Received: 1 August 2025 | Revised: 16 March 2026 | Accepted: 4 May 2026

DOI: 10.4218/etrij.2025-0313

ETRI Journal WILEY

ORIGINAL ARTICLE

Evaluating reinforcement learning from human feedback for task-oriented dialogue systems

Hyeok-Min Gwon[1] | **Yohan Lee**[1] | **Jin-Xia Huang**[1,2] | **Jonghyuk Lee**[3]

[1]Language Intelligent Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

[2]ETRI School, University of Science and Technology (UST), Daejeon, Republic of Korea

[3]ITS Division, Korea Expressway Corporation (KEC), Gimcheon, Republic of Korea

Correspondence
Yohan Lee, Language Intelligent Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea.
Email: carep@etri.re.kr

Funding information
This work was supported by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners).

Summary

Reinforcement learning from human feedback (RLHF) has shown strong potential for aligning language models, but its role in task-oriented dialogue (TOD) remains unclear. In TOD, models are typically trained with local turn-level supervision, while system behavior is evaluated through broader interaction-level properties. This mismatch becomes more challenging in online settings, where explicit dialogue-level rewards and human preference annotations are unavailable. In this work, we study whether RLHF can be usefully applied to TOD under this limitation. We consider two task-annotation regimes, partially annotated and fully annotated TOD data, and construct pseudo-preference pairs using empirical ranking heuristics motivated by prior work on synthetic feedback and model-based ranking signals. We then train reward models on the constructed pairs and optimize dialogue policies with Preference policy optimization (PPO) using simulator-generated online trajectories. Experiments on MultiWOZ 2.1 show that the proposed RLHF approach consistently improves corpus-based evaluation over supervised baselines, while simulator-based effects remain mixed and backbone-dependent.

KEYWORDS
deep learning, machine learning, natural language processing, reinforcement learning from human feedback, task-oriented dialogue

1 | INTRODUCTION

Task-oriented dialogue (TOD) systems are designed to help users accomplish specific goals through multi-turn interactions. Recent end-to-end TOD systems based on pretrained language models have achieved strong benchmark performance by treating dialogue modeling as a unified generation problem [1–4]. However, their training objectives remain largely local. In most end-to-end TOD systems, the model is trained to predict intermediate representations and the next system response at each turn. By contrast, system quality is judged over the course of the full interaction, for example, by whether the system completes the task efficiently and avoids unnecessary loops. Moreover, even under the same dialogue context and user goal, a given turn can still admit multiple appropriate system responses [5, 6]. This makes

H-M. Gwon and Y. Lee equally contributed to this work.

This study was conducted while the author, J. Lee, was affiliated with ETRI.

it difficult to assume that a single supervised target is sufficient for learning robust dialogue behavior.

This mismatch becomes particularly important in online settings, where the system must act without access to future turns and where dialogue-level rewards are not directly available during interaction. In such settings, improving turn-level generation quality does not necessarily provide a reliable training signal for broader interaction-level behavior. Prior work on dialogue reinforcement learning has also shown that obtaining reliable user-level reward signals is itself a non-trivial problem, even when explicit user ratings are available [7]. More broadly, TOD research has explored latent-action models, model-based critics, and multi-agent optimization in order to better connect local decisions with interaction-level outcomes [8–12]. These studies collectively suggest that a central difficulty in TOD is not only generating plausible next turns, but also learning a training signal that better reflects interaction-level behavior under partial or weak supervision.

From this perspective, we use reinforcement learning from human feedback (RLHF) to test whether reward learning from constructed pairwise comparisons can provide a useful optimization signal for TOD when explicit dialogue-level rewards and human preference annotations are unavailable. This is particularly relevant in our setting because direct supervised adaptation to online trajectories may be insufficient for robust optimization, while RLHF provides a mechanism for learning a reward signal from relative comparisons between candidate outputs. At the same time, applying RLHF to TOD is not straightforward. TOD does not provide an established human preference dataset for reward modeling, and directly eliciting preference information from users remains expensive and task-specific even in assistant settings [13]. Moreover, even when turn-level comparison signals can be constructed, it is not obvious how such signals will affect broader interaction-level behavior under online interaction.

To study this question, we investigate whether constructed pairwise comparison signals can support RLHF in TOD when explicit human preference annotations are unavailable. Our study considers two task-annotation regimes, partially annotated TOD data and fully annotated TOD data. Here, fully annotated does not mean that human preference labels are available. Rather, it indicates that the dataset includes human-provided structured TOD annotations, such as belief states and system actions, which are costly and time-consuming to obtain. To evaluate whether our framework remains applicable when such annotations are unavailable, we also consider a partially annotated regime. In both regimes, we construct pseudo-preference pairs using empirical ranking heuristics motivated by prior work on synthetic feedback and model-based ranking signals [14, 15].

