retrieval module, respectively.

Reference	Architecture	Training Method	Key Idea
[4]	- LSTM (E) - LSTM (D)	Train from scratch	Baseline of MultiWOZ dialogue policy
[116]	- GRU (E) - LSTM (D)	Train from scratch	Represent dialogue policy action as a latent variable
[61]	- GRU (E) - LSTM (D)	Train from scratch	Modify LaRL model by adding additional encoder
[102]	- GRU (E) - LSTM (D)	Train from scratch	Hierarchical dialogue action using VHRED network
[8]	- Transformer (E) - BERT + DSA (D)	- Train from Scratch (E) - Fine-tune (D)	- Disentangle different dialogue actions using DSA module - Hierarchical dialogue actions
[103]	- Transformer (E) - Transformer (D)	Train from scratch	Modify multi-label classification in HDSA to be a sequential classification
[57]	- GRU (E) - GRU (D) - BERT (RM)	Train from scratch	Use pre-trained BERT as a context-aware retrieval module
[3]	GPT-2	Fine-tune	The first work applying GPT-2 for word-level dialogue policy
[37]	GPT-2	Fine-tune	Modify previous work by generating system dialogue acts and responses sequentially
[71]	GPT-2	Fine-tune	Pre-train GPT-2 using various MDTD dataset before fine-tune it to MultiWOZ
[43]	GPT-2	Fine-tune	Remove token delimiter in the GPT-2 fine-tuning
[109]	GPT-2	Fine-tune	The first work that fine-tunes GPT-2 in dialogue session level
[52]	GPT-2	Fine-tune	Introduce data augmentation method using back-translation
[48]	GPT-2	Fine-tune	Introduce data augmentation method using a critic network based on DQN
[77]	BART	Fine-tune	Design a pairwise reward functionfor offline RL optimization
[39]	UniLM	Fine-tune	- Design unified dialogue act from 8 different dataset - Semi-supervised pre-training using consistency regularization
[28]	-BART -UniLM	Fine-tune	Introduce two reward functions that not only consider pairwise trajectories

meaningful feedback over the whole trajectory is challenging. Basically, each trajectory requires a unique definition of reward function. For example, if the RL process considers the number of turns in a dialogue session as the reward,

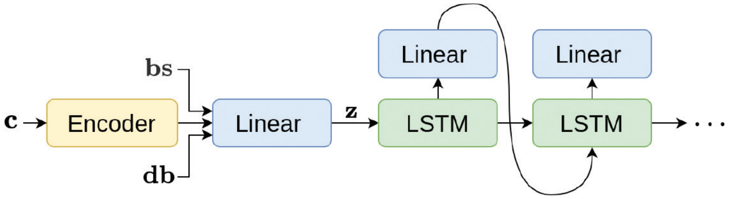


Figure 7: A word-level multi-domain dialogue policy via encoder-decoder framework where the recurrent neural network-based decoder is used.

the word-level policy gradient is then formulated as

$$\nabla_\theta J(\theta) = \mathbb{E}_\theta \left[\sum_{t=0}^{T} \sum_{j=0}^{U_t} r_{t,j} \nabla_\theta \log \pi_\theta \left(w_{t,j} | w_{<t,j}, \mathbf{c}_t \right) \right]. \tag{14}$$

Here, π_θ denotes the word-level dialogue policy, $w_{t,j}$ denotes the word token j in a conversation turn t, which is conditioned on previous tokens $w_{<t,j}$, $R_{t,j}$ denotes the reward function, $\mathbf{c}_t$ denotes the dialogue context, and U_t denotes the number of the generated tokens in every dialogue turn t. There are totally T turns. Notably, the policy network is to select a word $w_{t,j}$ for each token j in each turn t. If the RL setting only considers the evaluation of a specific conversation turn, the summation over T turns in (14) is discarded. The action space in RL for word selection is huge.

In order to mitigate the huge action space in RL process, an effective approach, called the latent action RL (LaRL) [116], was proposed to run RL in latent space by using the latent features $\mathbf{z}$ of words via an encoder. Dialogue generation is then implemented by using these latent features [11]. Accordingly, the policy gradient is calculated in latent space rather than word space by

$$\nabla_\theta J(\theta) = \mathbb{E}_\theta \left[\sum_{t=0}^{T} r_t \nabla_\theta \log \pi_\theta \left(\mathbf{z} | \mathbf{c}_t \right) \right]. \tag{15}$$

In this case, the policy π_θ is seen as an encoder and the reward function can be defined as the success rate of dialogue task. Because the dialogue context $\mathbf{c}_t$ contains the ground-truth responses from both user and system, this learning setting is similar to that in the contextual bandit learning [54]. In this study, the decoder p_ψ is modeled by a long short-term memory network [42] which was pre-trained along with the encoder model by maximizing the following ELBO

$$L_{\text{larl}}(\theta, \psi) = \mathbb{E}_{\mathbf{z} \sim \pi_\theta(\mathbf{z}|\mathbf{c})} \left[\log p_\psi \left(\mathbf{y} | \mathbf{z} \right) \right] - \text{KL}(\pi_\theta \left(\mathbf{z} | \mathbf{c} \right) \| p \left(\mathbf{z} \right)). \tag{16}$$

Importantly, different from the dialogue policy optimization with dialogue act using the observation $\mathbf{o}$ from DST component, $\mathbf{y}$ is a system response for dialogue policy optimization in word level given the current dialogue context $\mathbf{c}$ represented in the token level. During the RL training, the decoder network will be frozen. The posterior $\pi_\theta(\mathbf{z}|\mathbf{c})$ can be either a Gaussian or categorical distribution. If the posterior is a categorical distribution, the prior of $\mathbf{z}$ can be a uniform distribution.

Because the input data in multi-domain dialogue contain rich information, it is crucial to capture such an information for word-level dialogue representation. An approach called the latent action via VAE (denoted by LAVA) [61] was proposed to sequentially train two VAE encoders in the pre-training stage. One is for dialogue response $\mathbf{y}$ and the other is for dialogue context $\mathbf{c}$. The encoder of dialogue context $\mathbf{c}$ was then fine-tuned by using standard policy gradient the same as that in LaRL. At the beginning, the first VAE encoder with parameter ϕ_1 was trained to learn the reconstruction of system response $\mathbf{y}$ by maximizing the ELBO [106]

$$L_{\text{lava}}(\theta, \phi_1) = \mathbb{E}_{\mathbf{z}\sim\pi_{\phi_1}(\mathbf{z}|\mathbf{y})}\left[\log p_\theta\left(\mathbf{y}|\mathbf{z}\right)\right] - \mathrm{KL}(\pi_{\phi_1}\left(\mathbf{z}|\mathbf{y}\right)\|p\left(\mathbf{z}\right)). \tag{17}$$

Then, the second VAE encoder with parameter ϕ_2 was optimized according to the same VAE objective in (17) by replacing π_{ϕ_1} with π_{ϕ_2} and considering $\mathbf{c}$ as the input instead of $\mathbf{y}$. After two pre-training stages, two encoders $\{\phi_1, \phi_2\}$ and one decoder θ are jointly optimized by maximizing the ELBO

$$L_{\text{lava}}(\theta, \phi_1, \phi_2) = \mathbb{E}_{\mathbf{z}\sim\pi_{\phi_2}(\mathbf{z}|\mathbf{c})}\left[\log p_\theta\left(\mathbf{y}|\mathbf{z}\right)\right] - \mathrm{KL}(\pi_{\phi_2}\left(\mathbf{z}|\mathbf{c}\right)\|\pi_{\phi_1}\left(\mathbf{z}|\mathbf{y}\right)) \tag{18}$$

by considering the output of the first VAE encoder $\pi_{\phi_1}\left(\mathbf{z}|\mathbf{y}\right)$ as the prior distribution as shown in the final stage of VAE training in Figure 8. LAVA consists of two pre-training stages and one final stage. Finally, the encoder π_{ϕ_2}

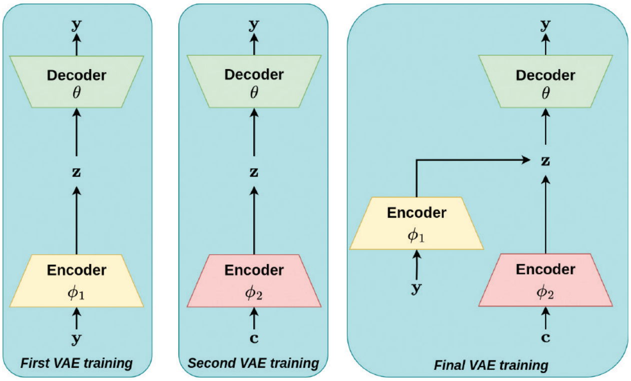


Figure 8: A word-level multi-domain dialogue representation with two VAE encoders for system response $\mathbf{y}$ and dialogue context $\mathbf{c}$ with parameters ϕ_1 and ϕ_2, respectively. Decoder with parameter θ is used for dialogue reconstruction.

is fine-tuned by using (15). For word-level dialogue representation, the encoder and decoder were implemented by GRU and LSTM, respectively. During inference, $\pi_{\phi_1}(\mathbf{z}|\mathbf{y})$ was not be used.

