Does Reinforcement Learning from Human Feedback Framework Still Work for Task-Oriented Dialogue Systems?

Anonymous ACL submission

Abstract

The paradigm of reinforcement learning from human feedback (RLHF) after supervised fine-tuning (SFT) language models has become widespread. In this work, we investigate whether RLHF with turn-level preferences is still effective in task-oriented dialogue (TOD) task that requires dialog-level rewards. Since there is no human preference dataset for TOD task, we develop two synthetic feedback generation methods for fully annotated or partially annotated TOD dataset. We compare these two methods to the corresponding SFT methods in an online environment where user goals are unknown. Despite the simplicity of the proposed methods, RLHF outperformed SFT on the partially annotated TOD dataset in both corpus-based and simulator-based evaluations. Our comprehensive experiments present a direction for effectively enhancing system performance using data generated while providing services in real-world environments.

1 Introduction

Task-oriented dialogue (TOD) systems are developed to help users achieve specific goals by interacting with them. These systems are composed of four key components: (1) natural language understanding module to comprehend the user's intent, (2) dialog state tracking module to summarize the dialog history into a dialog state, (3) policy module to decide the strategy by referring to the dialog state and external resources (i.e., database), and (4) natural language generation module to generate natural language responses based on the policy. With the advancement of pre-trained language models (PLMs), integrating PLMs into TOD systems has enhanced their performance. On the one hand, it has often been found that the responses from PLMs do not always reflect human preferences (Schramowski et al., 2022; Korbak et al., 2023). To address this issue, (Ouyang et al., 2022) proposed a procedural learning framework, known as reinforcement learning from human feedback (RLHF), which consists of supervised fine-tuning (SFT), reward modeling (RM), and reinforcement learning using proximal policy optimization (PPO) (Schulman et al., 2017).

With a reward model trained on the human preference dataset, it applies reinforcement learning on online data to align PLMs with human values. This is in line with improving the performance of the TOD systems in real-world scenarios where the user goals are unknown. However, there are challenges in applying RLHF to TOD systems. RLHF provides rewards based on the appropriateness of the model's response at each turn, whereas TOD systems require rewards to be given not only for individual turns but also for the appropriateness of the entire dialogue. Additionally, applying RLHF to TOD systems by manually building preference datasets demands significant effort and annotation costs. For these reasons, there has been no research investigating the effectiveness of RLHF in TOD systems, despite the fact that it can be applied to TOD systems.

In this paper, we investigate whether RLHF is effective for TOD systems, focusing on the policy optimization and the response generation task. To this end, we propose two methods to apply RLHF to TOD systems without the human preference annotation by using a rule of thumb to estimate the human preference. These two methods are based on assumptions that either (1) humans are better than models or (2) in-distribution models are better than out-distribution models. We compare these methods to corresponding fine-tuning method for TOD dataset where the TOD annotations (e.g., belief states and system actions) are either partially provided or fully provided. In the real-world scenarios where the user goals are unknown, our experimental results show that RLHF improves the performance of TOD systems when the TOD dataset

is partially annotated.

2 Related Work

Task-Oriented Dialogue The next token prediction task requires annotated TOD data and does not guarantee to generate diverse and informative responses (Zhang et al., 2020). To overcome these limitations, reinforcement learning (RL) has been studied for policy optimization and response generation, which can be formulated as a Markov Decision Process. The previous works can be categorized into three different approaches: offline RL (Zhao et al., 2019; Ramachandran et al., 2022), model-based RL (Peng et al., 2018; Wu et al., 2020), and multi-agent RL (Papangelis et al., 2019; Takanobu et al., 2020).

Our work falls into model-based RL approach where a user goal is not required to be annotated for online data. To the best our knowledge, the model-based RL methods have only been focused on the policy optimization, whereas we investigate the preference-based reward modeling for both policy optimization and response generation. One of our reward modeling methods, preferring the responses of in-distribution model, is similar to (Ramachandran et al., 2022). They train a turn-level reward model with a preferential objective function using K models trained by K-folded datasets. However, their reward model cannot be used for online data because it requires the user goal that is hard to obtain in the real-world scenarios.