Based on these constructed pairs, we train reward models and optimize the initial dialogue policy on simulator-generated online trajectories. We also compare this reward-guided framework with an online-only supervised fine-tuning baseline trained on the same interaction trajectories. This comparison allows us to examine whether RLHF provides a more effective use of online interaction data than direct supervised adaptation alone, while explicitly evaluating both corpus-based and online interaction settings.

In this paper, we study whether synthetic preference pairs constructed under partially and fully annotated task-oriented dialogue settings can support RLHF without explicit human preference annotations, and we analyze their effects under both corpus-based and online interaction settings.

2 | RELATED WORK

2.1 | Task-oriented dialogue

Recent end-to-end TOD systems increasingly model belief states, system actions, and responses within a unified language-modeling framework [1–4,16]. Although these approaches achieve strong benchmark performance, they are still trained primarily with turn-level supervision from annotated corpora. In addition, even under the same dialogue context and user goal, a single turn may admit multiple appropriate system responses [6], which weakens the assumption that a single supervised target is sufficient for learning robust dialogue behavior. RL has therefore been studied for TOD policy optimization. Prior work includes latent-action formulations for end-to-end dialogue agents [12], model-based approaches that improve policy learning with simulated interaction or learned critics [9, 11, 12, 17], and multi-agent formulations that jointly optimize the system side and user side of the dialogue process [8, 10]. Response-level feedback mechanisms and newer policy-learning formulations have also been explored for dialogue optimization [18–20]. Among these lines, our work is most closely related to CASPI, which learns fine-grained turn-level reward signals for TOD from offline data [15]. The main difference is that we do not assume reward labels derived from user goals during online interaction. Instead, we construct pseudo-preference pairs from available task annotations and study whether such signals can support reward modeling and policy optimization for both system behavior and response generation.

2.2 | Reinforcement learning from human feedback

RLHF has been widely studied as a framework for aligning language models with human judgments, typically using human-annotated preference data to train reward models and optimize policies [21–26]. However, collecting such data is expensive and task-specific, and TOD does not provide an established human preference corpus. Recent work has therefore explored synthetic feedback as a practical alternative to direct human annotation [14, 27]. Our work follows this direction, but differs in task setting and objective. We focus on TOD, where the final objective is defined at the dialogue level and explicit human preference annotations are unavailable. In this setting, we ask whether synthetic turn-level preferences can serve as a practical training signal for reward modeling and policy optimization under online interaction.

3 | METHOD

In this section, we introduce a framework for applying RLHF by generating a turn level preference dataset without any human intervention. Figure 1 illustrates the entire process. Typically, the SFT dataset used to train the initial RL policy and the preference data used to train the reward model(RM) are derived from different dialogue contexts. To achieve this, we partition the data to create separate datasets for SFT and for constructing the preference dataset. A portion of the partitioned data is used to train a stable initial RL policy via SFT, while the remaining data is employed to generate the preference dataset for training the RM. The constructed preference dataset is then used to train the RM and to generate an online dataset for RL. In the final step, we train the RL policy model on the online dataset using the rewards provided by the RM, via Preference policy optimization (PPO) [28].

3.1 | Dataset partitioning

We partition the dataset to separate the data use for training the RM from the data used for training the initial RL policy. Relying on the same data for both models risks propagating the dataset's inherent limitations and biases. When the dataset is skewed toward specific patterns or issues, both models may overfit to those patterns and struggle to generalize in new or online environments. This in turn creates challenges during subsequent RL training with an online dataset. To address these issues, we randomly split the data into K folds. One fold is used to train the initial RL policy, while the remaining folds generate the preference dataset. In our experiments, In our experiments, we use a 4-fold split and denote the subsets as D_0, D_1, D_2, and D_3. Experiments on both partially annotated and fully annotated TOD datasets depend on whether subsets D_1, D_2, and D_3 are annotated with system actions and belief states.