Another alternative to word-level dialogue policy learning is to formulate the encoder-decoder structure as a hierarchical reinforcement learning (HRL) problem. Such an approach builds a hierarchical structure between dialogue policy and natural language generation with option framework (denoted by HDNO [102]), namely considers the option framework setting [93]. The option framework refers to a way of breaking down a complex task into a sequence of sub-tasks, each of which can be executed by a separate policy, known as an option. In HDNO, the encoder acts as the high-level policy which generates the system dialogue act as a latent variable $\mathbf{z}$. The input to the encoder is a dialogue context vector $\mathbf{c}$, created by concatenating the text features extracted by a GRU encoder with the belief state vector and the database pointer vector. The context vector and dialogue context vector was defined differently in this work. The dialogue context vector $\mathbf{c}$ considers the previous conversation history meanwhile the context vector only considers the user utterance or sentence in current time step. The dialogue acts are then fed in the decoder which represents the low-level policy and is implemented by an LSTM network. The decoder generates the low-level actions, which are a sequence of words $\mathbf{x}$ that make up a sentence for system reply. The model was first pre-trained to maximize the objective in terms of ELBO which is obtained by modifying (16) in a form of

$$L_{\text{hdno}}(\theta_h, \theta_l) = \mathbb{E}_{\mathbf{z}\sim\pi_{\theta_l}(\mathbf{z}|\mathbf{x})}\left[\sum_{t=1}^{T}\log\pi_{\theta_h}(w_t|\mathbf{z},\mathbf{c})\right] - \beta\mathrm{KL}(\pi_{\theta_l}(\mathbf{z}|\mathbf{x})\,\|p(\mathbf{z})) \tag{19}$$

where π_{θ_h} and π_{θ_l} are the high-level and low-level policies, respectively, w_t is the t^{th} generated token, T is the length of tokens in system response, and β is the regularization parameter. Next, RL was employed in the fine-tuning stage where the reward was defined as the success rate and the additional feedback from language model [14, 110] to evaluate the comprehensibility.

4.2 Transformer Model

On the other hand, the transformer model [89] is feasible to carry out the word-level dialogue policy optimization. The first word-level multi-domain dialogue policy that incorporated a transformer model is the hierarchical disentangled self-attention (HDSA) network [8, 10]. This network was designed to address the challenge of controlling neural response generation in multi-domain setting where the possible combinations of semantic inputs can grow exponentially compared to the single-domain tasks. To cope with this challenge, the HDSA network reformulated the structure of dialogue acts as a multi-layer hierarchical

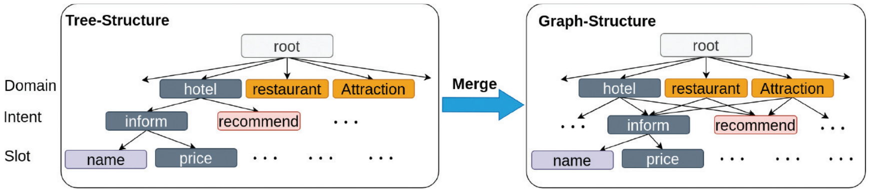


Figure 9: Structure of MultiWOZ dialogue acts in tree-based format (Left) and graph-based format (Right) where the input utterance corresponding to the dialogue act of domain, intent and slot is expressed by 'hotel'-'inform'-'price'. The grey nodes indicate the activated node.

graph by constructing a multi-layer tree that represents the entire dialogue act space based on their inter-relationships. The dialogue acts were interpreted as a root-to-leaf route on the graph, representing the hierarchical structure among domain, intent, and slot information. Then, the tree nodes with the same semantic meaning were merged to construct an acyclic multi-layer graph that simplified the representation of dialogue acts.

Figure 9 illustrates the representation of dialogue acts in a tree structure and the simplified version by using a graph structure, where different domains containing the same intents are connected to the same node. The multi-layer acyclic graph is then constructed as an inductive prior to feed into the HDSA network. The resulting HDSA architecture is depicted in Figure 10. Just like standard word-level policy, at the beginning, the dialogue context $\mathbf{c}$ is encoded to produce the sequence of feature representations $\{\bar{\mathbf{u}}, \mathbf{u}_1, \ldots, \mathbf{u}_m\}$. $\bar{\mathbf{u}}$ is a representation that captures the overall feature representation from different conversation turns in a context vector $\mathbf{c}$ as shown in the left. The context encoder can be built by using the convolutional neural network [55], LSTM [42] or transformer model [100]. Next, $\bar{\mathbf{u}}$ is concatenated with $\mathbf{bs}$ and $\mathbf{db}$ which in this work are defined as the one-hot vectors of belief state and DB pointer from dataset, respectively. Then, the dialogue act predictor is trained by minimizing the following loss as the negative log likelihood of dialogue act $\mathbf{A}$

$$\mathcal{L}\left(\theta_{\text{act}}\right) = -\log p_{\theta_{\text{act}}}\left(\mathbf{A} | \bar{\mathbf{u}}, \mathbf{bs}, \mathbf{db}\right) \tag{20}$$

where $\mathbf{A}$ is obtained by pre-processing the tree-structure of dialogue acts in the MultiWOZ dataset as the graph structure as shown in the Figure 9. θ_{act} denotes the parameter of dialogue act predictor. The ground-truth label in dialogue act prediction is obtained from the hierarchical graph structure, that can be represented either in multi-hot or one-hot vectors as illustrated in Figure 10. The graph of the predicted dialogue acts $\hat{\mathbf{A}}$ is then used to control the output of each disentangled self-attention (DSA) network (as shown in the middle) by activating its gating function or head G in accordance with the positive index (shown by orange) in the matrix $\hat{\mathbf{A}}$. The calculation of disentangled

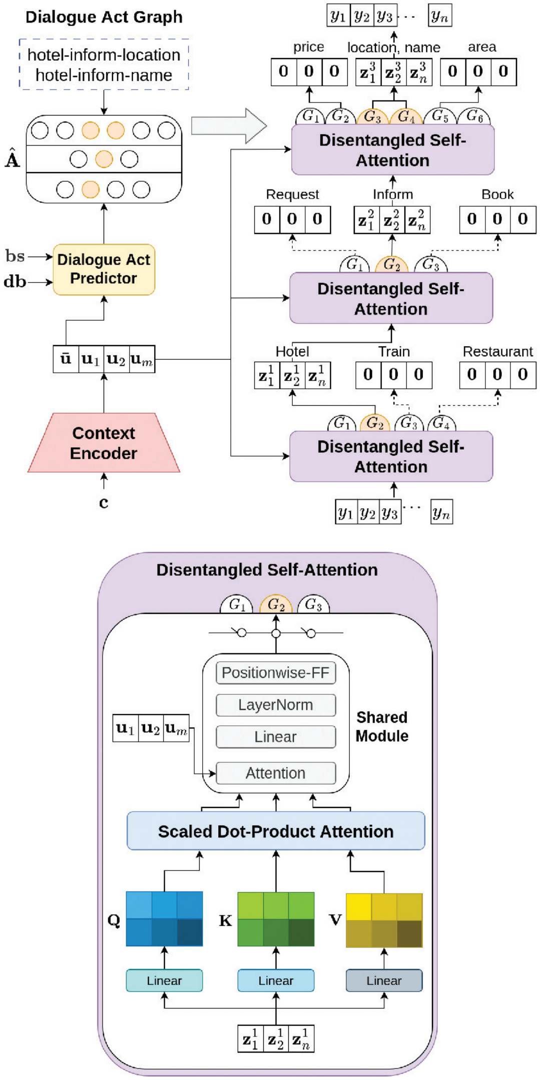

Dialogue Act Graph
hotel-inform-location
hotel-inform-name
$\hat{\mathbf{A}}$
bs
db
Dialogue Act Predictor
$\bar{\mathbf{u}}$ $\mathbf{u}_1$ $\mathbf{u}_2$ $\mathbf{u}_m$
Context Encoder
c
y_1 y_2 y_3 ··· y_n
price
location, name
area
0 0 0
$\mathbf{z}_1^3$ $\mathbf{z}_2^3$ $\mathbf{z}_n^3$
0 0 0
G_1 G_2 G_3 G_4 G_5 G_6
Disentangled Self-Attention
Request
Inform
Book
0 0 0
$\mathbf{z}_1^2$ $\mathbf{z}_2^2$ $\mathbf{z}_n^2$
0 0 0
G_1 G_2 G_3
Disentangled Self-Attention
Hotel
Train
Restaurant
$\mathbf{z}_1^1$ $\mathbf{z}_2^1$ $\mathbf{z}_n^1$
0 0 0
0 0 0
G_1 G_2 G_3 G_4
Disentangled Self-Attention
y_1 y_2 y_3 ··· y_n
Disentangled Self-Attention
G_1 G_2 G_3
Positionwise-FF
LayerNorm
Linear
Attention
Shared Module
$\mathbf{u}_1$ $\mathbf{u}_2$ $\mathbf{u}_m$
Scaled Dot-Product Attention
Q
K
V
Linear
Linear
Linear
$\mathbf{z}_1^1$ $\mathbf{z}_2^1$ $\mathbf{z}_n^1$

