JoTR: A Joint Transformer and Reinforcement Learning Framework for Dialogue Policy Learning

Wai-Chung Kwan[1,3*], **Huimin Wang**[2*], **Hongru Wang**[1,3], **Zezhong Wang**[1,3], **Bin Liang**[1,3], **Xian Wu**[2], **Yefeng Zheng**[2], **Kam-Fai Wong**[1,3]
[1]The Chinese University of Hong Kong, [2]Jarvis Lab, Tencent
[3]MoE Key Laboratory of High Confidence Software Technologies
{hmmmwang, kevinxwu, yefengzheng}@tencent.com
{wckwan, hrwang, zzwang, kfwong}@se.cuhk.edu.hk, bin.liang@cuhk.edu.hk

Abstract

Dialogue policy learning (DPL) aims to determine an abstract representation (also known as action) to guide what the response should be. Typically, DPL is cast as a sequential decision problem across a series of predefined action candidates. However, such static and narrow actions can limit response diversity and impede the dialogue agent's adaptability to new scenarios and edge cases. To overcome these challenges, we introduce a novel **Jo**int **T**ransformer **R**einforcement Learning framework, coined as **JoTR**, where a text-to-text Transformer-based model is employed to directly generate dialogue actions. More concretely, JoTR formulates a token-grained policy, facilitating more dynamic and adaptable dialogue action generation without the need for predefined action candidates. This method not only enhances the diversity of responses but also significantly improves the system's capability to manage unfamiliar scenarios. Furthermore, JoTR utilizes Reinforcement Learning with a reward-shaping mechanism to efficiently fine-tune the token-grained policy. This allows the model to evolve through interactions, thereby enhancing its performance over time. Our extensive evaluation demonstrates that JoTR surpasses previous state-of-the-art models, showing improvements of 9% and 13% in success rate, and 34% and 37% in the diversity of dialogue actions across two benchmark dialogue modeling tasks respectively. These results have been validated by both user simulators and human evaluators. Code and data are available at https://github.com/KwanWaiChung/JoTR.

Keywords: Dialogue Policy Learning, JoTR Framework, Reinforcement Learning, Language Model, Dialogue System.

1. Introduction

Dialogue Policy Learning (DPL) seeks to identify optimal actions for a dialogue agent to manage conversation flow and deliver contextually relevant responses. These actions, abstract representations of strategic decisions, are typically optimized using Reinforcement Learning (RL) (Lipton et al., 2016; Li et al., 2017; Peng et al., 2018; Takanobu et al., 2019; Wang et al., 2020; Li et al., 2020b; Wang et al., 2023; Kwan et al., 2023). A dialogue action typically includes one or more combinations of a domain name, intent type, and slot name, collectively known as an "atomic action" (Li et al., 2020a). Traditional DPL often prioritizes frequent actions to boost RL efficiency. This approach, while effective in some scenarios, may unintentionally restrict the range of responses and hinder the dialogue agent's ability to adapt to new circumstances and edge cases. This constraint stems from the static and narrow focus of the action candidates.(Wang and Wong, 2021). In real-world scenarios, relying solely on a set of incomplete action candidates, which only cover a subset of potential atomic actions or their combinations, may not always yield the most suitable responses. This is because the complexity and unpredictability of human interactions often require a broader range of responses than what is typically covered by the most frequent actions. This issue is illustrated in Figure 1, where the user concurrently requests the address, postcode, and phone number of a restaurant. Among the most frequently chosen action candidates, the optimal response to the user's query is "Action 1". However, this action only provides the address, thus only partially meeting the user's needs. In contrast, "Action 2", generated by our approach, delivers a more appropriate response. It thoroughly addresses all the requested slots, thereby enhancing the user experience. This instance highlights the necessity for a more flexible and adaptable strategy in Dialogue Policy Learning (DPL), which is precisely what our method strives to offer.

To expand action candidates, Li et al. (2020a) proposed a GRU-based decoder for sequential atomic action prediction. This approach enhances response flexibility but struggles with the exponential growth of dialogue state and atomic action space. Wang and Wong (2021) proposed a multi-agent RL framework, which, despite promising results, allows only one atomic action per turn, leading to more turns and potentially unnatural utterances. A related study used a Transformer for

*These authors contributed equally.

LREC-COLING 2024, pages 9578–9588
20-25 May, 2024.

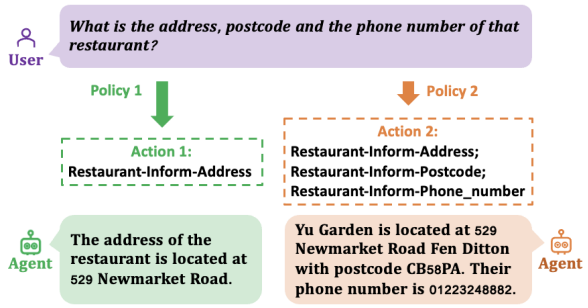

Figure 1: This demonstrates two distinct dialogue actions produced by different policies. The response generated from Action 1, selected from predefined candidates, addresses only one aspect of the user's question. In contrast, Action 2, generated by JoTR, comprehensively responds to all the required elements of the user's query.