Table 1 shows the distribution of domains within the 4-fold split of the data. Despite random splitting, the domain distribution remains relatively balanced,

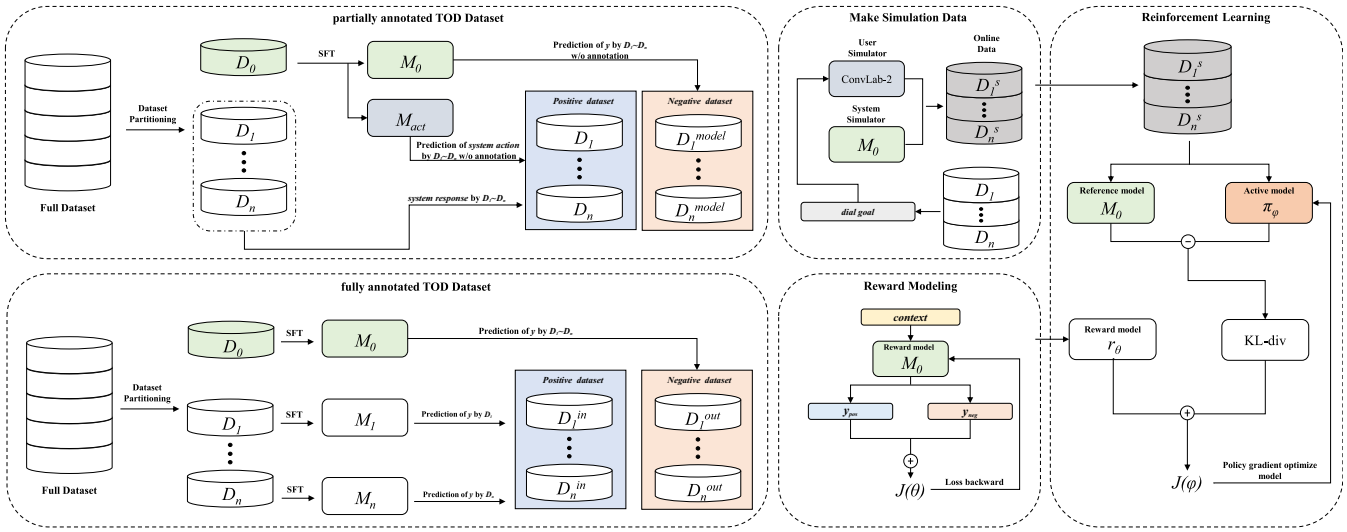


FIGURE 1 Overall pipeline of the proposed method. The framework partitions the dataset, constructs pseudo-preference pairs under partially annotated and fully annotated settings, trains a reward model, and optimizes the initial policy with PPO using simulator-generated online trajectories.

TABLE 1 Four-fold dataset domain distribution.

Domain(s)	D_0	D_1	D_2	D_3
Restaurant	280	317	325	277
Hotel	117	123	134	139
Hospital	79	55	70	83
Taxi	89	75	79	83
Police	54	65	59	67
Train	81	60	70	71
Attraction	27	33	36	31
Hotel-Train	287	277	239	274
Restaurant-Train	219	219	220	216
Attraction-Train	210	223	217	233
Hotel-Attraction	120	102	116	99
Restaurant-Attraction	88	101	109	97
Restaurant-Hotel	109	129	124	100
Restaurant-Hotel-Taxi	118	116	99	120
Restaurant-Attraction-Taxi	110	100	106	114
Hotel-Attraction-Taxi	120	113	105	106
Total	2108	2108	2108	2110

which helps to alleviate concerns about distributional discrepancies between the subsets. In our experiments, we use subset D_0 for the initial RL policy and employ the other subsets to construct the preference dataset for the reward model. A description of the data used here is provided in Section 4.1.

3.2 | Supervised fine-tuning

We perform supervised fine-tuning on the partitioned data and train M_0, M_1, M_2, and M_3 on subsets D_0, D_1, D_2, and D_3, respectively. The training follows the UBAR method [4]. For each turn t, given the previous dialogue history H_t, user utterance U_t, belief state B_t, database search result DB_t, system action A_t, and system response R_t, the model generates the output sequence in the order $(H_t, U_t, B_t, DB_t, A_t, R_t)$.

The model M_0 is later used as the initial policy model, as the initialization backbone for reward modeling, and as the system model for generating the online dataset. In contrast, M_1, M_2, and M_3 are used only during pseudo-preference construction in the fully annotated setting. When the data are not annotated, subsets D_1, D_2, and D_3 cannot be used for supervised training because they lack belief-state and system-action annotations. In that case, only M_0 is available.

In addition, for subsets without system-action annotations, we train a separate model M_{act} to predict the system action from the system response so that pseudo-preference pairs can be constructed. This model is trained on D_0 and generates output in the order (H_t, U_t, R_t, A_t).