Figure 10: An illustration of dialogue act predictor in a hierarchical disentangled self-attention network (Up). The predicted dialogue act $\hat{\mathbf{A}} = [[0,1,0,0],[0,1,0],[0,0,1,1,0,0]]$ is formed as a concatenation over three vectors representing domain, intent and slot. $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ denote the query, key and value matrices in the self-attention layer (Down) of a transformer model, respectively.

self-attention is quite similar to that of standard self-attention. The clear difference is the calculation in the scaled dot-product attention as shown in the right. In standard attention, the output of each head is concatenated

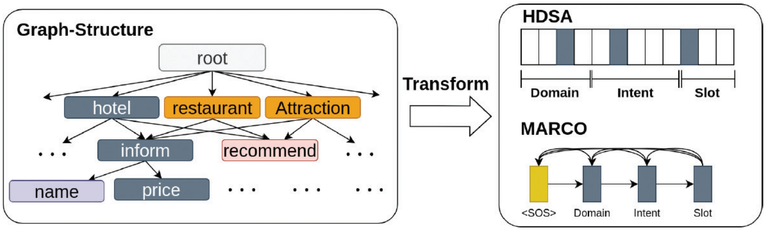


Figure 11: Dialogue act prediction based on HDSA and MARCO where the tasks of multi-label classification and sequence generation are implemented, respectively.

and then down projected to the desired matrix dimension. Meanwhile, in the disentangled self-attention, the output from each head is accumulated by following the activated gate before passing it to the next network. For example, in Figure 10, the set of latent variables $\{\mathbf{z}_1^3 \cdots \mathbf{z}_n^3\}$ are obtained by summing up the features from head 3 (G_3) and head 4 (G_4). The decoding process follows the standard transformer where the decoder input is taken from the encoder output of the last DSA layer or block, and the decoder parameter θ_{dsa} can be estimated by minimizing the negative logarithm of conditional likelihood of the target classes $\{y_1, \ldots, y_n\}$ which are sequence of tokens, given the encoded features $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$

$$\mathcal{L}\left(\theta_{\text{dsa}}\right) = -\log p_{\theta_{\text{dsa}}}(y_1, \ldots, y_n | \mathbf{u}_1, \ldots, \mathbf{u}_m, \mathbf{A}). \tag{21}$$

During inference time, $\hat{\mathbf{A}}$ is then used instead of $\mathbf{A}$. Overall, HDSA network contains the parameters of dialogue act predictor θ_{act} and disentangled self-attention θ_{dsa} which are estimated by minimizing the objectives (20) and (21), respectively.

The HDSA network was further extended to a new approach to dialogue act prediction, called the multi-domain dialogue acts and response co-generation (MARCO) [103]. This approach re-formulated the problem of dialogue act prediction through a sequential form of the vectorized graph representations. Therefore, the multi-label classification task in HDSA model is now changed to a kind of sequence generation task, as illustrated in Figure 11. The main motivation of this transformation is to provide a richer inter-relationship among different acts and generate a more flexible response since there may exist more than one dialogue act in a single turn. In the MARCO setting, the dialogue act prediction and the response generation are carried out concurrently. The response generation process is very close to the standard transformer decoder process, but MARCO introduced the dynamic act attention which allows cross attention between the predicted dialogue acts and the hidden states in the transformer decoder. The dialogue act predictor and the response generator shared an identical encoder which received an input from concatenation among

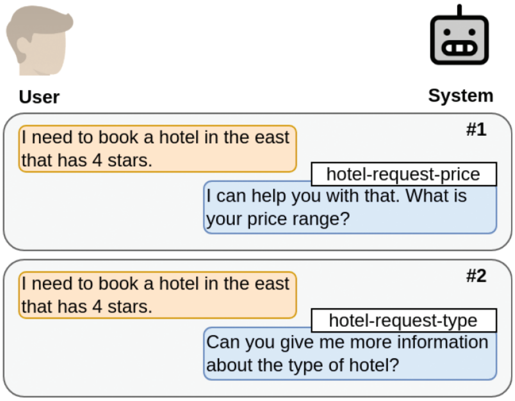

User
System
#1
I need to book a hotel in the east that has 4 stars.
hotel-request-price
I can help you with that. What is your price range?
#2
I need to book a hotel in the east that has 4 stars.
hotel-request-type
Can you give me more information about the type of hotel?

Figure 12: Example of one-to-many problem in multi-domain task-oriented dialogue.

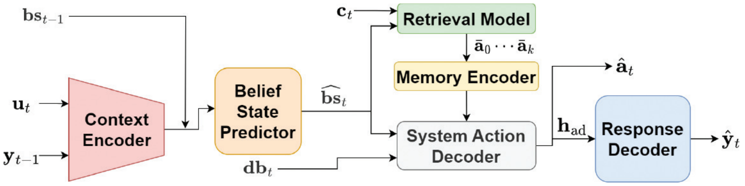


Figure 13: An overview of retrieve-and-memorize network.

dialogue context **c**, DB pointer (db) and belief state information (bs). All models were trained by minimizing the cross entropy loss similar to the learning objective in HDSA.

Because the multi-domain task-oriented dialogue (MDTD) system can be treated as a one-to-many mapping problem which means that one query may belong to many responses. Then, a specific solution to address this problem or phenomenon must be designed. An illustration of one-to-many mapping problem is depicted in Figure 12. A recent work proposed a retrieve-and-memorize framework to deal with this problem [57]. Figure 13 shows the model architecture of retrieve-and-memorize framework, named memory-augmented multi decoder (MAMD). The initial process is just similar to the other word-level dialogue policy learning which involves the context encoder. However, there exist some differences, e.g. the context encoder only considered the current user response $\mathbf{u}_t$ and the previous system response $\mathbf{y}_{t-1}$, and the new belief state predictor was introduced. The two main distinctive components included the context-aware retrieval module (or retrieval model) and the memory-augmented multi-decoder network (or memory encoder). The retrieval model was built by using the pre-trained language model, i.e. BERT [24], as an coder $f_{\mathrm{BERT}}(\cdot)$ which produces the sequence of hidden features in the BERT outputs

$$\mathbf{H} = f_{\text{BERT}}(e([\text{cls}]), e(\widehat{\mathbf{bs}}), e([\text{sep}]), e(\mathbf{c})) \tag{22}$$

where $e(\cdot)$ is an embedding function that transforms each token into the corresponding vector representation, $\widehat{\mathbf{bs}}$ is a sequence of tokens representing the predicted belief state, [sep] is a special token to separate different input sequences. From the BERT encoding, the model takes the class token $\mathbf{h}^{\text{cls}}$ from BERT outputs $\mathbf{H}$ and uses it to retrieve a set of top ranked candidate actions $\{\bar{\mathbf{a}}_0 \cdots \bar{\mathbf{a}}_k\}$ for system response. $\mathbf{h}^{\text{cls}}$ is a feature vector with a kind of class information from class token [cls] that captures the information of whole input sequence or equivalently represents the distributed representation of the dialogue context. The retrieval model in word-level policy model found a set of candidates of actions or system responses from dataset according to the ℓ_2-norm distance measure of the encoded feature vectors between $\mathbf{h}^{\text{cls}}$ of a dialogue context i and the other $\mathbf{h}_j^{\text{cls}}$ extracted from the other dialogue context j from the dataset

$$d\left(\mathbf{h}_i^{\text{cls}}, \mathbf{h}_j^{\text{cls}}\right) = \left\|\mathbf{h}_i^{\text{cls}} - \mathbf{h}_j^{\text{cls}}\right\|_2. \tag{23}$$

Here, k most similar candidate actions were chosen from the corresponding k most similar dialogue contexts following the result of ℓ_2-norm distance measure in (23). Next, the memory encoder encoded the concatenation of candidate actions $\{\bar{\mathbf{a}}_0 \cdots \bar{\mathbf{a}}_k\}$ retrieved from the retrieval model into a feature representation to facilitate the generation of dialogue acts. Finally, the prediction of system response decoder $\hat{\mathbf{y}}_t$ was generated with the condition on $\mathbf{h}_{\text{ad}}$ which is the last hidden state in the GRU-based system action decoder $\hat{\mathbf{a}}_t$. All individual components were trained jointly by minimizing the classification loss or cross-entropy loss of system output $\hat{\mathbf{y}}$ relative to the ground truth of the desired response $\mathbf{y}$.