action production but required complex domain-specific settings, forcing the agent to select from fixed words (Geishauser et al., 2022). To address these challenges and achieve efficient and effective dialogue action generation, we present a novel **Jo**int **T**ransformer **R**einforcement Learning Framework (**JoTR**). JoTR's primary innovation is its ability to directly generate actions with a token-grained policy, a significant departure from the traditional reliance on fixed, human-defined templates. This is achieved through a novel formulation for DPL, subtle instruction design, and the integration of Transformer and Reinforcement Learning. Specifically, JoTR uses a Transformer encoder to convert the flattened language representations of various dialogue information into embeddings. These embeddings are then fed into another Transformer, the token-grained policy, to autoregressively generate the dialogue actions in a structured format. To optimize this policy, JoTR employs Reinforcement Learning combined with reward shaping settings, resulting in more efficient responses that require fewer interaction turns, a critical aspect of enhancing user experience in dialogue systems. JoTR not only achieves efficient dialogue modeling but also improves response diversity and adaptation capabilities. We evaluate JoTR's performance on two benchmark multi-domain dialogue modeling tasks using common metrics like success rate, number of turns, and average rewards. To assess response diversity, we introduce a new metric–distinct actions–representing the total number of unique actions applied during a series of conversations.

In summary, our key contributions include:

- Treating DPL as a generation problem, we introduce JoTR, a transformer-based reinforcement learning framework that learns the token-grained policy.
- We integrate a reward-shaping mechanism into the reinforcement learning fine-tuning process to ensure efficient training.
- We conduct extensive experiments based on a new metric–distinct actions–for response diversity on two benchmarks, demonstrating JoTR's superior performance.

2. Related Work

Dialogue Policy Learning The conventional method for building a task-oriented dialogue (TOD) system involves a pipeline approach with four interconnected modules: natural language understanding, dialogue state tracking, dialogue policy learning (DPL), and natural language generation Kwan et al. (2023). Reinforcement learning has been the mainstream approach to optimize the dialogue policy (Levin et al.; Singh et al.; Gašić et al., 2010). To tackle the challenges of large state-action spaces and low exploration efficiency in DPL, hierarchical Reinforcement Learning has been employed to divide the complex task into subtasks (Budzianowski et al., 2017; Peng et al., 2017; Kristianto et al., 2018; Tang et al., 2018). Other researchers have used reward learning and reward shaping for denser rewards and faster learning (Su et al., 2015, 2016; Wang et al., 2020). Recently, some studies have applied multi-agent Reinforcement Learning to DPL (Liu and Lane; Zhang et al., 2020), with some proposing a joint learning process for the dialogue system and user agent, and others partitioning the action space into subspaces Wang and Wong (2021). These works frame model DPL as a classification problem where the policy chooses a suitable dialogue action from a predefined action list. Contrasting to these methodologies, our approach obviates the need for a predefined action list, generating the dialogue action instead.

Pre-trained Language Model Recent research has made significant strides in task-oriented dialogue (TOD) by fine-tuning pre-trained language models (Budzianowski and Vulić, 2019; Hosseini-Asl et al., 2020; Lee, 2021; Su et al., 2022; Zhao et al., 2022), effectively unifying the learning process of various pipeline modules. The SOLOIST model leverages GPT-2 to sequentially generate the belief state, dialogue action, and response Peng et al. (2021). It's first fine-tuned on a large dialogue corpus, then on the target dataset. He et al. (2022) extended this approach by incorporating unlabelled dialogue data through semi-supervised learning. However, sequential subtask generation can lead to error accumulation. To address this, PPTOD concurrently generates the dialogue action

and response after formulating the belief state, preventing error propagation (Su et al., 2021). These methods require a large dialogue corpus for pretraining and fine-tuning. In contrast, our approach only needs a small amount of data for the initial model warm-up and uses Reinforcement Learning for fine-tuning, eliminating the need for labeled data.

3. JOTR

We utilize a Transformer-based model to directly generate dialogue actions, distinguishing it from conventional approaches that rely on selecting dialogue actions from a predefined set of atomic dialogue actions and their combinations. The architecture of our model is illustrated in Figure 2. In this section, we first provide a formal definition of DPL, and then present the overview of our approach.

3.1. Problem Definition

The objective of DPL is to learn a policy capable of generating dialogue actions that interact with a user, given the belief state and the database results, with the aim of fulfilling the user's goal $G = (C, R)$. In this context, C denotes the user constraints (e.g. a flight ticket to Seattle) and R represents the information the user seeks (e.g. the price of the flight ticket). The belief state keeps track of the user's constraints throughout the dialogue. It is defined as a list of domain, slot, and value triplets (e.g. *[(flight, destination, Seattle), (flight, day, tomorrow)]*). The dialogue action is represented as a list of domain, intent, slot, and value quadruples (e.g. *[(flight, request, time, ?)]*). An external database provides relevant entries to the dialogue policy based on the belief state. Figure 1 provides an illustrative example of such a process.