3.3 | Synthetic preference dataset construction

To optimize the policy and generate responses via RLHF, we require a preference dataset for training the RM. Each preference instance consists of a dialogue context (H_t, U_t, B_t, DB_t) and two candidate outputs, a positive candidate $(A_{\text{pos}}, R_{\text{pos}})$ and a negative candidate $(A_{\text{neg}}, R_{\text{neg}})$. Because existing TOD corpora do not provide human preference annotations, we construct pseudo-preference pairs using empirical ranking heuristics motivated by prior work on synthetic feedback and model-based ranking signals [14, 15]. These heuristics are not intended to fully recover human judgment. Instead, they provide an operational training signal for reward modeling under different task-annotation regimes.

3.3.1 | Partially annotated setting

In the partially annotated setting, only subset D_0 is annotated and used for supervised training. As a result, subsets D_1, D_2, and D_3, which are used to construct the preference dataset, cannot be used to train partition-specific models because they lack the structured annotations required for standard TOD training. In this setting, we therefore construct pseudo-preference pairs by treating the original system response in the data as the proxy positive candidate and the response generated by the initial model M_0 as the proxy negative candidate.

Concretely, subsets D_1, D_2, and D_3 are used to construct the positive candidate sets D_n^{human} and the negative candidate sets D_n^{model}. For each dialogue context, M_0 predicts the system action and response from the user response, producing $(A_{\text{neg}}, R_{\text{neg}})$. The proxy positive response R_{pos} is taken directly from the original system response in the dataset. Because the system action is not available for these subsets, we use the model M_{act} to generate the corresponding proxy positive action A_{pos} from R_{pos}. The belief state and database search result used in the dialogue context are obtained from the available annotations and preprocessing pipeline. We denote the resulting pseudo-preference direction as

$$D_n^{\text{human}} \succ D_n^{\text{model}}, \; n = 1, 2, 3, \tag{1}$$

where $\succ$ indicates the direction of the constructed ranking signal rather than a validated human preference judgment.

3.3.2 | Fully annotated setting

In the fully annotated setting, subsets D_1, D_2, and D_3 are annotated and can therefore support supervised training. This allows us to train partition-specific models and use them to construct stronger pseudo-preference pairs. In this setting, we treat outputs generated by models trained on the target partition as proxy positive candidates and outputs generated by the initial model M_0 on the same partition as proxy negative candidates.

Specifically, because each subset is annotated, supervised fine-tuning can be applied to each partition to obtain partition-specific models. Using the fine-tuned model corresponding to each partition, we generate $(A_{\text{pos}}, R_{\text{pos}})$ to obtain $D_1^{\text{in}}, D_2^{\text{in}}$, and D_3^{in}. For the same subsets, the initial model M_0 is used to predict $(A_{\text{neg}}, R_{\text{neg}})$, thereby producing $D_1^{\text{out}}, D_2^{\text{out}}$, and D_3^{out}. The belief state and database search results used in the dialogue context are drawn from the original data annotations. We denote the resulting pseudo-preference direction as

$$D_n^{\text{in}} \succ D_n^{\text{out}},\ n = 1, 2, 3, \tag{2}$$

where $\succ$ again denotes the direction of the constructed ranking signal used for reward-model training.

3.4 | Reward modeling

In the reward modeling stage, we train the reward model on the constructed positive and negative candidate pairs. Given a dialogue context c and two candidate output sequences y_{pos} and y_{neg}, the reward model is trained to assign a higher score to the positive candidate than to the negative candidate. Here, each candidate sequence corresponds to the action-response sequence constructed for the same dialogue context.

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

$$J(\theta) = -\mathbb{E}_{(c, y_{\text{pos}}, y_{\text{neg}}) \sim D} \left[\log \left(\sigma \left(r_\theta(y_{\text{pos}}|c) - r_\theta(y_{\text{neg}}|c) \right) \right) \right]. \tag{3}$$

Here, c denotes the dialogue context, and y_{pos} and y_{neg} denote the positive and negative candidate output sequences, respectively. The reward model is initialized from M_0 and optimized on the constructed pseudo-preference dataset.

3.5 | Online dataset construction

We construct online interaction data using the user simulator provided by ConvLab-2 [29] as a proxy for real users. The user simulator employs a rule-based policy module, while the system simulator is set to the fine-tuned TOD model M_0. After extracting user goals from the training subsets D_1, D_2, and D_3, we use these goals to initiate simulated interactions between the user simulator and the system simulator.

Because the resulting online dialogues are later used for reinforcement learning, we sample 1 K user goals from the training data to avoid goal contamination in the test phase. We then collect online interaction trajectories through simulator-based dialogue rollouts.

