Co-Evolutional User Simulator and Dialogue System with Bias Estimator

Anonymous ACL submission

Abstract

Reinforcement learning (RL) has emerged as a promising approach to fine-tune offline pretrained GPT-2 model in task-oriented dialogue systems. In order to obtain human-like online interactions while extending the usage of RL, building pretrained user simulators (US) along with dialogue systems (DS) and facilitating jointly fine-tuning via RL becomes prevalent. However, existing methods usually asynchronously update US and DS to ameliorate the ensued non-stationarity problem, which could bring a lot of manual operations, lead to suboptimal policy and less sample efficiency. The paradigm of iterative training implicitly dress the distributional shift problem caused by compounding exposure bias. To take a step further for tackling the problem, we introduce an **C**o-**E**volutional framework of **T**ask-**O**riented **D**ialogue (**CETOD**) with bias estimator, which enables bias-aware synchronously update for RL-based fine-tuning whilst takes advantages from GPT-2 based end-to-end modeling on US and DS. Extensive experiments demonstrate that CETOD achieves state-of-the-art success rate, inform rate and combined score on MultiWOZ2.1 dataset.

1 Introduction

Traditionally, task-oriented dialogue (TOD) systems are trained via pipeline approaches by decomposing the task into multiple independent modules (Wen et al., 2017; Chen et al., 2020). Recently, recasting the TOD as a unified language modeling task with leveraging supervised pretrained language model like GPT-2 (Radford et al., 2019) becomes prevailing, which thoroughly avoids the cross-module error accumulation problem in the pipeline approach. However, GPT-2 suffers from exposure bias (He et al., 2019; Zhang et al., 2020a; Arora et al., 2022) problem that the model has never been exclusively exposed to its own predictions during training thus leads to accumulated errors in the

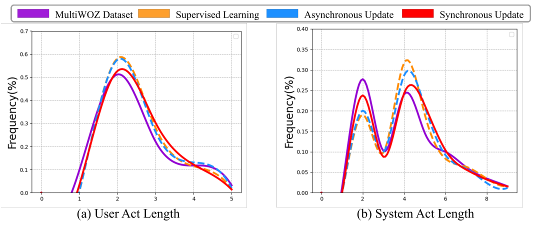


Figure 1: The length of user act and system act.

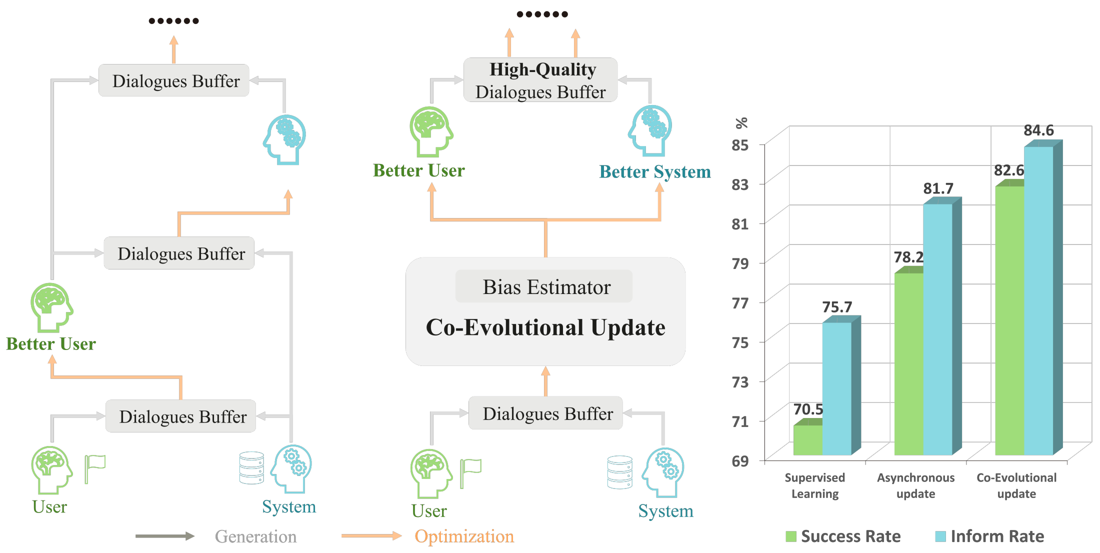
KL divergence(%)	User Act Length	System Act Length
Supervised Learning(SL)	17.48	2.08
Asynchronous Update	17.0	2.23
Co-Evolutional Update	**4.27**	**0.58**

Table 1: Comparison of KL divergence on length of user act, system act between different training methods and MultiWOZ2.1 dataset.

output generation process during test. To avoid such problem, leveraging reinforcement learning (RL) could be one of the antidotes (Keneshloo et al., 2020) because the optimization direct relies on its own outputs with rewards (e.g., success rate) as update guidance rather than the ground-truths.

RL requires large amounts of online interactions for training. However, interacting with human users is time-consuming and costly. An intuitive way for establishing communications with an RL-based dialogue system (DS) is training a GPT-2 based user simulator (US) which learns from real data to mimic human behavior (Shi et al., 2019). However, serving as each other's environment to interact with, joint update makes both US and DS learning under non-stationarity conditions (Liu and Lane, 2017), Existing methods usually employ asynchronous update (Fig. 2(a)) which update US first and then update DS to ameliorate this issue.

Joint update brings compounding exposure bias problem which is the deviation due to self-carrying bias and unseen input distribution from the environment in the process of online interactions. Comparing the act length of US and DS in Fig. 1, the distributional shift problem caused by it can be



(a) Asynchronous update (b) Co-Evolutional update (c) Online evaluation

Figure 2: (a) Asynchronous update usually iteratively update US first and then update DS, while (b) Co-Evolutional update use the same batch of data to synchronously update US and DS. The online evaluation results (c) show that our update method is superior to asynchronously update regarding dialogue success rate and inform rate.

inferred (Dataset VS. SL), it also can be mathematically calculated the KL divergence between their distributions in Table 1. Unfortunately, this asynchronous update paradigm feels challenging to continually adapt to changes in distribution shift, the gap between data distribution is still wide (SL VS. asynchronous update), and it ameliorates the problem by sacrificing sample efficiency and might lead to sub-optimal policy, a lot of manual operations are also introduced.