3.2. Dialogue State Text Encoding

The encoder generates state embeddings $e_u, e_s, e_b, e_d \in \mathbb{R}^d$ by encoding the flattened textual representations of four elements: user action, system action, belief state, and the database result. These linearized textual representations are referred to as the dialogue state text. The user action is a sequence of tokens derived from atomic dialogue action triplets, each comprising the domain D, intent I, and slot S. This sequence is represented as $D_1, I_1, S_1, \ldots, D_{N^u}, I_{N^u}, S_{N^u}$, where N^u denotes the number of atomic user dialogue actions. For instance, "*Attraction Request Address Attraction Request Phone*" is a valid example. The system action, similar to the user action, is represented as $D_1, I_1, S_1, \ldots, D_{N^s}, I_{N^s}, S_{N^s}$, with N^s indicating the number of atomic system dialogue actions. The belief state is a sequence formed by concatenating the belief state triplets, where each triplet is composed of the domain D, slot S, and value V. This sequence can be represented as $D_1, S_1, V_1, \ldots, D_{N^b}, S_{N^b}, V_{N^b}$. A typical example could be "*Attraction Name cherry hinton water play.*". The database result is represented as $D_1, Q_1, \ldots, D_{N^d}, Q_{N^d}$, with N^d denoting the number of queried domains and Q denoting the number of matched entities in the database. An illustrative example would be "*Attraction two*".

To obtain the state embeddings e_u, e_s, e_b, e_d, the [CLS] token, a common sentence representation placeholder, is prefixed to each dialogue state text (Devlin et al., 2018). The output representation of the [CLS] token of each dialogue state text is used as the state embedding. Initial experiments revealed inferior performance if the model is only fed with the state embeddings, likely due to the model's confusion about the varying types of state information being encoded. Therefore, a context embedding was constructed for each dialogue state text. The context embeddings are added with the state embeddings individually to produce the state $s \in \mathbb{R}^{4\times d}$.

3.3. token-grained Dialogue Policy

We have formulated the problem of dialogue policy learning (DPL) as a Markov Decision Process (MDP) on the word level. In this process, the system agent observes the current dialogue state s, executes an action a (generated by the policy function), receives a response, a reward r, and the updated dialogue state s'. This cycle continues until the conversation ends. The action a is textually represented as a sequence of words $w_{1:N} = w_1 \ldots w_N$. The policy function can be detailed as a series of conditional probabilities:

$$\pi_\theta(a|s) = \prod_{i=1}^{N} \mathcal{P}_\theta(w_i|w_{1:i-1}, s), \tag{1}$$

where $\mathcal{P}$ is approximated with a Transformer encoder-decoder network parameterized by θ, representing the probability of the word w_i condition on the preceding words and state. As shown in Figure 2, the Transformer decoder generates the dialogue action text word by word, beginning with the start signal "[start]", conditioned on the dialogue state, and proceeds until it encounters the stop signal "[end]". Additionally, an action interpreter decodes the dialogue action text into a structured format, populating slot values from the database, and yielding the final dialogue action. This process involves verifying whether the dialogue action text adheres to the domain, intent, and slot order, discarding any words that violate these conditions. The policy π,

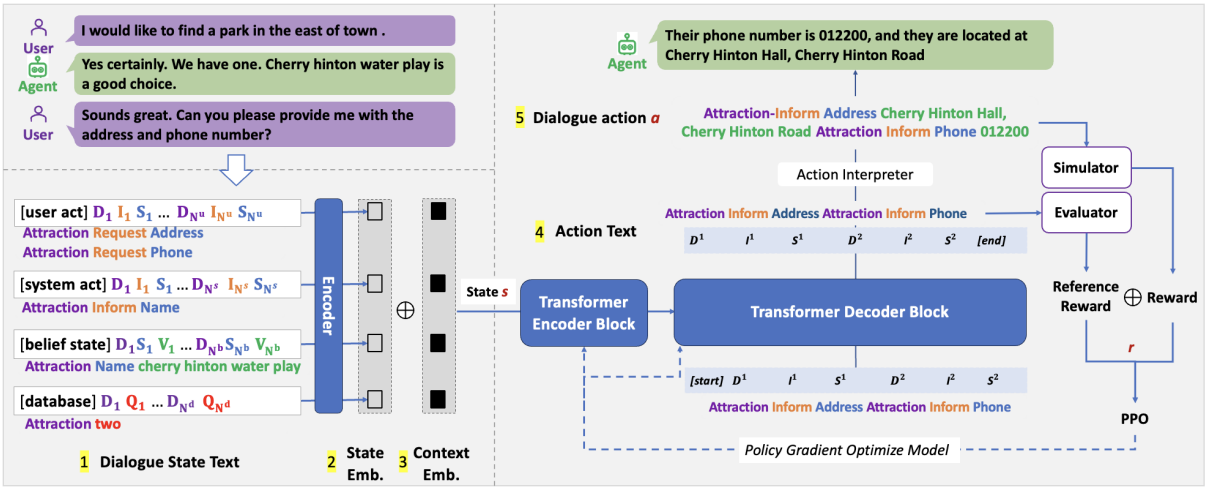

Figure 2: The joint Transformer and Reinforcement Learning framework illustration consists of: 1) (Left Part) Text Encoding - The encoder processes user act, system act, belief state, and database query results to form the state,**D** denotes the Domain (e.g., restaurant), **I** is the Intent (e.g., inform), **S** is the slot type (e.g., request), **V** is the slot's value (e.g., date), and **Q** represents the Number of matched queries in the database; 2) (Right Part) Model Optimization - The state directs action generation, with the Action Interpreter generating structured dialogue actions. The Transformer-based policy model undergoes interactive optimization through Reinforcement Learning from scratch.