In addition to employing the transformer encoder like BERT [24], recent studies have explored the utilization of transformer decoder, such as GPT-2 [76], to construct the word-level dialogue policy. The first approach was proposed in [3] where the belief state, database knowledge and user input were transformed into a simple text representation in token level. This GPT-2 model was fine-tuned by using standard autoregressive loss which maximized the likelihood over word sequence given by the ground-truth dialogue context. An illustration of this approach is depicted byin Figure 14 where this method directly estimates the word sequence without any prediction of system dialogue act. Further approaches have attempted to represent complete dialogue components consisting of NLU and NLG. Such an approach was used to find the word-level dialogue policy if the model was provided with user dialogue act and DST information from the other model or obtained from a dataset with the ground-truth information.

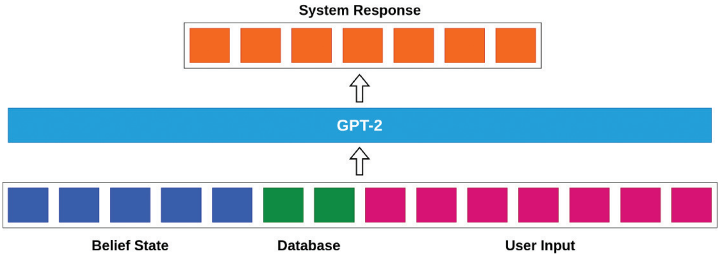


Figure 14: An illustration of word-level dialogue policy by using GPT-2 model.

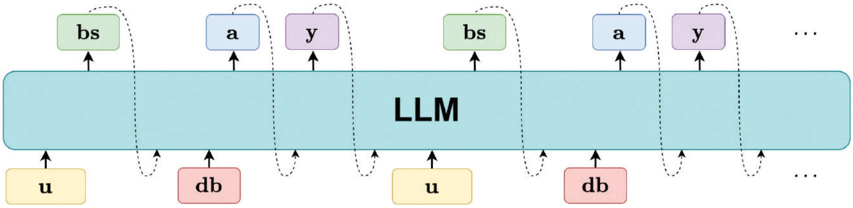


Figure 15: An illustration of end-to-end approach by using large language model (LLM).

In [37], an initial study applied GPT-2 model as a comprehensive neural model to build dialogue system. This model used the user utterance $\mathbf{u}$ to sequentially generate the distinct outputs derived from belief state $\mathbf{bs}$, system dialogue act $\mathbf{a}$ and system response $\mathbf{y}$, as illustrated in Figure 15. To distinguish various information within the token-level inputs, this work employed the sequence representation for specific tokens or markers including user utterance $\mathbf{u}$, dialogue act, system response, belief state and system action. Let $\mathbf{x} = [\mathbf{y}, \mathbf{a}, \mathbf{db}, \mathbf{bs}, \mathbf{c}]$ denote the inputs to GPT-2 model that concatenate the system response $\mathbf{y}$, system dialogue act $\mathbf{a}$, database search result $\mathbf{db}$, belief state $\mathbf{bs}$, and dialogue context $\mathbf{c}$ given an user input $\mathbf{u}$. Then, $p(\mathbf{x}) = p(\mathbf{y}, \mathbf{a}, \mathbf{db}, \mathbf{bs}, \mathbf{c})$ which can be factorized in an autoregressive manner by

$$\begin{aligned} p(\mathbf{x}) &= p(\mathbf{y}, \mathbf{a}, \mathbf{db}, \mathbf{bs}, \mathbf{c}) \\ &= p(\mathbf{y}|\mathbf{a}, \mathbf{db}, \mathbf{bs}, \mathbf{c}) \cdot p(\mathbf{a}|\mathbf{db}, \mathbf{bs}, \mathbf{c}) \\ &\quad \times p(\mathbf{db}|\mathbf{bs}, \mathbf{c}) \cdot p(\mathbf{bs}|\mathbf{c}) \cdot p(\mathbf{c}). \end{aligned} \tag{24}$$

Here, the probability $p(\mathbf{db}|\mathbf{bs}, \mathbf{c})$ is equal to 1 as the retrieved database result $\mathbf{db}$ is obtained from a deterministic pre-defined function conditioned on the predicted belief state $\mathbf{bs}$. Following the factorized autoregressive terms by taking negative logarithm of (24), three learning objectives are derived and optimized to find parameters θ. The first objective is a belief prediction loss which is expressed by

$$\mathcal{L}_b = -\log p(\mathbf{bs}|\mathbf{c}) = -\sum_{t=1}^{T_b} \log p_\theta(\mathbf{bs}_t|\mathbf{bs}_{<t}, \mathbf{c}) \tag{25}$$

where T_b is the length of token sequence in belief state $\mathbf{bs}$ and $\mathbf{bs}_{<t}$ means all tokens before t. The second learning objective is the prediction loss of system dialogue act which is yielded by

$$\mathcal{L}_a = -\log p(\mathbf{a}|\mathbf{db}, \mathbf{bs}, \mathbf{c}) = -\sum_{t=1}^{T_a} \log p_\theta(\mathbf{a}_t|\mathbf{a}_{<t}, \mathbf{db}, \mathbf{bs}, \mathbf{c}) \tag{26}$$

where T_a is the length of system dialogue act. The last objective is the response generation loss calculated by

$$\mathcal{L}_y = -\log p(\mathbf{y}|\mathbf{a}, \mathbf{db}, \mathbf{bs}, \mathbf{c}) = -\sum_{t=1}^{T_y} \log p_\theta(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{a}, \mathbf{db}, \mathbf{bs}, \mathbf{c}) \tag{27}$$

where T_y is the length of system response. Beside the minimization of (25)–(27), the prediction loss between positive example $\mathbf{x}$ and negative examples $\mathbf{x}'$ in a form of cross-entropy loss

$$\mathcal{L}_c = y \log p_\theta(\mathbf{x}) + (1-y)\log(1-p_\theta(\mathbf{x}')) \tag{28}$$

can be considered. Negative example is defined as a corrupted $\mathbf{x}$ in which y is replaced by a fake y in a binary classification.

In [71], a framework called SOLOIST was proposed to fulfill the pre-training stage by using the task-oriented dialogue dataset such as Taskmaster [5] and Schema [78]. SOLOIST further reduced the computation cost by removing $\mathbf{a}$ from $\mathbf{x}$ which means that this method did not calculate the dialogue act loss in (26). In addition, the negative sample $\mathbf{x}'$ in (28) was generated by replacing nearly 50% of elements in original sample $\mathbf{x}$ with different elements sampled from the dataset. The following work is called SimpleTOD [43] which was proposed by further removing the prediction loss in (28). All of these previous works obtained desirable performance. However, these works implemented the model training and evaluation only in the *dialogue turn* level. UBAR [109] is the first work attempted to fine-tune GPT-2 in the *dialogue session* level. The training scenario was basically identical with the SimpleTOD. The only difference was that the inputs in UBAR considered the previously generated belief state, dialogue act and system response.

The subsequent approach [52] introduced the data augmentation method to improve model generalization where the scheme of back-translation was applied. More recently, a method called GALAXY [39] was proposed by considering the pre-trained model with two additional datasets. The first dataset contains the unified dialogue act labels from several task-oriented dialogue datasets while the second dataset is an unlabeled dialogue dataset from several sources, e.g. online customer service logs. Initially, the model was pre-trained to estimate the system dialogue act by using the unified dataset. Next, the model was further pre-trained or fine-tuned by using the unlabeled dialogue dataset. The learning criterion was inspired by [30] in which different dropout rates of the same model should generate consistent outputs. The output consistency was calculated according to an objective based on KL divergence. After two pre-training stages were carried out, the model was fine-tuned by using MultiWOZ dataset with autoregressive loss using UniLM [25] as the backbone model.