In order to take a step further for tackling the distributional shift problem (SL VS. co-evolutional update), we propose a co-evolutional framework, which enables bias-aware synchronous update for RL-based fine-tuning with hierarchical reward and policy optimization combinations through the same batch of online data (Fig. 2(b)) whilst takes advantages from GPT-2 based end-to-end modeling on US and DS. We also propose bias estimator to deal with the non-stationary problem, which performs on both US and DS by taking uncertainty of transitions (Yu et al., 2020) into consideration to address the problem of distribution shift by trading off the risk of making mistakes and the benefit of diverse exploration. With such a complete mechanism, we build high-quality loops for policy learning and online data collection as shown in Fig. 2(b). Our contributions can be summarized as follows:

- We propose a novel bias-aware co-evolutional update framework for US and DS policy fine-tuning while ameliorating the distributional shift problem with the rewards that been explored from both hierarchical granularity and dialogue sub-task optimization combinations.
- CETOD provides end-to-end modeling on US and DS based on GPT-2 with the full ability to understand, make decisions, generate language, and enable naturally joint update with engaging the components of bias estimator.
- Extensive experiments demonstrate that CETOD outperforms SOTA methods on MultiWOZ2.1 and has achieved 79.0 success rate, 87.5 inform rate and 101.5 combined score.

2 Related Work

Pretrained language model for US and DS. The approaches of solving TOD have been transformed from traditional pipeline methods (Zhong et al., 2018; Zhang et al., 2019a; Chen et al., 2019) to end-to-end manner (Madotto et al., 2018; Lei et al., 2018; Zhang et al., 2020b; Zhao et al., 2022). With the development of pretrained language models such as GPT-2, GPT-based methods become dominant in TOD, e.g., SimpleTOD (Hosseini-Asl et al., 2020), SOLOIST (Peng et al., 2020), AuGPT (Kulhánek et al., 2021), UBAR (Yang et al., 2021). The literature of US modeling can be roughly summarized into two types: one is rule-based simulation such as the agenda-based user simulator (Li et al., 2016; Shah et al., 2018a), easy to apply but very limited under complex scenarios; the other is data-driven US modeling, (Eshky et al., 2012; Asri et al., 2016; Kreyssig et al., 2018; Shi et al., 2019; Shah et al., 2018a; Zhang et al., 2019b), which is more robust but requires large amounts of manual annotations and system-corresponding data. The most widely used benchmark dataset MultiWOZ (Budzianowski et al., 2018b) have about 8000 dialogues. Smaller datasets such as DSTC2 (Henderson et al., 2014) and M2M (Shah et al., 2018b) contain 1600 and 1500 dialogues respectively. In this work, CETOD leverages GPT-2 for end-to-end modeling of US and DS with MultiWOZ2.1 dataset.

Reinforcement Learning methods in TOD. Reinforcement learning aims to learn optimal policy to maximize long-term cumulative rewards. With different data collecting paradigm for policy update, (Sutton and Barto, 1998) divides RL into online RL and offline RL. Apply offline RL in TOD can avoid explicit construction of US and directly learn from offline dataset (Zhou et al., 2017; Lin et al., 2021; Jeon and Lee, 2022). However, offline RL struggles with a major challenge (Kumar et al., 2020) that it may fail due to overestimation of values caused by distribution shift between dataset and learning policies. Online RL (Gur et al., 2018; Tseng et al., 2021) needs to design a US to interact with DS (acting as their opponent's environment) and generate dialogues data which can be further used for policy optimization. To improve the sample efficiency of deep RL, (Wu et al., 2020) apply model-based RL which incorporates a model-based critic for the TOD system. CETOD builds the framework of US and DS through offline supervised learning (SL) to online RL. The offline stage focuses on building US and DS that communicate using natural language, whereas the online stage optimizes dialogue policy using the generated high-quality data.

Joint update of US and DS. The joint optimization scheme for end-to-end US and DS is the most relevant research direction of our work. (Takanobu et al., 2020) follows the idea of multi-agent reinforcement learning, which treats DS and US as two dialogue agents and utilizes role-aware reward decomposition in joint optimization. (Papangelis et al., 2019) learn both US and DS, but only applied in the single-domain dataset (DSTC2). In addition, most of them are based on traditional network architectures LSTM (Liu and Lane, 2017; Tseng et al., 2021), (Anonymous, 2022) firstly build a GPT-2 based trainable US. And in the way of joint update implementation, they (Liu and Lane, 2017; Anonymous, 2022) employ asynchronous update to weaken non-stationarity problem, which chooses to fix the system and update user first, and update system after obtaining a better user (Fig. 2(a)). CETOD is a co-evolutional fine-tuning framework (Fig. 2(b)) to tackle the distributional shift problem, which ameliorates the compounding exposure bias while ensuring stationarity.

3 Offline Supervised Learning for User Simulator and Dialogue System

To enable our online co-evolutional update framework, we first build DS and US via SL on the MultiWOZ2.1 dataset to establish communications via natural language between them.

3.1 Architecture Design

To simulate the entire dialogue process and information flow in real world, the end-to-end architecture of US and DS is designed as shown in Fig. 3(b). During the training phase, a pretrained language model such as GPT-2 is tuned to produce a conditional generative model. The whole input sequence c_t as described below: for US, the natural language sequential pairs $\{sr, uu\}_{1:t-1}$ of system response sr_t and user utterance uu_t is concatenated with the user's understanding un_t of dialogue history, dynamic goal state g_t, user act ua_t, and current user utterance uu_t, i.e.,

$$c_t^{\text{US}} = \{sr, uu\}_{1:t-1} \oplus un_t \oplus g_t \oplus ua_t \oplus uu_t \quad (1)$$

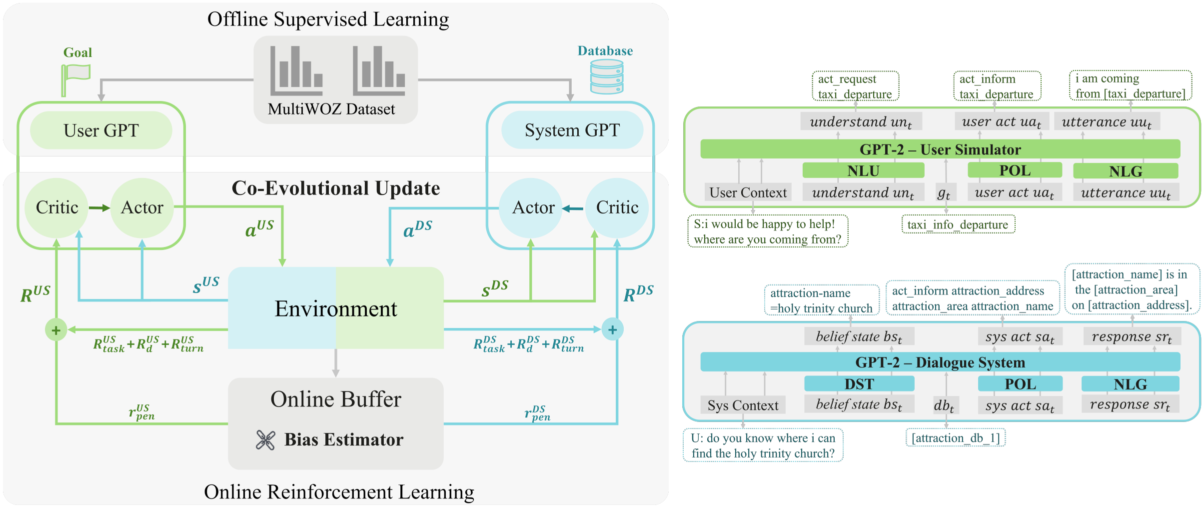
where $\oplus$ serves as the operation of concatenation, specific details are shown in Fig. 3(b). The natural language sequential pairs $\{uu, sr\}_{1:t-1}$ is highly symmetric for DS and is concatenated with the belief state bs_t, database query result db_t, system act sa_t and current system response sr_t, i.e.,

$$c_t^{\text{DS}} = \{uu, sr\}_{1:t-1} \oplus bs_t \oplus db_t \oplus sa_t \oplus sr_t \quad (2)$$