parameterized by θ, is optimized using Reinforcement Learning to minimize the negative expected cumulative future rewards:

$$\mathcal{L}_\theta = -\mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)} \left[\sum_{t=1}^{T} r(s_t, a_t) \right], \qquad (2)$$

where s_t and a_t are the state and dialogue action turn t, and T is the maximum turn. In practice, the expected gradient for a dialogue session can be approximated by using a Monte Carlo sample from $\mathcal{P}_\theta$. For each session example, the gradient is approximated as:

$$\nabla_\theta \mathcal{L}_\theta \approx -\sum_{t=1}^{T} r(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t|s_t) = -\sum_{t=1}^{T} r(s_t, w^t_{1:N^t}) \sum_{i=1}^{N^t} \nabla_\theta \log \mathcal{P}_\theta(w^t_i | w^t_{1:i-1}, s_t), \qquad (3)$$

where N^t is the length of the action text at turn t.

3.4. JoTR for Efficient Policy Training

We employ Proximal Policy Optimization (PPO) (Schulman et al., 2017) to optimize the policy. More specifically, we minimize the objective function for each session example.

$$\begin{aligned}\mathcal{L}_\theta &= \sum_{t=1}^{T} -\hat{\mathbb{E}}_t \left[\min \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}^\phi_t, \right. \right. \\ &\left. \left. \text{clip} \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}^\phi_t \right) \right] \\ &= \sum_{t=1}^{T} -\hat{\mathbb{E}}_t \left[\min \left(\frac{\prod_{i=1}^{N} \mathcal{P}_\theta(w_i | w_{1:i-1}, s_t)}{\prod_{i=1}^{N} \mathcal{P}_{\theta_{\text{old}}}(w_i | w_{1:i-1}, s_t)} \hat{A}^\phi_t, \right. \right. \\ &\left. \left. \text{clip} \left(\frac{\prod_{i=1}^{N} \mathcal{P}_\theta(w_i | w_{1:i-1}, s_t)}{\prod_{i=1}^{N} \mathcal{P}_{\theta_{\text{old}}}(w_i | w_{1:i-1}, s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}^\phi_t \right) \right], \end{aligned} \qquad (4)$$

where $\hat{A}^\phi_t = r(s_t, a_t) + \gamma V^\phi(s_{t+1}) - V^\phi(s_t)$ is the advantage estimation and $V^\phi(s_t)$ is the value function estimated by the critic parameterized by ϕ.

To improve the efficiency and quality of the dialogue response, we integrate reward-shaping into the Reinforcement Learning fine-tuning process. The goal is to prevent the policy from generating protracted yet predominantly irrelevant dialogue actions during optimization with PPO. To achieve this, we propose a reward-shaping function assigning supplementary rewards to guide the model to learn. Formally, we replace the $r(s_t, a_t)$ with $\hat{r}(s_t, a_t, G)$ defined as

$$\hat{r}(s_t, w^t_{1:N^t}, G) = r(s_t, a_t) + F(G, w^t_{1:N^t}), \qquad (5)$$

where G denotes the user goal and F represents the shaping reward. We design F to provide different rewards based on the following: (1) If the system informs a slot present in the user's request slot list, it receives an additional λ reward. Conversely, informing other slots receives an additional

-1 reward. (2) If the system requests a slot included in the user's inform slot list, it receives an additional λ reward. However, requesting other slots results in an additional -1 reward. λ is a hyperparameter that controls the aggressiveness of the dialogue agent to inform or request additional slots. We try λ with values 3, 4, 5, 6, 7. We find that the range of 3 to 5 yielded favorable results during validation. Higher λ values encourage the dialogue agent to attempt many actions in a single turn, where one successful inform or request action offsets the negative rewards incurred by other irrelevant actions. We pick $\lambda = 3$ for all the experiments.

4. Experiments and Results

Experiments are carried out on MultiWOZ 2.0 (Budzianowski et al., 2018), utilizing a publicly accessible agenda-based user simulator (Zhu et al., 2020), and on the SGD dataset with our developed rule-based simulator. Furthermore, we incorporate human evaluations, in which evaluators interact with various models and assess the success of the dialogue upon its completion. While all models, except SimpleTOD, are optimized in the dialogue action space, SimpleTOD takes the utterance dialogue history as input and generates both the dialogue action and the system utterance.

5. Evaluation Metrics

We employ three primary evaluation metrics: success rate, the average number of turns, and average rewards, aligning with previous work (Wang and Wong, 2021). Additionally, we introduce a new metric, distinct actions, to assess response diversity. A detailed explanation of these metrics: 1) **Success rate (*Succ.*)**. A dialogue session is considered successful if it fulfills all the user requests, and reserves an entity that meets the user's specifications if necessary. 2) **Average number of turns (*Turn*)** is calculated by counting the number of interactions between the two parties, with each full interaction counted as two turns. 3) **Average rewards (*Rew.*)** is the total cumulative reward obtained in each dialogue session. 4) **Distinct actions (*#Acts*)** calculate the number of different dialogue actions utilized during a series of conversations indicating the coverage of the dialogue actions.