Several latest approaches have attempted to utilize RL-based methods to train MDTD dialogue policy. As MultiWoZ contains several unsuccessful dialogues, GPT-critic [48] proposed a critic network that was used to generate new training data. The reward function was obtained from the external program like ConvLab-2 framework [118]. In the implementation, the optimum reward r_t^* was obtained from the external program, and the dialogue history $\mathbf{h}_t$ at time t consisted of $\mathbf{db}$, $\mathbf{bs}$ and $\mathbf{c}_t$. $\mathbf{h}_{t+1}^*$ at time $t+1$ was defined by replacing the original dialogue act $\mathbf{a}_t$ in $\mathbf{h}_{t+1}$ with the generated dialogue act $\mathbf{a}_t^*$. The new dataset $\mathcal{D}_{i+1}$ was obtained and updated at each new step $i+1$ according to the Q network with parameter ϕ by following

$$\mathcal{D}_{i+1} = \left\{ \left(g, \mathbf{h}_t, \mathbf{a}_t^*, r_t^*, \mathbf{h}_{t+1}^*\right) |\mathbf{a}_t^* = \underset{\substack{\mathbf{a}\in\{\mathbf{a}_i\}^N \\ \{\mathbf{a}_i\}^N \sim \pi_\theta^i(\mathbf{h}_t)}}{\arg\max} Q_\phi\left(\mathbf{h}_t, \mathbf{a}\right) \right\} \tag{29}$$

where $\mathbf{h}_t \in \mathcal{D}_i$ and the policy π_θ using GPT-2 model is adopted. This procedure generated N response candidates and only the response that was selected has the highest Q value. Different from the previous works that defined $\mathbf{a}$ as the system dialogue act, in this work, $\mathbf{a}_t$ was defined to represent both the system dialogue act and the corresponding system response. The critic network Q_ϕ was updated just like standard DQN. Next, the works [77] and [28] proposed a reward function for offline RL. The trajectories of the offline RL was determined by the dataset and the reward was used to evaluate the whole response generation directly, instead of evaluating in a word-by-word way. In [77], the pairwise reward learning was proposed to estimate the reward function to predict the trajectory with high rank score. The learned reward function was then used to fine-tune BART model [56] by using MultiWOZ dataset. In [28], two learnable reward functions were proposed. The first reward function was trained to evaluate or rank the quality of different trajectories which differed from the pairwise comparison. The second reward function was trained to directly predict the score for individual trajectories given in the MultiWOZ dataset. The score could be BLEU score, inform score, or combination of them. To evaluate the effectiveness of the learned reward functions, two different backbone models based on BART [56] and UniLM [25] were investigated.

5 Challenges and Difficulties

Despite the numerous methods that have been proposed to show the promising results in addressing the problems in multi-domain task-oriented dialogue (MDTD) system, there are still several issues that need to be tackled in the future. This section elaborates on these existing problems in detail. In particular, this paper points out the challenges in designing the simulated user and the standardized evaluation metric for multi-domain dialogue systems.

5.1 Simulated User Design for Reinforcement Learning

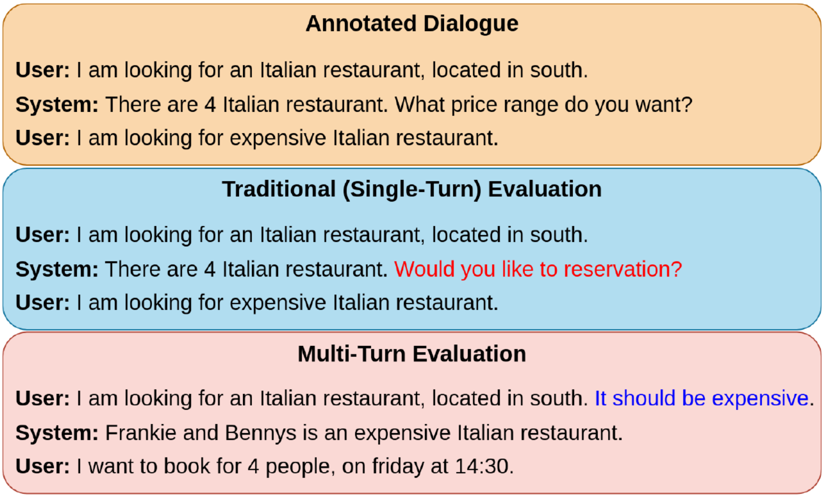
Creating an appropriate simulated user for training the dialogue policies with reinforcement learning (RL) stands as a significant challenge in MDTD system. As mentioned in the Introduction section, the standard approach is to use a rule-based policy with a specific dialogue agenda. However, some problems may emerge since designing such a rule-based policy requires domain expertise. Relying on the expert for designing the rule may restrict the diversity of the simulated user responses. Recent work has proposed different ways to enhance the diversity of the simulated user including the perturbation of language rules [69] or environment parameters [82] by additionally incorporating those data samples from multiple annotators or contributors [88], and possibly training multiple simulated users to provide the diverse state transitions in the RL environment. However, it is essential to control the level of diversity to prevent misguidance during the RL-based policy training. The previous work proposed an ensemble of the learnable simulated users with a frequency control to regulate the interaction with the dialogue agent [98]. The frequency control was defined to control the proportion of the stored trajectories obtained from the interaction between the expert simulated user and the ensembles of the learned simulated users. Such a scheme controlled the degree of diversity so that the stored trajectories do not diverge too far. Another study attempted to fine-tune the GPT-2 model to work as a simulated user for multi-domain task-oriented dialogue systems [59]. GPT-2 model does not only generate the user utterances and acts, but also constantly track the goal to evaluate the system response in every turn. While these solutions have demonstrated the improved performance, there is still considerable scope for further improvement by addressing this emerging challenge.

5.2 Multi-Domain Dialogue Policy Evaluation

Assessing the performance of the multi-domain task-oriented dialogue systems is crucial to measure their effectiveness in achieving their goals or objectives. Nonetheless, evaluating these systems is difficult because there are still no established standard evaluation metrics. The absence of standardized evaluation metrics has been thoroughly discussed in [97], where the distinction between the corpus-based (or equivalently single-turn) and the multi-turn evaluations as well as the distinction between the component-based and the end-to-end system evaluations are highlighted. Meanwhile, the study in [67] highlighted the inconsistencies in the MultiWOZ benchmark due to the non-existence of standard data pre-processing and evaluation script which resulted in unfair performance comparisons. This paper eventually suggests adding further realistic evaluation by considering the multi-turn evaluation and even the human-level subjective evaluation in addition to the corpus-based metrics. Basically, long-term evaluation of a dialogue with multiple turns is considerably important when compared with the short-term or corpus-based evaluation which treats the evaluation of individual turns independently.

In addition, the standard evaluation in MultiWOZ dataset utilizes a corpus-based evaluation method that can be either component-based or end-to-end system evaluation. In the end-to-end system evaluation, the evaluated model is provided with the ground-truth user input. In the component-based evaluation setting, in addition to receiving ground-truth data from the previous conversation turns, the evaluated models also obtain the ground-truth data from the upstream modules. For example, in the dialogue policy evaluation, the evaluated models are provided with the ground-truth data on belief state, database query and dialogue context. Therefore, the end-to-end approaches that have been trained to optimize all dialogue components from NLU to NLG can be evaluated in the same way as the word-level dialogue policy evaluation by giving the ground-truth of previous utterances, belief state and database pointer as given in the entities retrieved from the MultiWOZ database. The recent end-to-end approaches include three different methods which are abbreviated as DAMD [115], UBAR [109] and Galaxy [39].

Unfortunately, the assumption that a multi-domain task-oriented dialogue system can provide correct responses at all times does not hold true in real-world scenarios. This is because such systems involve a sequence of the related inquiries and responses between the user and the system, which can produce incorrect responses at any point in the sequence [9]. These errors are likely accumulated and will considerably affect the performance in subsequent turns. An illustration is shown in Figure 16. In traditional evaluation in a form of single-turn evaluation, the user response just follows the annotated data without considering the output from the previous system response. To address this issue, ConvLab-2 accordingly introduced the multi-turn and end-to-end



Annotated Dialogue

User: I am looking for an Italian restaurant, located in south.

System: There are 4 Italian restaurant. What price range do you want?

User: I am looking for expensive Italian restaurant.

Traditional (Single-Turn) Evaluation

User: I am looking for an Italian restaurant, located in south.

System: There are 4 Italian restaurant. Would you like to reservation?

User: I am looking for expensive Italian restaurant.

Multi-Turn Evaluation

User: I am looking for an Italian restaurant, located in south. It should be expensive.

System: Frankie and Bennys is an expensive Italian restaurant.

User: I want to book for 4 people, on friday at 14:30.