To examine whether RLHF remains effective under online data of different quality, we construct three online datasets using different response generation modules in the user simulator. As shown in Table S1, the resulting online datasets differ substantially in quality, which allows us to test the robustness of RLHF under varying online interaction conditions. Detailed descriptions of the response generation modules are provided in Section S1.

4 | EXPERIMENTS

We evaluate the proposed framework under two task-annotation regimes, partially annotated data and fully annotated data. For each regime, we construct synthetic preference pairs, train a reward model, and optimize the initial policy on simulator-generated online interaction data. To separate the effect of reward-guided optimization from that of simple additional data exposure, we also include an online-only supervised fine-tuning baseline trained on the same trajectories. All experimental settings, learning rates, and batch sizes follow the configurations summarized in Table 2.

4.1 | Dataset

MultiWOZ 2.1 [30], collected through human-to-human interactions, is an open-source dataset widely used for studying TOD systems. This dataset comprises TOD dialogues for a single domain and multiple domains across 7 domains (hotel, hospital, attraction, train, restaurant,

TABLE 2 Experiment hyperparameters for SFT, RM, and PPO stages and common hyperparameters.

Hyperparameter	Value
SFT stage	
Epochs	15
Batch size	16
Learning rate	1×10^{-4}
Online fine-tuning epochs	3
RM stage	
Model	M_0
Epochs	5
Batch size	16
Learning rate	1×10^{-4}
Development set size	10% of preference dataset
PPO stage	
Model	M_0
Steps	1600
Batch size	4 or 8
Learning rate	1×10^{-6}
β	0.2
Common	
Optimizer	AdamW
Learning rate scheduler	Linear scheduler without warmup
Training device	NVIDIA A100(80G)

police, and taxi). It also provides train/dev/test splits for 8434/1000/1000 dialogues. We follow the provided split and the preprocessing procedures described in DAMD-MultiWOZ.[4]

4.2 | Evaluation metric

We evaluate our method in two complementary settings: corpus-based evaluation and simulator-based evaluation. We retain the standard corpus-based metrics widely used in MultiWOZ-style TOD evaluation to ensure comparability with prior work, and we additionally use simulator-based evaluation to assess interaction-level behavior beyond turn-level response matching.

In the corpus-based setting, we evaluate model responses to user utterances in the corpus given the ground-truth belief states. We report Match, which measures whether the system provides the correct entity, Success, which measures whether the system provides the correct entity and all requested information, and BLEU [33], which measures surface-level similarity between generated and human responses. We also report the Combined score [34], defined as

$$(\text{Match} + \text{Success}) \times 0.5 + \text{BLEU}. \quad (4)$$

We use these corpus-based metrics as standard reference points rather than as complete measures of dialogue quality. To complement them, we also conduct simulator-based evaluation using the ConvLab-2 framework [29]. In this setting, Success Rate measures whether the system provides the correct entity and all requested information, including booking information when needed. Completeness measures whether the original user goal is fulfilled, and Turns denotes the average number of conversational turns per dialogue. Each simulator-based result is averaged over three independent runs. All simulations use user goals from the MultiWOZ 2.1 test split and the GPT-based response module. We interpret corpus-based and simulator-based results jointly, together with the diagnostic analyses in Section 4.3.

4.3 | Experimental results

In this section, we evaluate the proposed synthetic preference-based RLHF framework from three complementary perspectives. First, we examine whether synthetic preference construction improves corpus-based evaluation over the supervised baseline trained on the annotated subset. Second, we compare RLHF with online-only supervised fine-tuning to determine whether the observed gains come from reward-guided optimization rather than from simple additional exposure to online trajectories. Third, because simulator-based outcomes are not uniformly positive, we analyze how RLHF changes dialogue behavior across backbones and investigate the mechanisms behind these differences.

4.3.1 | Corpus-based gains from synthetic preference construction

We first examine whether synthetic preference construction can compensate for the lack of full dialogue annotation. In the partially annotated setting, only D_0 is available for supervised training, and the baseline M_0 is therefore the strongest model that can be obtained without exploiting the remaining unannotated dialogues. Table 3 shows that RLHF in the partially annotated setting consistently improves corpus-based performance over M_0.

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

TABLE 3 MultiWOZ 2.1 corpus-based and simulator-based evaluation results across backbone models and training settings.