Furthermore, as SimpleTOD uses text utterances as input, a natural language generation (NLG) component is required to transform the user's dialogue actions into utterances. However, SimpleTOD's performance heavily relies on the NLG component. Hence, we also evaluate it against a testing corpus, which we believe is a more effective method than using a user simulator equipped with natural language understanding and NLG modules that could potentially introduce variances.

6. Training & Implementation Details

We implement all the models in PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020). We use a randomly initialized Transformer encoder with 1 hidden layer, 1 head, and a hidden size of 256 as the dialogue state text encoder of JoTR. In $\text{JoTR}_{\text{pretrained}}$, we use DistilBert (Sanh et al., 2019) with pre-trained weights from Huggingface as the initial weights for the dialogue state text encoder. For the encoder-decoder model, we use a Transformer with 1 hidden layer, 1 attention head, and a hidden size of 256 for both the encoder and decoder. The total number of parameters is 5M. All variants of JoTR and MLP_{ppo} are warmed up before Reinforcement Learning by first performing supervised learning on a training set to be consistent with previous work (Wang and Wong, 2021). We use the same set of 10K dialogue turns sampled randomly from the original training set to warm up all models. A separate, non-overlapping set of 3K dialogue turns is used for validation in the warm-up phase. The same set of hyperparameters is used on MultiwWOZ and SGD in both pretraining and PPO fine-tuning. For the warm-up phase, the models are trained using a batch size of 32 and a learning rate of 3×10^{-4}. The models are trained for 80 epochs but we include an early stopping mechanism that halted training when no improvement is observed in the validation set over five consecutive epochs. In PPO training, we use an actor learning rate of 5×10^{-7} and a critic learning rate of 1×10^{-4}. The critic is a Transformer with identical architecture to the encoder-decoder model. The maximum interaction turn allowed is 40. The main reward provided by the environment is -1 in every turn and a reward of 80 or -40 at the end for successful or failed dialogue respectively.

7. Simulator Details

We employ an agenda-based simulator to sample a user goal based on the slot distribution in the MulwiWOZ training set. The user goal remains undisclosed to the dialogue agent. The simulator maintains a stack (i.e., user agenda) storing all necessary user actions to achieve the goal during the conversation, responding to system actions based on predefined rules. We've also implemented an agenda-based simulator for SGD, adhering to the widely accepted user simulator design approach from previous works (Schatzmann et al., 2007; Wang and Wong, 2021; Kwan et al., 2023). This implementation is a valuable tool for future research and development in this field.

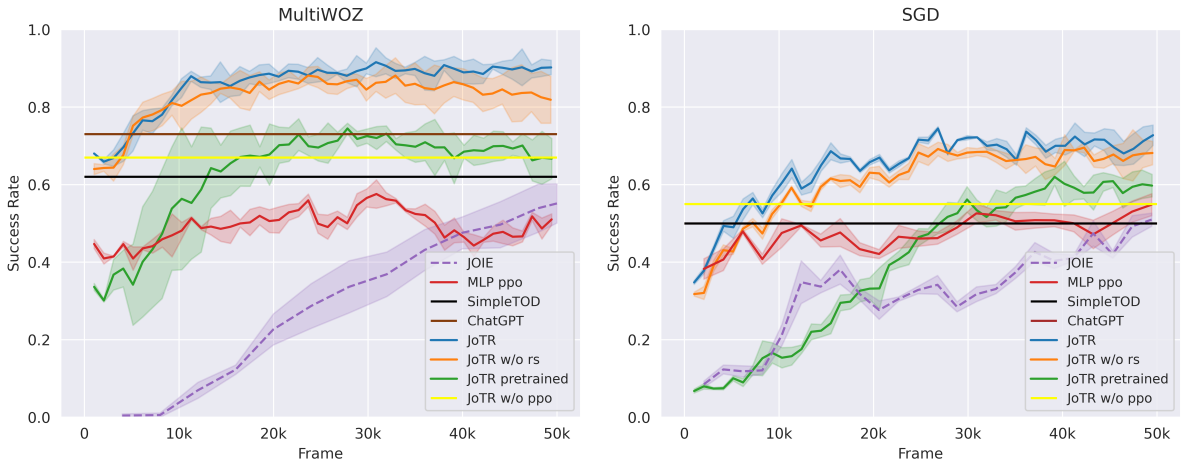

Figure 3: The learning curve of various models on MultiWOZ and SGD, with the mean and standard deviation illustrated over 5 runs. JoTR outperforms other models in performance and exhibits enhanced training efficiency, achieving a success rate of 0.93 within 50k frames. In contrast, JOIE, the previous state-of-the-art model, reaches a success rate of 0.91 over 400k frames.

7.1. Baseline Agents

We compare our model, JoTR, to five other models.

- **JOIE** (Wang and Wong, 2021), the current state-of-art(SOTA) on MultiWOZ, is a collaborative multi-agent model that generates an atomic action per turn.
- **MLP$_{ppo}$** is an agent optimized with PPO with fixed action candidates.
- **SimpleTOD** (Hosseini-Asl et al., 2020) is a GPT-2-based agent trained with supervised learning to generate dialogue actions along with both belief states and responses based on the dialogue history.
- **DASP** (Jhunjhunwala et al., 2020), is an LSTM-based agent that is trained with human supervision to select among N-best action candidates based on the dialogue history.
- **ChatGPT** built on InstructGPT (Ouyang et al., 2022), is an effective conversational agent used in our work to generate dialogue actions from action history using a zero-shot prompt.