Reinforcement Learning from Human Feedback RLHF has been proposed to reflect human's preferences to the language models (Ouyang et al., 2022; Ziegler et al., 2019). It is mainly studied for large language models with preference datasets annotated by human (Bai et al., 2022; Ethayarajh et al., 2022). While it has been reported that RLHF works on smaller models (like GPT-2) with the large human preference dataset , there is still a burden in collecting the human preference dataset. Fortunately, synthetic feedback that does not rely on human annotator has been shown to be able to reflect human preferences (Kim et al., 2023). Inspired by this work, we design criteria to generate synthetic feedback and then investigate their usefulness for TOD.

3 Method

In this section, we explain the entire framework depicted in Figure 1, following the RLHF process. First, SFT is conducted to enhance the PLMs' performance for specific data. Next, RM is performed using the preference dataset. Finally, through the RL algorithm, such as PPO, the model's parameters are adjusted to maximize the rewards from the reward model, with kullback-leibler (KL) regularization to prevent mode-collapse.

3.1 Dataset Partitioning and Supervised Fine-Tuning

We have adopted MultiWOZ 2.1 (Eric et al., 2020), containing over 10,000 dialogues across multiple domains. In order to consider both fully annotated and partially annotated datasets, the entire dataset is divided into 4-fold splits, named D_0, D_1, D_2, and D_3, respectively. In our experiments, the subset D_0 is used for SFT while other subsets are used in the process of constructing the preference dataset for RM. The experiments on either partially annotated or fully annotated TOD dataset depend on whether the subsets D_1, D_2, and D_3 are annotated with the system actions and the belief states.

3.2 Synthetic Preference Dataset Construction

To construct the preference dataset without human annotators, we leverage insights from previous studies, which have shown that humans achieve superior results in conversation compared to language models (Ou et al., 2023), and models fine-tuned on specific data perform better on in-distribution data (Lee et al., 2019). Based on these findings, we apply a rule of thumb to estimate the quality of responses as follows:

- Human $>$ Machine
- In-distribution model$>$Out-distribution model

The comparison between human and machine responses is utilized in the experiment where the TOD dataset is partially annotated, while the comparison between in-distribution model and out-distribution model is utilized in the experiment where the TOD dataset is fully annotated. We present experimental results supporting this assumption in Appendix A.

Human $>$ Machine Based on the assumption that human responses will achieve higher preferences than language model responses, the model M_0, fine-tuned on the subset D_0, generates outputs D_1^{model}, D_2^{model}, and D_3^{model} for D_1, D_2, and D_3 respectively. These outputs are then compared with the human-crafted original data D_1, D_2, and

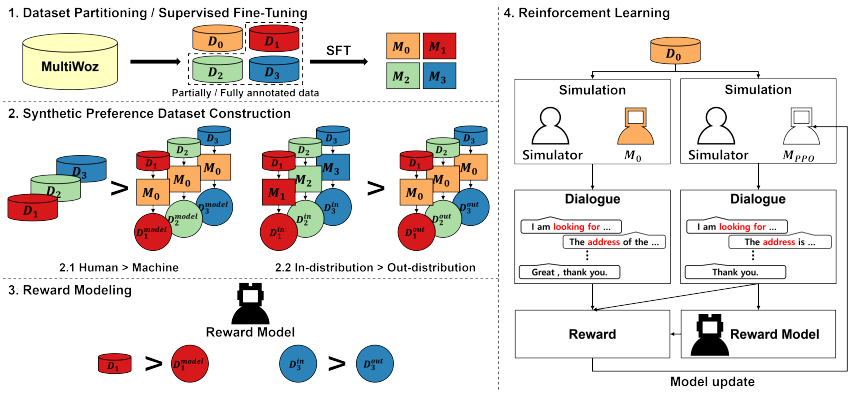

Figure 1: Overview of our proposed method. In this figure, M_n is the model fine-tuned on subset D_n. D_n^{model} and D_n^{out} represent the outputs generated by M_0 for D_n, while D_1^{in} is the output generated by M_n for D_n.

D_3. The preference between each pair of dataset is established as $D_n > D_n^{model}$ (where n=1, 2, 3). We designate D_n as positive responses and D_n^{model} as negative responses, forming binary pairs such as (D_1, D_1^{model}), (D_2, D_2^{model}), and (D_3, D_3^{model}). Since the annotations for TOD task are only needed to perform SFT, this method is applicable in semi-supervised environment where the annotations are partially provided.