		Corpus-based evaluation				Simulator-based evaluation		
Setting	**Online dataset generator**	**Match**	**Success**	**BLEU**	**Combined**	**Success rate**	**Completeness**	**Turns**
GPT-2[31]								
M_0	–	87.00	78.50	16.15	98.90	90.83	92.70	7.18
M_{online}	Template	81.60	73.10	15.39	92.74	89.10	92.40	7.69
	SCLSTM	81.90	71.70	14.78	91.58	80.47	82.47	8.69
	SCGPT	82.30	73.70	16.16	94.16	93.73	95.27	7.18
$M_{\text{partially}}$	Template	87.90	78.80	16.16	99.51	90.07	93.40	7.02
	SCLSTM	88.10	78.80	16.03	99.48	92.10	95.80	7.05
	SCGPT	88.20	79.60	16.35	100.25	91.23	93.63	7.05
M_{fully}	Template	87.80	77.80	16.10	98.90	88.77	92.00	7.01
	SCLSTM	88.30	79.70	16.04	100.04	92.13	94.60	7.02
	SCGPT	88.20	78.00	16.06	99.16	89.40	92.87	6.94
DistilGPT-2[a]								
M_0	–	85.20	75.10	16.89	97.04	97.53	99.20	7.42
M_{online}	Template	78.00	64.20	15.72	86.82	97.53	98.93	7.58
	SCLSTM	77.80	66.30	15.78	87.83	95.73	97.33	7.78
	SCGPT	75.00	65.30	16.49	86.64	97.70	99.00	7.53
$M_{\text{partially}}$	Template	86.70	77.30	16.67	98.67	96.87	98.80	7.37
	SCLSTM	86.20	77.00	16.48	98.08	96.77	98.83	7.30
	SCGPT	85.80	75.90	16.50	97.35	97.07	98.87	7.30
M_{fully}	Template	86.35	76.40	16.66	98.11	96.77	98.90	7.28
	SCLSTM	85.70	76.20	16.61	97.56	96.90	98.90	7.32
	SCGPT	85.40	76.20	16.43	97.23	96.30	98.83	7.26
Pythia-70M[32]								
M_0	–	82.90	71.60	12.94	90.19	93.90	97.30	7.58
M_{online}	Template	78.10	65.50	13.57	85.37	95.77	97.83	7.80
	SCLSTM	77.10	61.10	11.90	81.00	87.27	89.60	9.28
	SCGPT	79.80	66.30	13.34	86.39	95.00	97.67	7.61
$M_{\text{partially}}$	Template	84.30	71.70	12.97	90.97	93.67	96.87	8.06
	SCLSTM	83.50	71.10	13.11	90.41	93.90	97.17	7.77
	SCGPT	84.10	72.40	12.83	91.08	94.50	97.27	8.04
M_{fully}	Template	84.70	71.50	13.09	91.19	93.80	97.03	8.02
	SCLSTM	84.80	72.20	13.03	91.53	93.67	97.10	7.91
	SCGPT	84.60	72.10	13.03	91.38	94.13	97.30	7.88

Note: M_0 is the supervised baseline, M_{online} is the online-only supervised fine-tuning baseline, and $M_{\text{partially}}$ and M_{fully} are the RLHF-trained models under partial and full annotation settings, respectively.

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

For GPT-2, $M_{\text{partially}}$ improves Match from 87.00 to 87.90–88.20, Success from 78.50 to 78.80–79.60, and Combined from 98.90 to 99.48–100.25. DistilGPT-2 shows a similar pattern, where Match increases from 85.20 to 85.80–86.70, Success from 75.10 to 75.90–77.30, and Combined from 97.04 to 97.35–98.67. Pythia also benefits from RLHF under partial annotation, improving Match from 82.90 to 83.50–84.30, Success from 71.60 to 71.10–72.40, and Combined from 90.19 to 90.41–91.08.

These results indicate that unannotated dialogues, although unusable for direct supervised fine-tuning, can still provide useful learning signals once they are

converted into synthetic preference pairs. From a practical perspective, this is important because fully annotating belief states and system actions for dialogue logs is costly. At the same time, these gains should be interpreted as corpus-based improvements. We therefore analyze simulator-based behavior separately in the following subsections.

4.3.2 | RLHF better utilizes online data than online-only fine-tuning

We next compare RLHF with the online-data-only baseline M_{online}, which fine-tunes M_0 directly on the same online dataset without reward modeling or policy optimization. Across all three backbones, Table 3 shows that M_{online} yields substantially worse corpus-based performance than both $M_{\text{partially}}$ and M_{fully}.