3.2 Offline Supervised Learning

The training objective of offline supervised learning is the language modeling conditional likelihood objective (Bengio et al., 2000) as shown in Eq. 3:

$$L_{\text{SL}}^{\#} = \sum_{i}^{|C|} \log P(c_i^{\#} | c_{<i}^{\#}) \quad (3)$$



(a) Overall view of framework: CETOD.

(b) Architecture of US and DS.

Figure 3: (a) The overall view of our framework CETOD. We first obtain US and DS through offline SL and then use online RL and co-evolutional update with bias estimator to further optimize dialogue policies. (b) The architecture of our end-to-end (NLU or DST, POL, and NLG) US and DS.

where # denote US or DS, and $|\cdot|$ is the length of sequence, which maximizes the probability of the next word prediction, and it is the same for US and DS. In the online interactive phase, the US generates under the condition of a completed goal and history, while the DS is conditioned on the external database and history. First, they generate an understanding un_t or bs_t of the content based on previous context history. Then the goal state g_t and db_t are added to form a new sequence, lastly producing their corresponding actions ua_t or sa_t and delexicalized responses sr_t or uu_t.

4 Online Reinforcement Learning for User Simulator and Dialogue System

With US and DS obtained from offline learning as policy initialization, co-evolutional update is performed with hierarchical reward, policy optimization combinations and bias estimator. We present how online RL works in the following section.

4.1 Co-Evolutional Update

In TOD tasks, US tries to fully express the entire goal and responds to DS, while DS searches for entities that meet the requirements and replies in accordance with the request of US, finally they complete the dialogue goal successfully; it is essential to joint update which improves coordination and synchronization between US and DS.

In our framework CETOD shown in Fig. 3(a), it is crucial to accelerate online RL using offline learned policies of US π_θ^{US} and DS π_θ^{DS}. However, DS and US tend to express their own perspectives and generate poor quality dialogue data under the existing asynchronous update paradigm due to distribution shift; detailed examples are illustrated in Appendix B. CETOD improves their dialogue policies by synchronous update, which uses the same batch of data generated by the interaction between US and DS every epoch to concurrently optimize dialogue policy.

We apply PPO2 (Schulman et al., 2017) in our online RL framework, which has the advantage of trust region policy optimization (TRPO (Schulman et al., 2015)), and it is easier to implement, more generic, and empirically has better sample complexity. The objective proposed is the following:

$$L_\pi(\theta^\#) = \hat{\mathbb{E}}_t\Big[\frac{\pi_{\theta^\#}(a_t|s_t)}{\pi_{\theta^\#_{old}}(a_t|s_t)}\hat{A}_t, \\ \text{clip}\Big(\frac{\pi_{\theta^\#}(a_t|s_t)}{\pi_{\theta^\#_{old}}(a_t|s_t)}, 1-\epsilon, 1-\epsilon\Big)\hat{A}_t\Big)\Big] \tag{4}$$

where # denote US or DS, θ is the parameter of the policy network, s_t, a_t is the state and action in the markov decision process (MDP), which are token by token for GPT's input and output of our CETOD, the state is represented by the context of previous dialogue turns, the action is the response generated by the model each turn, and their space is composed of the generated tokens in an orderly manner, ϵ is a hyper-parameter, $\hat{A}_t$ is advantage function, the specific calculation formula can refer to PPO2 (Schulman et al., 2017). In order to fully exploit the performance of GPT-2 without generating redundant parameter models, we treat GPT-2 itself as the actor network for policy learning. To approximate the value function, we connect a small linear network to the hidden layers of GPT-2 as the critic network, which is aimed at minimizing:

$$L_V(\phi^{\#}) = (V_{\phi^{\#}}(s_t) - V_{\#}^{\text{target}})^2 \tag{5}$$

denote US or DS, where $V_{\phi^{\#}}$ is the value function, and ϕ is the parameter of the value network. According to the visualization (Fig. 1) of data distribution results, co-evolutional update can effectively ameliorate the compounding exposure bias between US and DS, thus preventing policy from falling into the sub-optimal range. Online interaction evaluation in Sec. 5 also demonstrates that it improves the sample efficiency compared to asynchronous update.

4.2 Reward Assignment

Reinforcement learning methods help to solve the inconsistency between train/test measurements in pretrained language models. However, it becomes difficult for policy learning when RL algorithms take place in an environment where rewards are sparse, so we explore the hierarchical dense reward with different levels of granularity and divide the reward into different levels:

Task Reward R_{task}: the success of the online dialogue is used as the Task Reward R_{task}, which can only be observed at the end of the conversation, and are shared for US and DS. R_{task} serves as the most important motivational signal to facilitate policy learning and performance improvement.

Domain Reward R_d: the success for a domain is defined as Domain Reward R_d, which is also shared for US and DS. In the dialogue of multiple domains, R_d assists in smoothing the process of policy learning at the node of domain conversion.

Turn Reward $R_{\text{turn}}^{\#}$: is designed separately for US and DS, and it can be observed at every turn.

1) US Turn Reward $R_{\text{turn}}^{\text{US}}$ concludes: it provides a new inform about the slot; it asks about a new attribute about an entity; and it correctly replies to the request from the DS side.

2) DS Turn Reward $R_{\text{turn}}^{\text{DS}}$ involves: it requests a new slot; it successfully provides the entity; and it correctly answers all attributes from the US side.

4.3 Policy Combinations

Previous studies learned to reinforced DS mainly focused on optimizing dialogue policy modules, using system acts for performing actions. An accepted idea is that dialog policy, which decides the next action that the dialog agent should take, plays a vital role in a TOD system. In our co-evolutional update framework, we explore different policy optimization combinations, which include executing action A_t, understanding context U_t, and generating natural language G_t.

4.4 Bias Estimator

We also propose a penalty reward based on the uncertainty of our learned transitions. Referring to the penalty reward of uncertainty in MOPO (Yu et al., 2020), $r_{\text{pen}}^{\#}$ is related to the probability of the generated output token in GPT-2:

$$r_{\text{pen}}^{\#} = \lambda\left(1 - \frac{\sum Num(prob > prob^{\star})}{\sum Num}\right) \tag{6}$$

λ and $prob^{\star}$ are two hyperparameters, $prob^{\star}$ is the artificially set threshold, *Num* represents the number of eligible tokens. In general, the bias estimator is used for dealing with untrusted data. We use the penalty reward mechanisms to guide policy learning and ensure that the data it produces does not end up in untrusted regions. Experimental results in Table 4 indicate that bias estimator are important to state-of-the-art performance.