In-distribution model>Out-distribution model To realize the assumption that language models trained on the distribution of specific divided dataset will perform better than those not specifically trained, we use the model M_n, fine-tuned on each subset D_n. The model M_0 then generates outputs D_1^{out}, D_2^{out}, and D_3^{out} for D_1, D_2, and D_3, respectively. Other models generate outputs corresponding to the trained subsets. These are referred to as D_1^{in}, D_2^{in}, and D_3^{in}. Here, the preference is established by treating D_n^{in} as positive responses, and D_n^{out} as negative responses. The final preference dataset forms binary pairs like (D_1^{in}, D_1^{out}), (D_2^{in}, D_2^{out}), and (D_3^{in}, D_3^{out}). In this method, the annotations for TOD task are needed to fine-tune the models for each subset.

3.3 Reward Modeling

In the process of RM, we utilize the positive and negative pairs generated from the previous step. The reward model may struggle to learn an ambiguous preference because the preference dataset is synthetically constructed. To encourage model discreteness, we inject a noise N into the loss function (Jang et al., 2016). Intuitively, the reward model may tend to avoid local minima, where the outputs of the sigmoid function are mostly 0.5, due to the noise. Here, we use the Gaussian noise with a standard deviation of 0.25, since we found that the Gaussian noise is better than Gumbel noise in our preliminary experiments. The details of the objective function are provided in Appendix B.1.

3.4 Reinforcement Learning

To simulate the real-world scenarios where the users interact with TOD systems, we adopt a user simulator, provided by ConvLab-2 (Zhu et al., 2020), as a real user. The user simulator is composed of a rule-based policy module and a template-based response generation module. To prevent goal contamination, where test user goals are included in the training data of the RL stage, we sample 1K goals from the training data. We then collect the dialogues using the sampled goals by interacting the fine-tuned TOD model M_0 with the user simulator. Note that the user goals are not used in the RL stage. For this online data, we train the fine-tuned TOD model M_0 using PPO. The objective function is described in Appendix B.2.

4 Experiments

We investigate the effectiveness of RLHF on TOD systems in the real-world scenarios, given two different data annotation conditions. When the annotations are fully provided in TOD dataset, the preference dataset can be built from the assumption that in-distribution models are better than out-distribution models. Based on this assumption, the model trained with PPO is called M_{ppo}^{dist}. It is comparable to the model supervised fine-tuned on the fully annotated data (called M_{full}) and the model further supervised fine-tuned on the online data (called M_{full}^{online}). On the other hand, when the annotations are partially provided in TOD dataset, the preference dataset can be built from the assumption

Model	Noise	Corpus-based Evaluation				Simulator-based Evaluation		
		Match	Success	BLEU	Combined score	Success rate	Completeness	Turns
M_{full}*	-	89.4	82.4	18.0	103.9	95.2	96.6	7.1
M_{full}^{online}	-	77.3	67.5	10.3	82.7	96.8	97.7	7.6
M_{ppo}^{dist}	-	86.7	79.7	16.1	99.3	96.1	98.6	7.3
	O	89.4	82.8	15.0	101.1	93.6	99.0	6.7
M_0	-	85.1	76.6	16.3	97.1	95.4	97.0	7.6
M_0^{online}	-	76.3	65.1	9.9	80.6	98.0	98.9	7.6
M_{ppo}^{human}	-	87.2	79.9	15.9	99.5	96.2	98.9	7.2
	O	87.2	80.3	15.9	99.6	95.8	98.4	7.2

Table 1: Experimental results with corpus-based and simulator-based evaluations. In the original paper, UBAR achieved scores of 92.7 / 81.0 / 16.7 / 103.6 for the corpus-based evaluation. We set our better model M_{full} as a baseline from combined score perspective.

that humans are better than models. We call the model, trained with PPO under this assumption, M_{ppo}^{human}. As mentioned in section 3, the dataset is divided into 4-fold splits, of which only D_0 is assumed to be annotated for the belief states and the system actions. We compare M_{ppo}^{human} to the model supervised fine-tuned on the D_0 (called M_0) and the model further supervised fine-tuned on the online data (called M_0^{online}). The details of our implementation, dataset, and evaluation metrics can be found in Appendix C.