Figure 16: An illustration of different evaluation settings in the multi-domain task-oriented dialogue system. Red colored text represents a mismatch between the system response of a trained system and the corresponding annotated response. Blue colored text shows the difference between the sentence generated by agenda-based policy and the annotated dialogue data.

system evaluation. This involves an agenda-based user simulator that generates the sentences following the goals in the MultiWOZ dataset. The generated sentences may be different from the sentences in the annotated data because the user responses should be adaptive to the previous system responses, as shown in Figure 16. In this evaluation, each component of the dialogue system must produce correct output, such as the NLU predicting the correct user dialogue acts and the dialogue policy generating the appropriate system dialogue acts. Otherwise, the dialogue system will respond incorrectly and lead to low scores in the evaluation metrics. As shown in Table 8, while all word-level dialogue act policies using different models performed well in single-turn dialogue evaluation, their performance significantly declined when evaluated in the multi-turn setting. In addition, compared with the DA-level dialogue policy, the word-level dialogue policy that was based on LLM model did not exhibit improvement in the multi-turn evaluation. Such a performance comparison is illustrated in Table 9 where DA-level dialogue policy generally performs better than word-level dialogue policy. In this comparison, different methods for DA-level policy learning are consistently implemented by using NLU-DST and NLG based on BERT and template NLG, respectively.

In contrast to single-turn evaluation, which focuses primarily on the metrics such as inform, success rate, and BLEU score, the multi-turn dialogue evaluation includes additional metrics such as the average turn (averaged number of

Table 8: The distinctions of evaluating the dialogue performance between single-turn interactions and multi-turn interactions.

Model	Single-Turn Evaluation			Multi-Turn Evaluation			
	Inform	Success Rate	BLEU	Inform	Success Rate	Complete Rate	Average Turn
[116]	82.8	79.2	12.8	48.0	47.7	68.9	13.1
[8]	82.9	68.9	23.6	50.0	34.3	39.2	15.9
[61]	96.4	83.6	14.0	54.5	40.0	49.2	26.7
[109]	87.5	74.4	17.6	79.8	74.3	79.8	14.2
[52]	83.5	67.3	17.2	70.3	60.1	89.4	12.7
[48]	90.1	76.6	17.8	79.0	77.7	84.3	16.3

Table 9: Performance comparison between DA-level and word-level dialogue policies in the end-to-end system with multi-turn evaluation.

Model	Multi-Turn Evaluation			
	Inform	Success Rate	Complete Rate	Average Turn
DA-level dialogue policy				
[96]	69.0	54.1	68.3	10.9
[44]	78.9	66.0	77.0	13.9
[22]	89.5	81.7	89.1	14.0
[81]	84.0	82.5	87.4	12.5
Word-level dialogue policy				
[116]	48.0	47.7	68.9	13.1
[8]	50.0	34.3	39.2	15.9
[61]	54.5	40.0	49.2	26.7
[109]	79.8	74.3	79.8	14.2
[52]	70.3	60.1	89.4	12.7
[48]	79.0	77.7	84.3	16.3

turns) required to fulfill user goals, and the completion rate, which measures the proportion of user constraints that are met during a dialogue session. For single-turn and multi-turn evaluations, both inform and success rate are examined. Inform is the F1 score to measure if all requested information has been informed. Success rate is measured by judging whether the constraints and requests in the user goals have been satisfied by system. BLEU score of the generated sentences (the higher the better) is not considered in the multi-turn evaluation since the generated conversations between the simulated user and the dialogue system are generally different compared to the annotated data, but still controlled by the user goals defined by MultiWOZ dataset. However, it's worth noting that these metrics are only available in the MultiWOZ 2.1

version, which features the annotated user dialogue acts defined heuristically by the ConvLab-2 developer. Consequently, comparing different methods that use different versions of the MultiWOZ dataset in multi-turn and end-to-end system evaluations may present a challenge in the study on multi-domain task-oriented dialogue system. Based on the aforementioned findings, this paper highly suggests that the standardized dialogue policy evaluation in the MDTD task should include both automatic and human evaluation.

6 Conclusions

This paper has presented a survey that explores recent advances in optimizing multi-domain task-oriented dialogue policies on the MultiWOZ dataset. The survey provides a detailed explanation of the differences in input-output definition and in optimization strategy between two common approaches including dialogue act-level (DA-level) and word-level dialogue policy. Among the DA-level approaches, the reinforcement learning-based optimization is the most preferred technique. This approach defines the input as an observation in the POMDP setting, which contains various information features, such as the belief state information, the last system response, and the current user response. The output of the DA-level policy is the system's dialogue acts. On the other hand, among the word-level policies, the most common setting is to use a recurrent neural network-based or transformer-based model, which is then optimized by using the classification loss. The input consists of the belief state, database pointer, and dialogue context from the dataset, and the output of the word-level policy is the sequence of words that represent the system responses. Finally, this paper points out the challenges and difficulties that need to be further addressed in the future, including the simulated user design and the standardized multi-domain dialogue evaluation.

Acknowledgement

This work was supported in part by the National Science and Technology Council, Taiwan, under Contract NSTC 111-2634-F-A49-012.

Biographies

Mahdin Rohmatillah received his M.S. degree from National Sun Yat-sen Univesity, Taiwan in 2018, in electrical engineering. He is now pursuing Ph.D. degree in National Yang Ming Chiao Tung University. His research interests include machine learning and dialogue system.

Jen-Tzung Chien is the Lifetime Chair Professor in the electrical and computer engineering, and the computer science at National Yang Ming Chiao Tung University, Taiwan. He served as the tutorial speaker for top conferences including AAAI, IJCAI, ACL, KDD, MM, ICASSP, ICME, CIKM, IJCNN, COLING and Interspeech. Dr. Chien has published extensively, including three books and 250 peer-reviewed articles, many on machine learning, deep learning and Bayesian learning with applications on natural language processing and computer vision.

References

[1] D. Abel, J. Salvatier, A. Stuhlmüller, and O. Evans, "Agent-Agnostic Human-in-the-Loop Reinforcement Learning," in *Proc. of NIPS Workshop on the Future of Interactive Learning Machines*, 2016.

[2] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, and P. Fung, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," *arXiv preprint arXiv: 2302.04023*, 2023.

[3] P. Budzianowski and I. Vulić, "Hello, It's GPT-2 - How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems," in *Proc. of Workshop on Neural Generation and Translation*, 2019, 15–22.

[4] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, "MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2018, 5016–26.

[5] B. Byrne, K. Krishnamoorthi, C. Sankar, A. Neelakantan, B. Goodrich, D. Duckworth, S. Yavuz, A. Dubey, K.-Y. Kim, and A. Cedilnik, "Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2019, 4516–25.

[6] J. D. Chang, M. Uehara, D. Sreenivas, R. Kidambi, and W. Sun, "Mitigating Covariate Shift in Imitation Learning via Offline Data with Partial Coverage," in *Advances in Neural Information Processing Systems*, 2021.

[7] M.-Y. Chen, M. Rohmatillah, C.-H. Lee, and J.-T. Chien, "Meta Learning for Domain Agnostic Soft Prompt," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[8] W. Chen, J. Chen, P. Qin, X. Yan, and W. Y. Wang, "Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2019, 3696–709.

[9] Q. Cheng, L. Li, G. Quan, F. Gao, X. Mou, and X. Qiu, "Is MultiWOZ a Solved Task? An Interactive TOD Evaluation Framework with User Simulator," *arXiv preprint arXiv:2210.14529*, 2022.

[10] J. T. Chien and Y. H. Huang, "Bayesian Transformer Using Disentangled Mask Attention," in *Proc. of Annual Conference of the International Speech Communication Association*, Vol. 2022, 2022, 1761–5.

[11] J.-T. Chien and Y.-H. Chen, "Learning Continuous-Time Dynamics With Attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2), 2023, 1906–18.

[12] J.-T. Chien and Y.-C. Chiu, "Bayesian Multi-Temporal-Difference Learning," *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022, 1–31.

[13] J.-T. Chien and P.-C. Hsu, "Stochastic Curiosity Exploration for Dialogue Systems," in *Proc. of Annual Conference of International Speech Communication Association*, 2020, 3885–9.

[14] J.-T. Chien and Y.-C. Ku, "Bayesian Recurrent Neural Network for Language Modeling," *IEEE Transactions on Neural Networks and Learning Systems*, 27(2), 2015, 361–74.

[15] J.-T. Chien and W. X. Lieow, "Meta Learning for Hyperparameter Optimization in Dialogue System," in *Proc. of Annual Conference of International Speech Communication Association*, 2019, 839–43.

[16] J.-T. Chien and T.-C. Luo, "Flow-Based Variational Sequence Autoencoder," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2022, 1418–25.