Intuitively, with the co-evolutional update, greater dialogue success rates can be achieved while improving sample efficiency. As a result, co-evolutional update forms high-quality cycles for policy learning and data collection.

The experimental results show that all the different types of rewards plays an essential role in performance improvement. In summary, the composition of our global reward $R^{\#}$ is as follows:

$$R^{\#} = R_{\text{task}} + R_d + R_{\text{turn}}^{\#} + r_{pen}^{\#} \tag{7}$$

During the start stage of online fine-tuning, distribution shift may result in severe bootstrap errors. To ensure the purity of our dialogue date in online buffer and continued training during the RL phase, we apply a structural bias estimator to pick out fatal dialogues that impact the optimization process.

5 Experiments

Dataset. We perform all experiments using MultiWOZ2.1 (Eric et al., 2020), which is currently

Model	**Pretrained Model**	**RL-based**	**Inform Rate**	**Success Rate**	**BLEU**	**Combined Score**
SimpleTOD(Hosseini-Asl et al., 2020)	DistilGPT2	w/o	84.4	70.1	15.0	92.3
AuGPT(Kulhánek et al., 2021)	variantGPT-2	w/o	76.6	60.5	16.8	85.4
SOLOIST(Peng et al., 2020)	GPT-2	w/o	82.3	72.4	13.6	90.9
UBAR(Yang et al., 2021)	DistilGPT2	w/o	83.4	70.3	17.6	94.4
PPTOD(Su et al., 2022)	T5models	w/o	83.1	72.7	18.2	96.1
BORT(Sun et al., 2022)	T5-small	w/o	85.5	77.4	17.9	99.4
MTTOD(Lee, 2021)	T5-base	w/o	85.9	76.5	19.0	100.2
GALAXY(He et al., 2021)	UniLM	w/o	85.4	75.7	**19.64**	100.2
MTTOD(Lee, 2021)	T5-base	w/o	85.9	76.5	19.0	100.2
JOUST(Tseng et al., 2021)	LSTM	w	83.2	73.5	17.6	96.0
SGA-JRUD(Anonymous, 2022)	DistilGPT-2	w	85.0	74.0	19.11	98.61
CETOD-DS(Ours)	DistilGPT2	w	**87.5**	**79.0**	18.25	**101.5**

Table 2: Empirical comparison of End-to-End TOD systems models in the official leaderboard. CETOD achieve the state-of-the-art results of success rate, inform rate and the combined score.

still widely being used in TOD, and the results published on the official leaderboard are all using MultiWOZ2.0/2.1. It is a large-scale multi-domain Wizard of Oz dataset for TOD. There are 3406 single-domain conversations that include booking if the domain allows for that and 7032 multi-domain conversations consisting of at least 2 to 5 domains. Each dialogue consists of a goal, multiple user utterances, and system responses. Also, each turn contains a belief state and a set of dialogue actions with slots for each turn. TOD system is usually defined by an ontology, which defines all entity properties called slots and all possible slot values. Details can be found in the appendix E. The user's understanding works as a reception of DS's output messages, and it's not available in MultiWOZ, we use labeled file according to JOUST, which is open sourced.

Evaluation Metrics. Three automatic metrics are included to ensure better interpretation of the results. Among them, the first two metrics evaluate the completion of dialogue tasks: whether the system has provided an appropriate entity (*Inform rate*) and then answered all the requested attributes (*Success rate*); while fluency is measured via *BLEU* score (Papineni et al., 2002). Following (Mehri et al., 2019) ,the *Combined Score* performance (Combined) is also reported, calculated as (0.5*(Inform + Success) + BLEU). The overall goal in TOD domain is getting a strong DS, which is achieved by fair offline evaluation compared to other methods(such as JOUST, SGA-JRUD etc. on the leaderboard). Online evaluation is used to measure the respective method's performance in the joint update process.

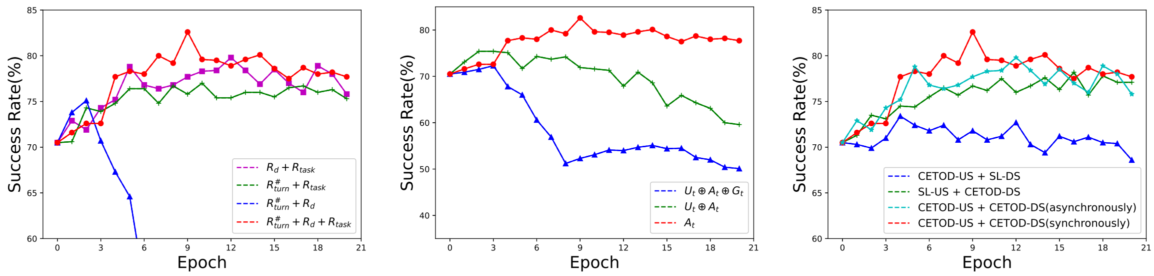
Training Procedure. First, we train US and DS with offline supervision on the MultiWOZ2.1 (Eric et al., 2020) dataset, defined as SL-US and SL-DS. We implement our framework with HuggingFace's Transformers (Wolf et al., 2019) of DistilGPT2 (Sanh et al., 2019), a distilled version of GPT-2. Then we collect online interactive data through the communication between SL-US and SL-DS for later RL experiments with the objective Eq. 4 and Eq. 5, and the constructed goal is sampled from the train or dev dataset. Thus we get two co-evolutional update models defined as CETOD-US and CETOD-DS. More details about the experiments and hyper-parameters can be found in Appendix A.

Diversity	**SL-US**	**CETOD-US**	**SL-DS**	**CETOD-DS**
distinct-1(‰)↑	5.961	**6.249**	4.872	**5.125**
distinct-2(‰)↑	31.848	**32.098**	26.549	**27.617**
Self-BLEU(%)↓	24.722	**21.025**	27.008	**22.161**

Table 3: Results of diversity matrix distinct.

Offline Benchmark Evaluation. We first show the offline benchmark results of different supervised-trained DS in an end-to-end manner in Table 2. All the contents we use are ground truth from the US side; it mainly evaluates the ability of DS. The scripts [1] we strictly followed are released by Paweł Budzianowski from Cambridge Dialogue Systems Group (Budzianowski et al., 2018a; Ramadan et al., 2018; Eric et al., 2020; Zang et al., 2020). Those end-to-end pretrained model-based methods use the dialogue history as input to generate the belief states, actions, and responses simultaneously. Regardless of the type of pretrained model and whether the RL methods are used, the overall goal in TOD domain is getting a strong DS, CETOD achieves state-of-the-art results: success rate of 79.0, inform rate of 87.5, and combined

[1]The evaluation code is released at `https://github.com/budzianowski/multiwoz`.