This gap is especially clear when the online data are generated by SCLSTM. For GPT-2, M_{online} with SCLSTM reaches Match 81.90, Success 71.70, and Combined 91.58, whereas $M_{\text{partially}}$ with the same online data improves these values to 88.10, 78.80, and 99.48. DistilGPT-2 exhibits a similar pattern, where SCLSTM-based M_{online} achieves 77.80, 66.30, and 87.83, while $M_{\text{partially}}$ increases them to 86.20, 77.00, and 98.08. For Pythia, the same comparison is 77.10, 61.10, and 81.00 for M_{online} versus 83.50, 71.10, and 90.41 for $M_{\text{partially}}$.

These results show that merely exposing the policy to online interaction data is not sufficient. The online data alone can even degrade corpus-based quality when optimized only through supervised imitation. In contrast, RLHF makes substantially better use of the same data by combining online interaction trajectories with reward signals learned from synthetic preferences. This suggests that the benefit of RLHF in our setting is not simply additional data exposure, but a more effective way of extracting useful supervision from online interactions.

4.3.3 | Effect of online dataset quality

The benefit of RLHF should also be interpreted in light of the quality of the online datasets themselves. Table S1 shows that the three online data generators produce substantially different interaction quality. Template-based data achieve a success rate of 96.25 and completeness of 98.25 with 7.10 turns, while GPT-based data achieve 96.87 and 99.13 with 6.81 turns. In contrast, the LSTM-based online data are markedly weaker, with a success rate of 82.98, completeness of 85.46, and 8.24 turns.

These differences are reflected most strongly by M_{online}. In Table 3, the models trained only on online data are most sensitive to the weaker SCLSTM setting. This effect is especially pronounced for Pythia, where SCLSTM-based M_{online} records the lowest corpus-based and simulator-based performance among the online-only models. Therefore, online-only supervised fine-tuning largely inherits the strengths and weaknesses of the collected online data.

RLHF, however, appears less tightly coupled to these raw data characteristics. Although the quality of the online dataset still matters, the robustness analysis in Table S2 shows that the direction of turn change remains stable across all three online datasets. GPT-2 consistently shows negative Δ Turns, DistilGPT-2 also shows negative Δ Turns, and Pythia consistently shows positive Δ Turns. This suggests that RLHF does not merely imitate the online data distribution, but instead reshapes behavior through reward-guided optimization. These results suggest that RLHF may make more stable use of such data than online-only supervised fine-tuning in our setting.

4.3.4 | Backbone-dependent turn-level effects of RLHF

We further examine whether RLHF changes dialogue length in a backbone-dependent manner. Table 4 summarizes turn-related changes over all matched SFT–PPO simulation pairs. A clear pattern emerges: PPO reduces the average number of turns for GPT-2 and DistilGPT-2, but increases it for Pythia.

For GPT-2, PPO reduces the average number of turns by 0.190 while leaving the quality-related metrics nearly unchanged (Δ Success $= -0.744$ pp, Δ Complete $= +0.456$ pp). In addition, 30.9% of paired dialogues become shorter without any drop in success or completeness.

DistilGPT-2 shows the same direction of change, with Δ Turns $= -0.136$. However, unlike GPT-2, its quality changes are slightly less favorable (Δ Success $= -0.878$ pp, Δ Complete $= -0.322$ pp), even though 33.1% of paired dialogues still become shorter without any quality drop.

In contrast, Pythia does not exhibit turn reduction. Instead, PPO increases dialogue length by 0.361 turns on average, while 37.8% of paired dialogues become longer without any gain in success or completeness.

Overall, these results show that RLHF can change interaction length, but the direction of that change is backbone-dependent. For stronger backbones, PPO tends to shorten interactions while largely preserving task-related quality. For the weaker backbone, the same optimization can instead lead to longer trajectories without clear dialogue-level benefit.

TABLE 4 Backbone-level turn changes after PPO training, averaged over all matched SFT–PPO simulation pairs (3 online datasets × 3 run suffixes).

		Average change (PPO − SFT)			Dialogue-level ratio (%)			
Backbone	N_{pairs}	**Δ Turns**	**Δ Success**	**Δ Complete**	**Reduced**	**Increased**	**Shorter/no drop**	**Longer/no gain**
GPT-2	9	−0.190	−0.744	+0.456	35.2	31.4	30.9	28.0
DistilGPT-2	9	−0.136	−0.878	−0.322	34.2	27.9	33.1	27.2
Pythia	9	+0.361	−0.033	−0.156	31.0	39.6	29.4	37.8

Note: PPO reduces average turns for GPT-2 and DistilGPT-2 but increases them for Pythia, indicating that the turn-efficiency benefit of RLHF is backbone-dependent.