To further demonstrate the advantages of our model, JoTR, we also compare it to three variants. **JoTR$_{w/o\ rs}$** does not use reward shaping. **JoTR$_{w/o\ ppo}$** is only pre-trained with supervised learning and not further fine-tuned with PPO. **JoTR$_{pretrained}$** uses a pre-trained BERT as the context encoder but with the weights fixed.

7.2. Main Results

The learning curve depicted in Figure 3 shows the superior performance and efficiency of the proposed JoTR model. Notably, JoTR outperforms the previous SOTA model JOIE (0.93 vs 0.91 for MultiWOZ and 0.79 vs 0.51 for SGD) despite only being trained with 50K frames, compared to JOIE's 400K frames (Wang and Wong, 2021). Notice that JOIE only obtained a success rate of 0.55 when trained with 50K frames similar to JOTR, demonstrating a significant improvement in training efficiency. Moreover, the ability of JoTR to improve significantly with such short training makes it more suitable for real-world applications.

Table 1 shows that JoTR requires significantly fewer turns than JOIE to satisfy the user goal. This efficiency can be attributed to JoTR's capacity to generate multiple atomic actions in one turn, in contrast to JOIE's single-action prediction. This characteristic not only reduces the total number of interaction turns but also enhances JoTR's practicality for everyday use.

All JoTR variants outperform MLP$_{ppo}$. Notably, JoTR$_{w/o\ ppo}$ surpasses MLP$_{ppo}$ without additional RL fine-tuning, suggesting that the robust learning capacity of the Transformer is effectively utilized to learn the specific structural properties of dialogue actions. Compared to SimpleTOD, JoTR exhibits superior performance due to its use of dialogue actions as input, which reduces noise and complexity compared to SimpleTOD's language utterances. Moreover, JoTR demonstrates significantly more diversity in the dialogue actions generated, as evidenced by the distinct actions. It generated 249 and 494 different actions in MultiWOZ and SGD respectively, which is 34% and 37% more dialogue actions than SimpleTOD, the most diverse among the previous models. This indicates that JoTR has a profound understanding of atomic dialogue actions, enabling it to combine them effectively to handle unseen dialogue scenarios. Furthermore, JoTR's

Figure 4: The dialogue example provided illustrates a user goal that requires: 1) querying the address, postcode, and phone number of a park located in the east; 2) booking a guesthouse with parking service located in the center for 8 people on Saturday, staying for 5 days, and asking for the phone number; 3) calling a taxi to arrive at 18:15 and requesting information about the car type and phone number. User utterances are in italics, while agents' responses are highlighted in bold. Boxes in different colors are used for different models. The system's dialogue actions (text on the right) highlighted in yellow underscore JoTR's ability to manage complex and out-of-domain user actions, a task that other models find challenging. The dialogue actions highlighted in pink demonstrate JoTR's skill in proactively providing relevant slots, an area where other models tend to underperform.

encoder-decoder model structure is more effective at capturing context information than SimpleTOD's decoder-only model. Lastly, JoTR significantly outperforms ChatGPT. Most errors made by ChatGPT fall into two categories: 1. hallucination on domain, slot, and values, and 2. violations of output format constraints.

7.3. Ablation Study

7.3.1. The Effectiveness of Reward Shaping

Applying reward shaping improves the success rate from 0.89 to 0.93 in MultiWOZ, and from 0.72 to 0.79 in SGD. This improvement is clearly demonstrated when comparing JoTR without reward shaping ($\text{JoTR}_{\text{w/o rs}}$) and JoTR. It indicates that reward shaping significantly contributes to enhancing the success rate. As Figure 3 illustrates, JoTR maintains a higher success rate than $\text{JoTR}_{\text{w/o rs}}$ throughout the fine-tuning process. Additionally, without reward shaping, the generated dialogue actions are 17% and 13% less diverse in MultiWOZ and SGD, respectively. It suggests that the reward shaping encourages the policy to explore more actions. Overall, these performance differences highlight the advantage of using a dense and well-designed reward in RL fine-tuning, consistent with previous findings (Wang et al., 2022).

7.3.2. The Necessity of RL Fine-Tuning

$\text{JoTR}_{\text{w/o ppo}}$ markedly underperforms without being further fine-tuned with RL, exhibiting a reduction of up to 28% in success rate on MultiWOZ. The diversity of generated actions also diminishes notably in the absence of RL fine-tuning, showing a 27% decrease in unique dialogue actions. This is expected since the training set only covers a limited amount of dialogue actions, thus the policy does not have the chance to explore a broader action set compared to those fine-tuned with RL. This underlines the critical role of RL fine-tuning in refining the behavior of the policy model through the reward signal.

7.3.3. Importance of Training from Scratch

JoTR significantly outperforms $\text{JoTR}_{\text{pretrained}}$, achieving a success rate of 0.93 versus 0.76 in Multiwoz and 0.79 versus 0.64 in SGD. As evidenced by Figure 3, $\text{JoTR}_{\text{pretrained}}$ exhibits a notably lower success rate initially. This can be attributed to the distinct structure of the dialogue actions' input space, which markedly differs from the natural language space where the model was originally pre-trained. This discrepancy cannot be bridged effectively by supervised training during the warm-up phase or Reinforcement Learning in the fine-tuning phase.