4.1 Experimental Results

Table 1 shows the results for corpus-based and simulator-based evaluations. We will describe the main results and provide discussion about them.

RLHF vs. SFT In fully supervised setting, the RLHF-applied models M_{ppo}^{dist} show lower performance in the corpus-based evaluation compared to the SFT model M_{full}. In semi-supervised setting, in contrast, the RLHF-applied models M_{ppo}^{human} achieve better performance than the SFT model M_0. This indicates the effectiveness of applying RLHF to TOD systems when only a subset of the entire TOD dataset is annotated. Additionally, we observed that the RLHF-applied models consistently outperform the SFT models in the simulator-based evaluation, except for M_{ppo}^{dist} with noise.

Effect of noise In the corpus-based evaluation, the noise injection is helpful for both settings, but there is no significant gain in semi-supervised setting. This may be consistent with our observation in RM stage, that distinguishing between human and model responses are easier than distinguishing between in-distribution and out-distribution model responses. Because it is more ambiguous to differentiate the preference between models, the noise can be more helpful in this case.

RLHF vs. SFT on online data Further fine-tuning of the SFT models on online data improves performance on simulator-based evaluation, but significantly degrades performance on corpus-based evaluation. We conjecture this phenomenon comes from overfitting to the user simulator. This can be detrimental to serving the dialogue systems in real-world environments where the online data is utilized for the sustainability of the systems.

5 Conclusion

We have explored whether RLHF can improve TOD systems when the user goals and additional annotations are not provided, as in real-world scenarios. Furthermore, two preferential criteria are presented to unburden the cost of annotating human preference. One, which favors human responses over model responses, is applied to the partially annotated TOD dataset, and the other, which favors in-distribution models over out-distribution models, is applied to the fully annotated TOD dataset. Our experiments demonstrate that RLHF is effective for TOD systems in the semi-supervised setting, rather than fully supervised setting. RLHF may be better suited for maintaining and enhancing system performance by leveraging data generated in real-world environments. We hope that our research can inspire more future works on applying RLHF to TOD systems.

6 Limitations

Our work has limitations in that it completely excluded the DST task and examined one method of RLHF. Although the models that are further supervised fine-tuned on the online data perform better in the simulator-based evaluation, they seem to suffer from a forgetting problem. Additionally, while noise injection is helpful for both settings in the corpus-based evaluation, there is no significant gain in the semi-supervised setting, and the noise-injected models perform worse in the simulator-based evaluation. It is necessary to investigate whether similar phenomena occur with various RLHF methods (Rafailov et al., 2024; Azar et al., 2023). Addressing these limitations could provide a more comprehensive understanding of how RLHF can be applied to TOD systems.

References

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. 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*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

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

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Yoo, and Minjoon Seo. 2023. Aligning large language models through synthetic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13677–13700, Singapore. Association for Computational Linguistics.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR.

Kenton Lee, Jacob Devlin, Ming-Wei Chang, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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

Jiao Ou, Junda Lu, Che Liu, Yihong Tang, Fuzheng Zhang, Di Zhang, Zhongyuan Wang, and Kun Gai. 2023. Dialogbench: Evaluating llms as human-like dialogue systems. *arXiv preprint arXiv:2311.01677*.

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.

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*, pages 92–102.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn.

2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Govardana Sachithanandam Ramachandran, Kazuma Hashimoto, and Caiming Xiong. 2022. [CASPI] causal-aware safe policy improvement for task-oriented dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 92–102, Dublin, Ireland. Association for Computational Linguistics.

Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A Rothkopf, and Kristian Kersting. 2022. Large pre-trained language models contain human-like biases of what is right and wrong to do. *Nature Machine Intelligence*, 4(3):258–268.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

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*, pages 625–638.

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*, pages 854–863.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218, Minneapolis, Minnesota. 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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Experimental results

Table 2 compares the performance between in-distribution and out-distribution through the corpus-based evaluation. The results indicate that models trained on in-distribution data achieved better performance across all metrics compared to those trained on out-distribution data. This supports the reliability of our experiments conducted under the assumption that in-distribution models are better than out-distribution models.