(a) Reward Settings (b) Policy Scheme (c) Update Pattern

Figure 4: Comparative analysis of different combinations of rewards settings, policy schemes and update patterns.

Model	Online Evaluation		Offline Evaluation			
	Inform	Success	Inform	Success	BLEU	Combined
JOUST(Tseng et al., 2021)	**84.6**	73.0	83.2	73.5	17.6	96.0
CETOD-w/o R_{task}	79.9	75.1	82	74.9	18.23	96.68
CETOD-w/o R_d	82.4	76.7	86.6	77.4	17.55	99.55
CETOD-w/o $R^{\#}_{\text{turn}}$	83.2	79.8	86.5	77.2	17.64	99.49
CETOD-[POL = $U_t \oplus A_t \oplus G_t$]	77.5	72.3	83.9	76.5	16.67	98.87
CETOD-[POL = $U_t \oplus A_t$]	80	75.4	84.6	76.5	**18.71**	99.26
CETOD-[SL-US + SL-DS]	75.7	70.5	70.5	69.8	18.1	91.95
CETOD-[CETOD-US + SL-DS]	78.8	73.4	70.5	69.8	18.1	91.95
CETOD-[SL-US + CETOD-DS]	81.7	78.2	85.2	77.4	17.98	99.28
CETOD-[Asynchronous Update]	82	78.6	85.9	77.2	17.51	99.06
CETOD-w/o R_{pen}	84	80.6	85.5	78	17.8	99.55
CETOD [POL = A_t], w R_{pen} w R_{task} R_d $R^{\#}_{\text{turn}}$ (Ours)	**84.6**	**82.6**	**87.5**	**79.0**	18.25	**101.5**

Table 4: Empirical comparison of interaction quality of generated dialogues using the 1k test corpus user goals.

score of 101.5 points.

Online Interactive Evaluation. In order to verify the effectiveness of our online RL optimization, we let US and DS interact with each other. In this process, the US can only receive the information from the goal and system response, and DS feeds back the entities through the database according to user utterance; there is no ground truth in the process of online interactive dialogues. In addition to DS, this evaluation also indicates the capabilities of the US. Note that we do not show the BLEU score since there is no reference available in online interactions. Some existing methods are not compared here because of the inconsistent evaluation methods (the reason why SGA-JRUD has better performance under online evaluation is that they used different and uncommonly used evaluation scripts (Shi et al., 2019)). The experimental results are shown in Table 4 and Fig. 4.

Under the same test method, the success rate of CETOD is significantly better than JOUST (Tseng et al., 2021), which verifies that our CETOD achieves the purpose of an efficient loop of

Percentage(%)	SL-US + SL-DS	CETOD-US + CETOD-DS
Success	36.0	**64.0**
US Humanoid	40.0	**60.0**
DS Quality	43.0	**57.0**
Fluency	38.0	**62.0**

Table 5: Results of human evaluation.

data collection and policy learning. Table 3 shows the results of distinct-k, which measures the degree of diversity by calculating the number of distinct uni-grams and bi-grams in generated responses. It can be seen that the text generated with our RL optimization is of higher diversity, and A lower Self-BLEU (Zhu et al., 2018) score also implies more diversity of the document.

Human Evaluation. Human evaluation of dialogue quality is performed on the Amazon Mechanical Turk platform to confirm the improvement of our proposed method CETOD. It is to verify that method has improved from SL to RL. We randomly sample 100 dialogues by US and DS, and each dialogue is evaluated by five turkers. Four evaluation indicators involve: **1) Success**: Which interactive dialogue completes the goal of the task more successfully? **2) US Humanoid**: Which US behaves more like a real human user and whether the US expresses the constraints completely in an organized way? **3) DS Quality**: Which DS behaves more intelligently and provides US with the required information? **4) Fluency**: Which dialogue is more natural, fluent, and efficient?

The results of the human evaluation shown in Table 5 are consistent with the results of the online evaluation. DS is more efficient at completing dialogues with our proposed online RL optimization. Furthermore, joint optimization of US can produce behavior more closely resembling that of a human. Improvements under two agents produce a more natural and efficient dialogue flow.

6 Ablation Study

Hierarchical Dense Rewards. A major challenge of putting RL into practice is the sparsity of reward feedback (Rengarajan et al., 2022). As described in Sec. 4.1, we specially design fine-grained dialogue turn reward $R_{\text{turn}}^{\#}$, domain reward R_d and overall task reward R_{task} according to the characteristics of US and DS in TOD. The evaluation results are shown in the second row of Table 4. In Fig. 4(a), we plot the online interaction success rate curve, which is based on different reward settings during online RL optimization.

As we can see from the result, the three types of designed dense rewards all have final positive effects on the success of the task. It is worth noticing that R_{task} plays a major role. The success rate will dramatically drop if there is no R_{task}. R_d and $R_{\text{turn}}^{\#}$ both improve the performance of online and offline evaluation, which indicates the importance of our dense reward for realizing optimal performance.

Choice of RL Policy Scheme. In RL, the policy represents a probabilistic mapping from states to actions. CETOD's framework contains not only reinforced end-to-end DS, but also reinforced the end-to-end US, and their policies include executing action A_t, understanding context U_t, and generating natural language G_t.

We conduct three experiments and their RL policies are $U_t \oplus A_t \oplus G_t$, $U_t \oplus A_t$ and A_t respectively. Based on different policy schemes during online RL optimization, the success rate curves are shown in Fig. 4(b). The best performance results are obtained when only the dialogue policy is optimized, while adding the optimization of the component of understanding and generation does not enhance the success rate. It can be seen from Table 4 that using A_t for policy achieves the highest online evaluation results with large margins. In offline evaluation, using A_t also achieves the best results. The reason is that the quality of the policy directly influences the quality of the dialogue, and the generation module generally has an excellent performance in SL. In the case of three modules being optimized simultaneously, the training of the online RL process becomes more trembling and the guidance of reward becomes oblique and falls into sub-optimal.

Validity of Co-Evolutional update. The third row of Table 4 demonstrates the effectiveness of co-evolutional update. When we use RL to optimize only US or DS, the performance drops significantly compared with the co-evolutional update. In particular, when we only update the US, the performance improvement is even smaller. We also compare the performance between synchronous and asynchronous update in our CETOD framework, asynchronous update is lower than ONCE but comparable to SGA-JRUD, especially the success rate and inform rate, which shows that co-evolutional update is efficient and better. The main reason is that it helps US and DS coordinate with each other and effectively solve the problem of distribution shift. As shown in Fig. 4(c), the online interaction success rate curve based on different reinforced agents during online RL optimization also verifies the conclusion.