[17] J.-T. Chien and S.-H. Yang, "Model-Based Soft Actor-Critic," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2021, 2028–35.

[18] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," in *Proc. of Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, 103–11.

[19] C.-T. Chu, M. Rohmatillah, C.-H. Lee, and J.-T. Chien, "Augmentation Strategy Optimization for Language Understanding," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, 7952–6.

[20] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A Recurrent Latent Variable Model for Sequential Data," in *Advances in Neural Information Processing Systems*, 2015.

[21] T. Cordier, T. Urvoy, F. Lefèvre, and L. M. R. Barahona, "Few-Shot Structured Policy Learning for Multi-Domain and Multi-Task Dialogues," in *Findings of the Association for Computational Linguistics: EACL*, 2023, 432–41.

[22] T. Cordier, T. Urvoy, F. Lefèvre, and L. M. Rojas Barahona, "Graph Neural Network Policies and Imitation Learning for Multi-Domain Task-Oriented Dialogues," in *Proc. of the Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2022.

[23] Y. Dai, H. Yu, Y. Jiang, C. Tang, Y. Li, and J. Sun, "A Survey on Dialog Management: Recent Advances and Challenges," *arXiv preprint arXiv:2005.02233*, 2020.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, 4171–86.

[25] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified Language Model Pre-training for Natural Language Understanding and Generation," in *Advances in Neural Information Processing Systems*, 2019.

[26] L. El Asri, J. He, and K. Suleman, "A Sequence-to-Sequence Model for User Simulation in Spoken Dialogue Systems," *Proc. of Annual Conference of the International Speech Communication Association*, 2016, 1151–5.

[27] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu, "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures," in *Proc. of the International Conference on Machine Learning*, 2018, 1407–16.

[28] Y. Feng, S. Yang, S. Zhang, J. Zhang, C. Xiong, M. Zhou, and H. Wang, "Fantastic Rewards and How to Tame Them: A Case Study on Reward Learning for Task-oriented Dialogue Systems," in *Proc. of International Conference on Learning Representations*, 2023.

[29] J. Fu, K. Luo, and S. Levine, "Learning Robust Rewards with Adverserial Inverse Reinforcement Learning," in *Proc. of International Conference on Learning Representations*, 2018.

[30] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2021, 6894–910.

[31] C. Geishauser, C. van Niekerk, H.-c. Lin, N. Lubis, M. Heck, S. Feng, and M. Gašić, "Dynamic Dialogue Policy for Continual Reinforcement Learning," in *Proc. of International Conference on Computational Linguistics*, 2022, 266–84.

[32] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proc. International Conference on Neural Information Processing Systems*, 2014, 2672–80.

[33] G. Gordon-Hall, P. Gorinski, and S. B. Cohen, "Learning Dialog Policies from Weak Demonstrations," in *Pro. of Annual Meeting of the Association for Computational Linguistics*, 2020, 1394–405.

[34] G. Gordon-Hall, P. J. Gorinski, G. Lampouras, and I. Iacobacci, "Show Us the Way: Learning to Manage Dialog from Demonstrations," *arXiv preprint arXiv:2004.08114*, 2020.

[35] I. Gür, D. Hakkani-Tür, G. Tür, and P. Shah, "User Modeling for Task Oriented Dialogues," in *Proc. of IEEE Spoken Language Technology Workshop*, IEEE, 2018, 900–6.

[36] P. de Haan, D. Jayaraman, and S. Levine, "Causal Confusion in Imitation Learning," in *Advances in Neural Information Processing Systems*, 2019.

[37] D. Ham, J.-G. Lee, Y. Jang, and K.-E. Kim, "End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 583–92.

[38] T. He, J. Zhang, Z. Zhou, and J. Glass, "Exposure Bias versus Self-Recovery: Are Distortions Really Incremental for Autoregressive Text Generation?" In *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2021, 5087–102.

[39] W. He, Y. Dai, Y. Zheng, Y. Wu, Z. Cao, D. Liu, P. Jiang, M. Yang, F. Huang, L. Si, *et al.*, "Galaxy: A Generative Pre-Trained Model for Task-Oriented Dialog with Semi-Supervised Learning and Explicit Policy Injection," in *Proc. of AAAI Conference on Artificial Intelligence*, Vol. 36, No. 10, 2022, 10749–57.

[40] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep Q-Learning from Demonstrations," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2018, 3223–30.

[41] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning," *Advances in Neural Information Processing Systems*, 2016.

[42] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 9(8), 1997, 1735–80.

[43] E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher, "A Simple Language Model for Task-Oriented Dialogue," in *Advances in Neural Information Processing Systems*, Vol. 33, 2020, 20179–91.

[44] C.-E. Hsu, M. Rohmatillah, and J.-T. Chien, "Multitask Generative Adversarial Imitation Learning for Multi-Domain Dialogue System," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, 2021, 954–61.

[45] X. Huang, J. Qi, Y. Sun, and R. Zhang, "Semi-Supervised Dialogue Policy Learning via Stochastic Reward Estimation," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 660–70.

[46] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. of AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1, 2018.

[47] J.-W. Jang, M. Rohmatillah, and J.-T. Chien, "AVAST: Attentive Variational State Tracker in a Reinforced Navigator," in *Proc. of International Joint Conference on Natural Language Processing*, 2022, 424–33.

[48] Y. Jang, J. Lee, and K.-E. Kim, "GPT-critic: Offline Reinforcement Learning for End-to-End Task-Oriented Dialogue System," in *Proc. of International Conference on Learning Representations*, 2022.

[49] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to Trust Your Model: Model-Based Policy Optimization," in *Neural Information Processing Systems*, 2019.

[50] S. Keizer, M. Gašić, F. Jurčíček, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Parameter Estimation for Agenda-Based User Simulation," in *Proc. of SIGDIAL Conference*, 2010, 116–23.

[51] S. Kim, S. Yang, G. Kim, and S.-W. Lee, "Efficient Dialogue State Tracking by Selectively Overwriting Memory," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 567–82.

[52] J. Kulhánek, V. Hudeček, T. Nekvinda, and O. Dušek, "AuGPT: Dialogue with Pre-trained Language Models and Data Augmentation," *arXiv preprint arXiv:2102.05126*, 2021.

[53] W.-C. Kwan, H.-R. Wang, H.-M. Wang, and K.-F. Wong, "A Survey on Recent Advances and Challenges in Reinforcement Learning Methods for Task-oriented Dialogue Policy Learning," *Machine Intelligence Research*, 20, 2023, 318–34.

[54] J. Langford and T. Zhang, "The Epoch-Greedy Algorithm for Contextual Multi-Armed Bandits," *Advances in Neural Information Processing Systems*, 2007.

[55] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, 86(11), 1998, 2278–324.

[56] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Com-

prehension," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 7871–80.

[57] Y. Li, Y. Yang, X. Quan, and J. Yu, "Retrieve & Memorize: Dialog Policy Learning with Multi-Action Memory," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, 2021, 447–59.

[58] Z. Li, S. Lee, B. Peng, J. Li, J. Kiseleva, M. de Rijke, S. Shayandeh, and J. Gao, "Guided Dialogue Policy Learning without Adversarial Learning in the Loop," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2020, 2308–17.

[59] H. Liu, Y. Cai, Z. Ou, Y. Huang, and J. Feng, "A Generative User Simulator with GPT-based Architecture and Goal State Tracking for Reinforced Multi-Domain Dialog Systems," *arXiv preprint arXiv:2210.08692*, 2022.

[60] P. Lu, T. Bai, and P. Langlais, "SC-LSTM: Learning Task-Specific Representations in Multi-Task Learning for Sequence Labeling," in *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, 2396–406.

[61] N. Lubis, C. Geishauser, M. Heck, H.-c. Lin, M. Moresi, C. van Niekerk, and M. Gasic, "LAVA: Latent Action Spaces via Variational Autoencoding for Dialogue Policy Optimization," in *Proc. of International Conference on Computational Linguistics*, 2020, 465–79.

[62] T.-C. Luo and J.-T. Chien, "Variational Dialogue Generation with Normalizing Flows," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, 7778–82.

[63] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" In *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2022, 11048–64.

[64] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *Proc. of The International Conference on Machine Learning*, Vol. 48, 2016, 1928–37.

[65] 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(7540), 2015, 529–33.

[66] O. Nachum, S. Gu, H. Lee, and S. Levine, "Data-Efficient Hierarchical Reinforcement Learning," in *Proc. of International Conference on Neural Information Processing Systems*, 2018, 3307–17.

[67] T. Nekvinda and O. Dušek, "Shades of BLEU, Flavours of Success: The Case of MultiWOZ," in *Proc. of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics*, 2021, 34–46.