B Mathematical Formulations

B.1 Reward Modeling

The loss function for the reward model is defined as follows:

$$J(\theta) = -E_{(c,y_{pos},y_{neg})\sim D} [log(\sigma(r_\theta(y_{pos}|c) - r_\theta(y_{neg}|c)) + N)] \quad (1)$$

In this equation, c, y_{pos}, and y_{neg} represent a dialogue context, positive response, and negative response, respectively. The injected noise N encourages the model to avoid local minima by making the outputs of the sigmoid function less likely to be mostly 0.5. Gaussian noise with a standard deviation of 0.25 is used, as it was found to be more effective than Gumbel noise in preliminary experiments.

B.2 Reinforcement Learning

The objective function for the RL stage is defined as follows:

$$J(\phi) = -E_{(c,y)\sim D_{online}} [r_\theta(y|c) - \beta D_{\mathrm{KL}}[\pi_\phi(y|c)||\pi_0(y|c)]] \quad (2)$$

In this equation, β is a coefficient for KL regularization, and π_ϕ and π_0 are the trainable model policy and the initial model (M_0) policy, respectively. The reward $r_\theta(y|c)$ is provided by the reward model trained with the synthetic preference dataset. For all experiments, we set β to 0.2.

Model	Data	Corpus-based Evaluation			
		Match	Success	BLEU	Combined score
M_0	D_1	71.0	57.8	16.5	80.9
M_1	D_1	**74.4**	**63.7**	**24.5**	**93.6**
M_0	D_2	71.1	59.2	16.4	81.6
M_2	D_2	**74.0**	**63.6**	**25.1**	**93.9**
M_0	D_3	69.8	58.7	16.5	80.8
M_3	D_3	**74.4**	**64.6**	**24.5**	**94.0**

Table 2: Performance Comparison Based on Distributions

C Experiment Configurations

C.1 Implementation details

We use UBAR (Yang et al., 2021) which is one of state-of-the-arts in TOD systems for all models. In SFT stage, we use DistilGPT2[1], which has 82M parameters, as a backbone. We train the SFT models for 15 epochs with a batch size of 16 and a learning rate of 1e-4. We further fine-tune these models on the online data for 3 epochs with the same configuration. In RM stage, all reward models are initialized with the fine-tuned model M_0. And then, the models are trained for 5 epochs. We use 10% of the preference dataset as a development set for model selection. In PPO stage, all models are equipped with a rule-based DST provided by ConvLab-2 to focus on the policy optimization and the response generation. The batch size is set to 32 and the learning rate is set to 1e-6. In this stage, all models are also initialized with M_0, and trained for only one epoch. For all experiments, we use AdamW optimizer and linear scheduler without warmup. The experiments were consistently performed on a 48G Quadro RTX 8000.

C.2 Dataset

MultiWOZ (Budzianowski et al., 2018), collected through human-to-human interactions, is an open-source dataset extensively used for examining the TOD systems. This dataset encompasses TOD dialogues for a single domain and multiple domains across 7 domains (hotel, hospital, attraction, train, restaurant, policy, and taxi). It also provides train/dev/test splits for 8,434/1,000/1,000 dialogues. We follow the provided split and the pre-processing procedures described in DAMD-MultiWOZ[2].

[1]https://huggingface.co/distilbert/distilgpt2

[2]https://github.com/thu-spmi/damd-multiwoz

C.3 Evaluation Metric

We evaluate our method using two approaches: corpus-based evaluation, where we assess the model responses to user utterances in the corpus given the ground-truth belief states, and simulator-based evaluation, where we use the rule-based user simulator to simulate the interaction between real user and dialogue system. The corpus-based evaluation includes **Match**, measures whether the dialogue system has provided the correct entity, **Success** evaluates whether the dialogue system has provided the correct entity and has fully answered all the information requested by the user, and **BLEU** (Papineni et al., 2002), which measures how similar the generated responses are to human responses. We also report **Combined score** (Mehri et al., 2019), which is calculated as, (Match + Success) * 0.5 + BLEU, to assess the overall quality of the dialogue system. For the simulator-based evaluation, **Success Rate** measures whether the dialogue system has provided not only the correct entity but also all the information requested by the user including booking information if available, **Completeness** measures whether user has fulfilled for own goal, and **Turns** means the average of the number of turns of the simulated dialogues. We report the average of five runs for the metrics of simulator-based evaluation, because the user simulator has diverse policies unlike corpus-based evaluation. All simulations have been conducted for the user goals in the test split of MultiWOZ.