Validity of Bias Estimator. The fourth row of Table 4 demonstrates the effectiveness of our bias estimator. Concretely, the penalty reward help CETOD maximizes a lower bound of the return in the true MDP, careful use of the model in regions outside of the data support, and find the optimal trade-off between the return and the risk (Yu et al., 2020).

7 Conclusion and Discussion

Our contribution is that we propose a bias-aware concurrent joint update framework compared to existing RL-based TOD systems, bias estimator are modules that make the online RL process more stable and improve the final performance. Compared with the asynchronous update, synchronous joint update greatly reduces the proportion of manual operations, and optimizes it as an automated process, when terminating the optimization of US or DS is not easy and difficult to balance in asynchronous update. It performs offline SL on dataset to learn GPT-2-based end-to-end US and DS, both of which possess features of natural language understanding, dialogue policy management, and natural language generation. Finally, we achieved the current state-of-the-art results.

As for future work, CETOD will be applied to more complex dialogues tasks and other scenarios. Although CETOD currently achieves state-of-the-art results, its performance may still be limited by the pretrained language model and online reinforcement learning algorithms, so it will be interesting to explore stronger neural network models or robust RL algorithms. Last but not least, another research direction is to create the US with a variety of personalities to support DS policy learning.

Limitations

Throughout the perspective of distributional visualizations, the problem of distribution shift caused by compounding exposure bias and non-stationarity still persists. However, we have made claims about our desire to take a step further to address it, which can be proved from our experimental results and the gap of distribution between ours and the original dataset is shrunk. Thus we can focus on more effective methods in the future and provide a theoretical basis for solving this problem.

Meanwhile, due to a large amount of parameters of the GPT model, it is difficult and time-consuming to train the two GPT-based US and DS in the online RL process. At the same time, according to the conclusion of optimizing the GPT with different granularity of policy schemes. In future work, we can consider optimizing only parts of parameters of GPT itself to achieve better performance and improve the efficiency of RL algorithms and computing resources.

Ethics Statement

Our method and implementation are based on the existing public dataset MultiWOZ (Eric et al., 2020), without any personal identity and subjective feelings. While our approach has no negative effects on society, we also hope to contribute to the development of task-oriented dialogue. At the same time, we also pay attractive salaries to the turkers of Amazon Mechanical Turk; in addition to thanking them for their assistance in human evaluation, we also want to encourage more scholars to participate and offer part-time job opportunities.

References

Anonymous. 2022. Jointly reinforced user simulator and task-oriented dialog system with simplified generative architecture.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 700–710. Association for Computational Linguistics.

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 1151–1155. ISCA.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938. MIT Press.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018a. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7521–7528. AAAI Press.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3696–3709. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 422–428. European Language Resources Association.

Aciel Eshky, Ben Allison, and Mark Steedman. 2012. Generative goal-driven user simulation for dialog management. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 71–81. ACL.

Izzeddin Gur, Dilek Hakkani-Tür, Gökhan Tür, and Pararth Shah. 2018. User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, pages 900–906. IEEE.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James R. Glass. 2019. Quantifying exposure bias for neural language generation. *CoRR*, abs/1905.10617.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, Jian Sun, and Yongbin Li. 2021. GALAXY: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. *CoRR*, abs/2111.14592.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 263–272. The Association for Computer Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

Hyunmin Jeon and Gary Geunbae Lee. 2022. DORA: towards policy optimization for task-oriented dialogue system with efficient context. *Comput. Speech Lang.*, 72:101310.

Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K. Reddy. 2020. Deep reinforcement learning for sequence-to-sequence models. *IEEE Trans. Neural Networks Learn. Syst.*, 31(7):2469–2489.

Florian Kreyssig, Iñigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, pages 60–69. Association for Computational Linguistics.

Jonás Kulhánek, Vojtech Hudecek, Tomás Nekvinda, and Ondrej Dusek. 2021. Augpt: Dialogue with pre-trained language models and data augmentation. *CoRR*, abs/2102.05126.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with A simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1296–1303. Association for Computational Linguistics.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1437–1447. Association for Computational Linguistics.

Xiujun Li, Zachary C. Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *CoRR*, abs/1612.05688.

Zichuan Lin, Jing Huang, Bowen Zhou, Xiaodong He, and Tengyu Ma. 2021. Joint system-wise optimization for pipeline goal-oriented dialog system. *CoRR*, abs/2106.04835.

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

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1468–1478. Association for Computational Linguistics.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskénazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 165–177. Association for Computational Linguistics.

Alexandros Papangelis, Yi-Chia Wang, Piero Molino, and Gökhan Tür. 2019. Collaborative multi-agent dialogue model training via reinforcement learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 92–102. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. SOLOIST: few-shot task-oriented dialog with A single pre-trained auto-regressive model. *CoRR*, abs/2005.05298.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 432–437.

Desik Rengarajan, Gargi Vaidya, Akshay Sarvesh, Dileep M. Kalathil, and Srinivas Shakkottai. 2022. Reinforcement learning with sparse rewards using guidance from offline demonstration. *CoRR*, abs/2202.04628.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gökhan Tür. 2018a. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 3 (Industry Papers)*, pages 41–51. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tür, Gökhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry P. Heck. 2018b. Building a conversational agent overnight with dialogue self-play. *CoRR*, abs/1801.04871.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. 2019. How to build user simulators to train rl-based dialog systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1990–2000. Association for Computational Linguistics.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4661–4676. Association for Computational Linguistics.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2022. BORT: back and denoising reconstruction for end-to-end task-oriented dialog. *CoRR*, abs/2205.02471.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press.

Ryuichi Takanobu, Runze Liang, and Minlie Huang. 2020. Multi-agent task-oriented dialog policy learning with role-aware reward decomposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 625–638. Association for Computational Linguistics.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. 2021. Transferable dialogue systems and user simulators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 152–166. Association for Computational Linguistics.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3732–3741. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yen-Chen Wu, Bo-Hsiang Tseng, and Milica Gasic. 2020. Actor-double-critic: Incorporating model-based critic for task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 854–863. Association for Computational Linguistics.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. UBAR: towards fully end-to-end task-oriented dialog system with GPT-2. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14230–14238. AAAI Press.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. 2020. MOPO: model-based offline policy optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, ACL 2020*, pages 109–117.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020a. Minimize exposure bias of seq2seq models in joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 236–246. Association for Computational Linguistics.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9604–9611. AAAI Press.

Zheng Zhang, Lizi Liao, Minlie Huang, Xiaoyan Zhu, and Tat-Seng Chua. 2019a. Neural multimodal belief tracker with adaptive attention for dialogue systems. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2401–2412. ACM.

Zhirui Zhang, Xiujun Li, Jianfeng Gao, and Enhong Chen. 2019b. Budgeted policy learning for task-oriented dialogue systems. *CoRR*, abs/1906.00499.