[68] A. Y. Ng and S. Russell, "Algorithms for Inverse Reinforcement Learning," in *Proc. of International Conference on Machine Learning*, 2000.

[69] T. Niu and M. Bansal, "Automatically Learning Data Augmentation Policies for Dialogue Tasks," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2019, 1317–23.

[70] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Gray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training Language Models to Follow Instructions with Human Feedback," in *Advances in Neural Information Processing Systems*, 2022.

[71] B. Peng, C. Li, J. Li, S. Shayandeh, L. Liden, and J. Gao, "SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching," *Transactions of the Association for Computational Linguistics*, 9, 2021, 807–24.

[72] B. Peng, X. Li, J. Gao, J. Liu, and K.-F. Wong, "Deep Dyna-Q: Integrating Planning for Task-Completion Dialogue Policy Learning," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2018, 2182–92.

[73] B. Peng, C. Zhu, C. Li, X. Li, J. Li, M. Zeng, and J. Gao, "Few-shot Natural Language Generation for Task-Oriented Dialog," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, 172–82.

[74] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network," *Advances in Neural Information Processing Systems*, 1, 1988.

[75] J. Qiu, H. Zhang, and Y. Yang, "Reward Estimation with Scheduled Knowledge Distillation for Dialogue Policy Learning," *Connection Science*, 35(1), 2023, 2174078.

[76] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI Blog*, 1(8), 2019, 9.

[77] G. S. Ramachandran, K. Hashimoto, and C. Xiong, "Causal-Aware Safe Policy Improvement for Task-Oriented Dialogue," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2022, 92–102.

[78] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, "Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset," in *Proc. of AAAI Conference on Artificial Intelligence*, Vol. 34, No. 5, 2020, 8689–96.

[79] M. Rohmatillah and J.-T. Chien, "Causal Confusion Reduction for Robust Multi-Domain Dialogue Policy," in *Proc. of Annual Conference of the International Speech Communication Association*, 2021, 3221–5.

[80] M. Rohmatillah and J.-T. Chien, "Corrective Guidance and Learning for Dialogue Management," in *Proc. of ACM International Conference on Information & Knowledge Management*, 2021, 1548–57.

[81] M. Rohmatillah and J.-T. Chien, "Hierarchical Reinforcement Learning With Guidance for Multi-Domain Dialogue Policy," *IEEE Transactions on Audio, Speech, and Language Processing*, 31, 2023, 748–61.

[82] N. Ruiz, S. Schulter, and M. Chandraker, "Learning To Simulate," in *Proc. of International Conference on Learning Representations*, 2019.

[83] D. E. Rumelhart and J. L. McClelland, "Learning Internal Representations by Error Propagation," in, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, 1987, 318–62.

[84] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, "Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System," in *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics*, 2007.

[85] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," in *Proc. of International Conference on Learning Representations*, 2016.

[86] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[87] P. Shah, D. Hakkani-Tur, B. Liu, and G. Tür, "Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning," in *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics*, 2018, 41–51.

[88] K. Stasaski, G. H. Yang, and M. A. Hearst, "More Diverse Dialogue Datasets via Diversity-Informed Data Collection," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 4958–68.

[89] Y. Su, K. Fan, N. Bach, C.-C. J. Kuo, and F. Huang, "Unsupervised Multi-Modal Neural Machine Translation," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 10474–83.

[90] Y. Su, Y. Huang, and C.-C. J. Kuo, "Dependent bidirectional RNN with extended-long short-term memory," 2018.

[91] Y. Su and C.-C. J. Kuo, "On Extended Long Short-Term Memory and Dependent Bidirectional Recurrent Neural Network," *Neurocomputing*, 356, 2019, 151–61.

[92] Y. Su and C.-C. J. Kuo, "Recurrent Neural Networks and Their Memory Behavior: A Survey," *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022, DOI: 10.1561/116.00000123.

[93] R. S. Sutton, D. Precup, and S. Singh, “Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning,” *Artificial Intelligence*, 112(1-2), 1999, 181–211.

[94] R. S. Sutton, “Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming,” in *Machine Learning Proceedings*, 1990, 216–24.

[95] R. Takanobu, R. Liang, and M. Huang, “Multi-Agent Task-Oriented Dialog Policy Learning with Role-Aware Reward Decomposition,” in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 625–38.

[96] R. Takanobu, H. Zhu, and M. Huang, “Guided Dialog Policy Learning: Reward Estimation for Multi-Domain Task-Oriented Dialog,” in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2019, 100–10.

[97] R. Takanobu, Q. Zhu, J. Li, B. Peng, J. Gao, and M. Huang, “Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation,” in *Proc. of Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2020, 297–310.

[98] Z. Tang, H. Kulkarni, and G. H. Yang, “High-Quality Dialogue Diversification by Intermittent Short Extension Ensembles,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, 2021, 1861–72.

[99] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A Physics Engine for Model-Based Control,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, 5026–33.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, 2017.

[101] H. Wang and K.-F. Wong, “A Collaborative Multi-Agent Reinforcement Learning Framework for Dialog Action Decomposition,” in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2021, 7882–9.

[102] J. Wang, Y. Zhang, T.-K. Kim, and Y. Gu, “Modelling Hierarchical Structure between Dialogue Policy and Natural Language Generator with Option Framework for Task-oriented Dialogue System,” in *Proc. of International Conference on Learning Representations*, 2021.

[103] K. Wang, J. Tian, R. Wang, X. Quan, and J. Yu, “Multi-Domain Dialogue Acts and Response Co-Generation,” in *Proc. of Annual Meeting of the Association for Computational Linguistics*, 2020, 7125–34.

[104] Z. Wang, T.-H. Wen, P.-H. Su, and Y. Stylianou, “Learning Domain-Independent Dialogue Policies via Ontology Parameterisation,” in *Proc. of Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015.

[105] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, "Sample Efficient Actor-Critic with Experience Replay.," in *Proc. of International Conference on Learning Representations*, 2017.

[106] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[107] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, 8, 1992, 279–92.

[108] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, "An Explanation of In-context Learning as Implicit Bayesian Inference," in *Proc. of International Conference on Learning Representations*, 2022.

[109] Y. Yang, Y. Li, and X. Quan, "UBAR: Towards Fully End-to-End Task-Oriented Dialog System with GPT-2," in *Proc. of AAAI Conference on Artificial Intelligence*, Vol. 35, No. 16, 2021, 14230–8.

[110] Z. Yang, Z. Hu, C. Dyer, E. P. Xing, and T. Berg-Kirkpatrick, "Unsupervised Text Style Transfer Using Language Models as Discriminators," *Advances in Neural Information Processing Systems*, 2018.

[111] T. Yu, R. Zhang, H. Er, S. Li, E. Xue, B. Pang, *et al.*, "CoSQL: A Conversational Text-to-SQL Challenge Towards Cross-Domain Natural Language Interfaces to Databases," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2019, 1962–79.

[112] J. Zhang, K. Hashimoto, C.-S. Wu, Y. Wang, P. Yu, R. Socher, and C. Xiong, "Find or Classify? Dual Strategy for Slot-Value Predictions on Multi-Domain Dialog State Tracking," in *Proc. of Joint Conference on Lexical and Computational Semantics*, 2020, 154–67.

[113] S. Zhang, J. Zhao, P. Wang, Y. Li, Y. Huang, and J. Feng, "“Think Before You Speak”: Improving Multi-Action Dialog Policy by Planning Single-Action Dialogs," in *Proc. of International Joint Conference on Artificial Intelligence*, 2022, 4510–6.

[114] X. Zhang, B. Peng, K. Li, J. Zhou, and H. Meng, "SGP-TOD: Building Task Bots Effortlessly via Schema-Guided LLM Prompting," *arXiv preprint arXiv:2305.09067*, 2023.

[115] Y. Zhang, Z. Ou, and Z. Yu, "Task-Oriented Dialog Systems That Consider Multiple Appropriate Responses under the Same Context," in *Proc. of AAAI Conference on Artificial Intelligence*, Vol. 34, No. 5, 2020, 9604–11.

[116] T. Zhao, K. Xie, and M. Eskenazi, "Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models," in *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, 1208–18.

[117] Q. Zhu, K. Huang, Z. Zhang, X. Zhu, and M. Huang, "CrossWOZ: A Large-Scale Chinese Cross-Domain Task-Oriented Dialogue Dataset," *Transactions of the Association for Computational Linguistics*, 8, 2020, 281–95.

[118] Q. Zhu, Z. Zhang, Y. Fang, X. Li, R. Takanobu, J. Li, B. Peng, J. Gao, X. Zhu, and M. Huang, “ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems,” in *Proc. of Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020, 142–9.