Model	MultiWOZ				SGD			
	Succ.↑	*Turn*↓	*Rew.*↑	*#Acts*↑	*Succ.*↑	*Turn*↓	*Rew.*↑	*#Acts*↑
JOIE	0.55	18.90	40.82	147	0.51	11.10	15.32	210
MLP_{ppo}	0.56	30.72	-26.76	162	0.54	23.43	16.50	233
SimpleTOD‡	0.62	-	-	186	0.50	-	-	361
DASP‡	0.85	-	-	-	0.70	-	-	-
ChatGPT	0.73	13.10	41.05	165	0.50	**11.04**	15.48	242
JoTR	**0.93**	**9.94**	**68.46**	**249**	**0.79**	15.23	**49.25**	**494**
$\text{JoTR}_{\text{w/o rs}}$	0.89	9.95	66.42	207	0.72	16.53	38.84	429
$\text{JoTR}_{\text{w/o ppo}}$	0.67	18.44	32.18	189	0.55	24.76	14.62	357
$\text{JoTR}_{\text{pretrained}}$	0.76	14.19	44.87	195	0.64	19.25	28.18	372

Table 1: Performance of various models in dialogue act modeling on MultiWOZ and SGD datasets, assessed using metrics: success rate (*Succ.*), average turns (*Turn*), average (*Rew.*), and distinct actions (*#Acts*). from the original paper are underscored. Models marked with ‡ are evaluated using a test corpus; the highest scores are highlighted in bold.

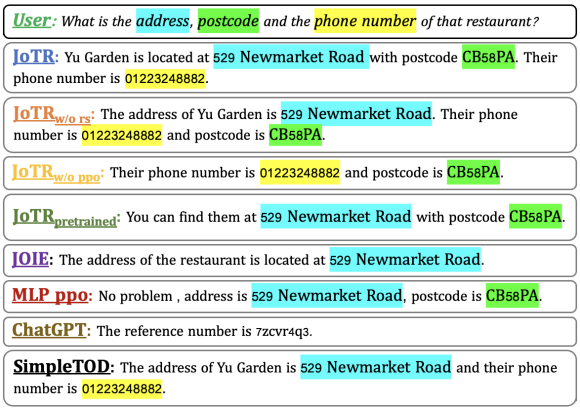

Figure 5: This figure presents a comparison of various responses to the same user query. In this instance, the user is seeking information on three slots: address, postcode, and phone number. We use distinct colors to emphasize the parts of the response that correspond to the user's requested slots. Upon examination, we find that only JoTR and JJoTR$_{w/ors}$ inform all the slots requested by the user.

7.4. Case Study

As demonstrated in Figure 5, we observe that when the user requests a slot combination that is never seen in the training set, both JoTR and $\text{JoTR}_{\text{w/o rs}}$ can inform all requested slots successfully. This demonstrates their robust ability to generate effective and efficient dialogue actions. In contrast, $\text{JoTR}_{\text{w/o ppo}}$, $\text{JoTR}_{\text{w/o pretrained}}$, and SimpleTOD were unable to inform all requested slots, potentially a reflection of their inferior performance relative to JoTR. JOIE only informed one requested slot, likely due to its design limitation of generating a single action per turn. Moreover, MLP_{ppo} could not inform the complete slots as well, since the dialogue action for informing address, postcode, and phone number is not found within its predefined action set. Lastly, ChatGPT responded inappropriately, for reasons elaborated in the preceding section.

We also provide a full dialogue example of various models interacting with the user simulator in Figure 4. The user requested the address, postcode, and phone of the park in the second turn. There is not a single training example that requests these three slots simultaneously, showcasing the ability of different models to respond to complex and out-of-domain user actions. Consistent with Figure 5, JoTR and $\text{JoTR}_{\text{w/o rs}}$ were able to inform all three slots while other models can't (highlighted in yellow). Furthermore, when the user requested a guesthouse in turn three, JoTR is able to provide the phone number without being explicitly asked while $\text{JoTR}_{\text{w/o rs}}$ and other models failed to do so as (highlighted in pink). This illustrates that rewarding shaping can incentivize the model to provide additional information preemptively. In this example, we can also see the dialogues of other models are significantly longer than those of JoTR. Therefore, JoTR is able to achieve the user's goal efficiently.

7.5. Human Evaluation

Model	Succ.(MultiWOZ)↑	Succ.(SGD)↑
JOIE	0.56	0.53
MLP_{ppo}	0.52	0.56
SimpleTOD	0.62	0.50
DASP	-	-
ChatGPT	0.66	0.52
JoTR	**0.92**	**0.76**
$\text{JoTR}_{\text{w/o rs}}$	0.84	0.70
$\text{JoTR}_{\text{w/o ppo}}$	0.66	0.56
$\text{JoTR}_{\text{pretrained}}$	0.68	0.60

Table 2: Human evaluation results. We use the models trained with 50K frames for all agents.

We further conduct a human evaluation to validate the simulation results using the models trained

with 50K frames. We recruited 3 volunteer student helpers as evaluators to interact with different models. For each model, we held 50 dialogue sessions. In each session, an evaluator is assigned a randomly selected model and user goal. The evaluators are instructed to interact with the model in accordance with the user goal, with a maximum of 20 turns per session, aligning with the settings used in the experiments in previous sections. At the end of each session, the evaluators assessed the success or failure of the dialogue. The results are illustrated in Table 2, which are consistent with the previous results using a user simulator.