Xinyan Zhao, Bin He, Yasheng Wang, Yitong Li, Fei Mi, Yajiao Liu, Xin Jiang, Qun Liu, and Huanhuan Chen. 2022. Unids: A unified dialogue system for chit-chat and task-oriented dialogues. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, DialDoc@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 13–22. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1458–1467. Association for Computational Linguistics.

Li Zhou, Kevin Small, Oleg Rokhlenko, and Charles Elkan. 2017. End-to-end offline goal-oriented dialog policy learning via policy gradient. *CoRR*, abs/1712.02838.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

A Training Details

We implement US and DS models with Huggingface Transformers repository of version 4.2.2. We initialize it with DistilGPT-2, a distilled version of GPT-2. During offline supervised learning, the minibatch base size is set to be 2 with gradient accumulation steps of 16, we use AdamW optimizer and a linear scheduler with 20 warm up steps and maximum learning rate 1×10^{-4}, and the gradient clip is set to be 5. The total epochs are 30 (it takes about 20 hours on NVIDIA Tesla 2V100-SXM2-32GB) and we select the best model on the test set.

In the stage of online RL, we connect three linear layers (768*512 $\rightarrow$ ReLU $\rightarrow$ 512*512 $\rightarrow$ ReLU$\rightarrow$512*1) as our value network. The learning rate of policy and value are 1×10^{-6} and 5×10^{-6} respectively. The batch size for RL optimization is 4, and the hyper-parameters is PPO2: γ is 0.99, ϵ is 0.1 and τ is 0.95. Two important hyper-parameters in policy constraint λ we set to be 0.75 and the probability threshold is 0.9. The replay buffer size of our algorithm is 200. The whole RL optimized epoch is 20 (it takes about 4 hours on a single NVIDIA Tesla V100-SXM2-32GB), we will evaluate the online interaction quality after every epoch (about 1 hour) and choose the excellent model for offline evaluation (about 40 min).

The reward setting of our framework: Task Reward R_{task}, Domain Reward R_d and Turn Reward $R^{\#}_{\text{turn}}$ are listed in Table 6:

Reward Type	Success	Failure
R_{task}	20	-10
R_d	5	-5
User $R^{\#}_{\text{turn}}$	0 0 0	-1 -1 -1
System $R^{\#}_{\text{turn}}$	0 0 0	-2.5 -1 -2.5

Table 6: Reward setting of our online RL experiment.

B Examples of Bad Case

After offline supervised learning to obtain US and DS with dialogue ability, when we let them interact online, we will find that there are some problems in the dialogue, we regard it as low-quality data, : 1) A large number of repetitions of meaningless words appear in the sentence; 2) The key special token representing the start or end of the sequence does not appear; etc. These two types of data are shown in the Table 8:

C Examples of Improvement from RL

In order to demonstrate the effectiveness of the RL method, we show in Table 7 the DS obtained based on supervised learning and the DS after RL optimization with different feed backs for the input of the same goal and US. We enumerate the original failed dialogues and the successful dialogues after policy optimization.

MUL0072.json	
Utter	yes, i also need a hotel room with free wifi also in [$hotel\_area$], [$hotel\_stars$] star, and [$hotel\_pricerange$] please.
Act	act_inform hotel_name act_offerbook hotel_none
Resp	the [$hotel\_name$] meets your requirement -s. would you like me to book this for you?
Act_{SL}	act_inform hotel_choice act_request hotel_area
$Resp_{SL}$	there are [$value\_count$] hotel -s that meet your criteria. is there a certain area you would like to stay in?
Act_{RL}	act_inform hotel_choice hotel_internet hotel_name hotel_name hotel_name hotel_pricerange hotel_stars hotel_type act_offerbook hotel_none
$Resp_{RL}$	there are [$value\_count$] hotel -s that meet your criteria [$hotel\_name$], [$hotel\_name$], [$hotel\_name$], [$hotel\_name$], and [$hotel\_name$]. would you like to book [$value\_count$] of these?
MUL0116.json	
Utter	yes, i am looking for a [$restaurant\_pricerange$] -ly priced restaurant with [$restaurant\_food$] food.
Act	act_offerbook restaurant_none act_recommend restaurant_name
Resp	i can recommend [$restaurant\_name$]. would you like to book a table with them?
Act_{SL}	act_inform restaurant_area restaurant_choice restaurant_food restaurant_pricerange act_request restaurant_food
$Resp_{SL}$	i have [$value\_count$] [$restaurant\_pricerange$] [$restaurant\_food$] restaurant -s in the [$restaurant\_area$]. do you have a preferred area of town?
Act_{RL}	act_inform restaurant_name act_offerbook restaurant_none
$Resp_{RL}$	[$restaurant\_name$] fits your criteria. would you like me to book a table for you?

Table 7: Example of RL improvement.

D Examples of Diversity

Through the observation of online interactive dialogue, we can find that RL helps our model has the ability to generate richer dialog action. At the same time, our natural language generation is also richer and more diverse. We enumerate examples of which are shown in the Table 9, which also

Bad Type	Examples
Redundant sequence	• < sos_sa > [act_inform] [attraction_type] [attraction_type] [attraction_type] [act_request] [attraction_type] < eos_sa > < sos_sr > what type of attraction are you looking for? in the south, we have [attraction_type], [attraction_type], [attraction_type],[attraction_type],[attraction_type], ,[attraction_type] [attraction_type],[attraction_type],[attraction_type], [attraction_type], [attraction_type],[attraction_type],[attraction_type],and [attraction_type]. [attraction_type], if you tell me what you are looking for, we can narrow it • <sos_sa> [act_inform] [hotel_area] [hotel_area] [hotel_area] [hotel_choice] [hotel_choice] [hotel_choice] [hotel_choice] [hotel_choice] [hotel_choice] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars][hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars] [hotel_stars][hotel_stars] [hotel_stars] [hotel_stars]
No key special token	• < sos_ua > [act_inform] [hotel_area] [hotel_stars] star hotel -s in the [hotel_area]. <eos_uu> • < sos_ua > [act_inform] [hotel_area] [hotel_stars] star hotel -s in the [hotel_area]. <eos_uu> • < sos_ua > with free parking and wifi? <eos_uu> • < sos_ua > [act_inform] [attraction_address] [attraction_area] [attraction_fee] [value_price] pounds to enter. <eos_sr>

Table 8: Low quality data in our online generation.

SNG01290.json	
$Utter_{GT}$	i am interesting in info about [hotel_stars] star hotel -s and such that offer free wifi.
$Resp_{GT}$	i have [value_count] hotel -s that match the description of [hotel_stars] star rating -s. are you looking for a hotel in the centre or south area?
$Utter1$	i am looking for a [hotel_pricerange] hotel with free wifi.
$Resp1$	i have [value_count] [hotel_pricerange] hotel -s with free wifi. is there a specific area you would like to stay in?
$Utter2$	i am looking for a [hotel_pricerange] place to stay with free wifi.
$Resp2$	i have [value_count] [hotel_pricerange] hotel -s with free wifi. is there a particular area you would like to stay in?
$Utter_{GT}$	the location does not matter so long as it has free wifi
$Resp_{GT}$	[hotel_name] is supposed to be great and it offers free wifi. would you like me to book you a room?
$Utter1$	i would like to stay in the centre of town, and i am open to place -s that have a [hotel_stars] star rating.
$Resp1$	[hotel_name] meets your criteria. would you like to book a room?
$Utter2$	i would like to stay in a hotel with a [hotel_stars] star rating please.
$Resp2$	i have [value_count] options for you to choose from. do you prefer a certain area of town?
$Utter_{GT}$	book it for [hotel_people] people and [hotel_stay] nights starting from [hotel_day].
$Resp_{GT}$	sorry, it looks like the hotel cannot accommodate you for that amount of time.
$Utter1$	yes, please. i need it for [hotel_stay] nights starting on [hotel_day]. there will be [hotel_people] people.
$Resp1$	booking was successful. reference number is [hotel_reference]. is there anything else i can help you with?
$Utter2$	no, i do not have a preference. i just need to book it for [value_count] people staying for [value_count] nights.
$Resp2$	i have booked you at the [hotel_name]. your reference number is [hotel_reference]. is there anything else i can help you with?
$Utter_{GT}$	perhaps starting [hotel_day] instead?
$Resp_{GT}$	great, your reference number is [hotel_reference]. is there anything else i can help with?
$Utter1$	no, that s all i need. thank you for your help.
$Resp1$	thank you for using our system!
$Utter2$	no, that s all i need. thank you!
$Resp2$	thank you for using our service. goodbye.!
$Utter_{GT}$	no that s it all. thanks for your help.
$Resp_{GT}$	wonderful. glad to help.
$Utter1$	
$Resp1$	
$Utter2$	
$Resp2$	

Table 9: Example of diversity.

explains why the BLEU value drops in our experiments.

E Ontology

The ontology defines all entity properties called slots and all possible values for each slot, which concludes goal slot, act slot and belief state slot, special token conclude the start and end token of sentences or actions, database query result and padding token. Special tokens and ontology are illustrated as shown in Table 10.

Type	Representations
Goal Slot Tokens	'restaurant_info_area', 'restaurant_info_food', 'restaurant_info_name', 'restaurant_info_pricerange','restaurant_book_day', 'restaurant_book_people', 'restaurant_book_time', 'restaurant_reqt_address', 'restaurant_reqt_area', 'restaurant_reqt_food', 'restaurant_reqt_phone', '$restaurant_{r}eqt_{p}ostcode$', 'restaurant_reqt_pricerange', 'hotel_info_area', 'hotel_info_internet', 'hotel_info_name', 'hotel_info_parking', 'hotel_info_pricerange', 'hotel_info_stars', 'hotel_info_type', 'hotel_book_day', 'hotel_book_people', 'hotel_reqt_type', 'hotel_book_stay', 'hotel_reqt_address', 'hotel_reqt_area', 'hotel_reqt_internet', 'hotel_reqt_parking', 'hotel_reqt_phone', 'hotel_reqt_postcode', 'hotel_reqt_pricerange', 'hotel_reqt_stars', 'attraction_info_area', 'attraction_info_name', 'attraction_info_type', 'attraction_reqt_address', 'attraction_reqt_area', 'attraction_reqt_fee', 'attraction_reqt_phone', 'attraction_reqt_postcode′, 'attraction_reqt_type', 'train_info_arriveBy', 'train_info_day', 'train_info_departure', 'train_info_destination', 'train_info_leaveAt', 'train_book_people', 'train_reqt_arriveBy', 'train_reqt_duration', 'train_reqt_leaveAt', 'train_reqt_price', 'train_reqt_trainID', 'taxi_info_arriveBy', 'taxi_info_departure', 'taxi_info_destination', 'taxi_info_leaveAt', 'taxi_reqt_type', 'taxi_reqt_phone', 'police_reqt_address','police_reqt_phone', 'police_reqt_postcode', 'hospital_info_department', 'hospital_reqt_address', 'hospital_reqt_phone','hospital_reqt_postcode',
Special Tokens	'<pad>', '<unk>', '<eos_g>', '<eos_ua>', '<eos_uu>', '<eos_b>', '<eos_d>', '<eos_sa>', '<eos_sr>', '<sos_g>', '<sos_ua>','<sos_uu>', '<sos_b>', '<eos_d>', '<sos_sa>', '<sos_sr>', '<sos_db>', '<eos_db>', 'restaurant_db_0', 'restaurant_db_1','restaurant_db_2', 'hotel_db_0', 'hotel_db_1', 'hotel_db_2', 'attraction_db_0', 'attraction_db_1', 'attraction_db_2', 'train_db_0', 'train_db_1', 'train_db_2'
Action Slot Tokens	['act_inform', 'general_none', 'act_request', 'act_reqmore', 'restaurant_food', 'act_thank', 'act_offerbook', 'train_leaveAt', 'restaurant_name', 'restaurant_area', 'restaurant_pricerange', 'hotel_area', 'act_offerbooked', 'hotel_name', 'train_destination', 'hotel_type', 'train_departure', 'hotel_pricerange', 'attraction_type', 'train_arriveBy', 'train_day', 'attraction_area', 'act_bye', 'attraction_name', 'hotel_stars', 'act_welcome', 'hotel_stay','restaurant_none', 'act_recommend', 'attraction_address', 'hotel_none', 'train_trainID', 'restaurant_time', 'hotel_parking', 'hotel_internet', 'hotel_day', 'train_none', 'train_price', 'attraction_fee', 'restaurant_day', 'restaurant_address', 'restaurant_choice', 'attraction_phone','hotel_people', 'train_people', 'attraction_postcode', 'restaurant_people', 'restaurant_reference', 'act_nooffer', 'hotel_reference', 'train_reference', 'act_select', 'restaurant_phone', 'taxi_type', 'attraction_choice', 'act_greet', 'train_choice', 'restaurant_postcode', 'taxi_phone', 'taxi_departure', 'taxi_leaveAt', 'hotel_address', 'train_duration', 'taxi_destination', 'act_nobook', 'booking_none', 'hotel_phone', 'hotel_postcode', 'taxi_arriveBy', 'taxi_none', 'booking_day', 'attraction_none', 'booking_time', 'booking_people', 'hospital_postcode', 'hospital_phone','hospital_address', 'police_address', 'police_postcode', 'police_phone','hospital_department', 'hospital_none', 'police_name', 'attraction_pricerange', 'booking_stay', 'police_none', 'train_leaveat', 'booking_reference', 'train_arriveby', 'booking_name', 'taxi_leaveat','hotel_time', 'attraction_open', 'restaurant_stay', 'taxi_arriveby','hotel_choice']

Table 10: Speicial tokens and ontology defined in our experiment.