8. Conclusion

We introduced JoTR, a versatile framework for dialogue policy learning using joint text-to-text Transformer Reinforcement Learning. It trains token-grained policies that can generate dialogue actions without the need for predefined templates. Empirical results from two benchmark datasets show that our model, which does not rely on predefined action templates, outperforms the strongest baseline in terms of both policy learning efficiency and dialogue action quality as determined by simulated and human evaluations.

9. Acknowledgements

This research work is partially supported by CUHK direct grant No. 4055209 and CUHK Knowledge Transfer Project Fund No. KPF23GWP20.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Iñigo Casanueva, Lina M. Rojas-Barahona, and Milica Gašić. 2017. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 86–92, Saarbrücken, Germany. Association for Computational Linguistics.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Milica Gašić, Filip Jurčíček, Simon Keizer, Francois Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of POMDP-based dialogue managers. In *Proceedings of the SIGDIAL 2010 Conference*, pages 201–204, Tokyo, Japan. Association for Computational Linguistics.

Christian Geishauser, Carel van Niekerk, Hsien-chin Lin, Nurul Lubis, Michael Heck, Shutong Feng, and Milica Gašić. 2022. Dynamic dialogue policy for continual reinforcement learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 266–284, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 10749–10757.

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.

Megha Jhunjhunwala, Caleb Bryant, and Pararth Shah. 2020. Multi-action dialog policy learning with interactive human teaching. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 290–296, 1st virtual meeting. Association for Computational Linguistics.

Giovanni Yoko Kristianto, Huiwen Zhang, Bin Tong, Makoto Iwayama, and Yoshiyuki Kobayashi. 2018. Autonomous sub-domain modeling for dialogue policy with hierarchical deep reinforcement learning. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 9–16, Brussels, Belgium. Association for Computational Linguistics.

Wai-Chung Kwan, Hong-Ru Wang, Hui-Min Wang, and Kam-Fai Wong. 2023. A survey on recent advances and challenges in reinforcement learning methods for task-oriented dialogue policy learning. *Machine Intelligence Research*, pages 1–17.

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*, pages 1296–1303, Punta Cana, Dominican Republic. Association for Computational Linguistics.

E. Levin, R. Pieraccini, and W. Eckert. Learning dialogue strategies within the markov decision process framework. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 72–79.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*.

Ziming Li, Julia Kiseleva, and Maarten de Rijke. 2020a. Rethinking supervised learning and reinforcement learning in task-oriented dialogue systems. *arXiv preprint arXiv:2009.09781*.

Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Shahin Shayandeh, and Jianfeng Gao. 2020b. Guided dialog policy learning without adversarial learning in the loop. *arXiv preprint arXiv:2004.03267*.

Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. 2016. Efficient exploration for dialog policy learning with deep bbq networks & replay buffer spiking. *CoRR abs/1608.05081*.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Yun-Nung Chen, and Kam-Fai Wong. 2018. Adversarial advantage actor-critic model for task-completion dialogue policy learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6149–6153. IEEE.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240, Copenhagen, Denmark. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347 [cs]*.

Satinder Singh, Michael Kearns, Diane Litman, and Marilyn Walker. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Pei-Hao Su, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Online active reward learning for policy optimisation in spoken dialogue systems. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2431–2441, Berlin, Germany. Association for Computational Linguistics.

Pei-Hao Su, David Vandyke, Milica Gašić, Nikola Mrkšić, Tsung-Hsien Wen, and Steve Young. 2015. Reward shaping with recurrent neural networks for speeding up on-line policy learning in spoken dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 417–421, Prague, Czech Republic. Association for Computational Linguistics.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.

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)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. 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)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Da Tang, Xiujun Li, Jianfeng Gao, Chong Wang, Lihong Li, and Tony Jebara. 2018. Subgoal discovery for hierarchical dialogue policy learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2298–2309, Brussels, Belgium. Association for Computational Linguistics.

Hongru Wang, Huimin Wang, Zezhong Wang, and Kam-Fai Wong. 2022. Integrating pretrained language model for dialogue policy evaluation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6692–6696. IEEE.

Huimin Wang, Wai Chung Kwan, and Kam-Fai Wong. 2023. Dialog action-aware transformer for dialog policy learning. In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 142–148, Prague, Czechia. Association for Computational Linguistics.

Huimin Wang, Baolin Peng, and Kam-Fai Wong. 2020. Learning efficient dialogue policy from demonstrations through shaping. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6355–6365, Online. Association for Computational Linguistics.

Huimin Wang and Kam-Fai Wong. 2021. A collaborative multi-agent reinforcement learning framework for dialog action decomposition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7882–7889, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zheng Zhang, Lizi Liao, Xiaoyan Zhu, Tat-Seng Chua, Zitao Liu, Yan Huang, and Minlie Huang. 2020. Learning goal-oriented dialogue policy with opposite agent awareness. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 122–132, Suzhou, China. Association for Computational Linguistics.

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*, pages 13–22, Dublin, Ireland. Association for Computational Linguistics.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. *arXiv preprint arXiv:2002.04793*.