TABLE 5 Dialogue-level diagnostic analysis of turn changes.

Backbone	Diagnostic subset	Mean Δ Turns	Mean Δ Offerbook	Mean Δ RefusalPersist	Interpretation
GPT-2	Shorter, no quality drop	−2.278	−0.532	−0.105	Fewer booking loops
DistilGPT-2	Shorter, no quality drop	−1.955	−0.675	−0.151	Branch pruning/simplified booking
Pythia	Longer, no quality gain	+2.745	+1.459	+0.488	Persistent booking behavior

Note: Values are averaged within selected subsets of paired dialogues. The clearest non-zero differences are booking-related. GPT-2 and DistilGPT-2 become shorter when PPO reduces booking prompts and booking persistence, whereas Pythia becomes longer when these behaviors increase.

4.3.5 | Diagnostic analysis of turn-level changes

To better understand why turn counts change after PPO, we conduct a paired dialogue-level diagnostic analysis based on the simulation logs. The diagnostics reported in Table 5 are heuristic measures derived from system action and we use them only to interpret the observed turn changes rather than as primary evaluation metrics.

The clearest non-zero diagnostic differences are booking-related. For GPT-2, the subset of dialogues that becomes shorter without any quality drop shows a mean Δ Turns of −2.278, together with decreases in booking prompts (Δ Offerbook = −0.532) and booking persistence after user refusal (Δ RefusalPersist = −0.105). This indicates that GPT-2 PPO tends to reduce unnecessary booking-related interaction loops.

DistilGPT-2 exhibits a similar but slightly weaker pattern. In the shorter-without-quality-drop subset, Δ Turns is −1.955, while Δ Offerbook and Δ RefusalPersist decrease to −0.675 and −0.151, respectively. This supports the descriptive finding that DistilGPT-2 becomes shorter after PPO, but the mechanism is better characterized as pruning booking-related branches than as uniformly improved dialogue control.

Pythia shows the opposite trend. In the subset of dialogues that becomes longer without any quality gain, Δ Turns rises to +2.745, accompanied by increases in booking prompts (Δ Offerbook = +1.459) and booking persistence (Δ RefusalPersist = +0.488). This suggests that PPO amplifies booking-related interaction loops in Pythia rather than compressing them.

Taken together, these diagnostics provide a concrete interpretation of the backbone-dependent turn pattern observed in Table 4. RLHF shortens dialogues when the backbone can suppress redundant booking-related behavior, but it may lengthen dialogues when the policy fails to stabilize these behaviors under reward optimization. This interpretation is also consistent with the qualitative examples in Section S3. Figures S1 and S2 show representative cases where RLHF shortens the interaction for GPT-2 and DistilGPT-2, whereas Figure S3 shows the opposite case for Pythia, where the RLHF policy continues booking-related responses after user refusal and drifts into another domain, increasing dialogue length without improving task success or completeness.

5 | CONCLUSIONS

This paper investigated whether reinforcement learning from human feedback can be applied to TOD when explicit human preference annotations are unavailable. To study this problem, we proposed two synthetic preference construction strategies for partially annotated and fully annotated task-oriented dialogue data and used the resulting preference pairs to train reward models for policy optimization.

Across the evaluated settings, synthetic-preference RLHF consistently improved corpus-based evaluation over the supervised baseline trained on the annotated subset. In addition, RLHF made better use of the same online interaction trajectories than online-only supervised fine-tuning, which often degraded when trained directly on simulator-generated data. These findings suggest that converting dialogue data that are not directly usable for standard supervised training into preference pairs can turn them into useful training signals.

At the same time, our results show that the effect of RLHF is not uniform in simulator-based evaluation. The turn-level effect of PPO is backbone-dependent. For stronger backbones, PPO tends to shorten interactions while largely preserving task-related quality. For the weaker backbone, the same optimization can instead lead to longer trajectories without clear dialogue-level benefit. Our diagnostic analysis suggests that these differences are closely related to booking-related interaction loops, which are suppressed in some backbones but amplified in others.

Overall, our findings support a narrower conclusion than claiming that RLHF broadly solves alignment for task-oriented dialogue. Instead, they show that synthetic turn-level preferences can provide a practical training signal for task-oriented dialogue in the absence of manual preference annotations, while the resulting gains remain condition-dependent and sensitive to backbone behavior and the quality of the constructed feedback. Future work should directly examine how well the constructed preferences align with human judgment and extend the evaluation beyond a single benchmark and a simulator-based online setting.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

ORCID

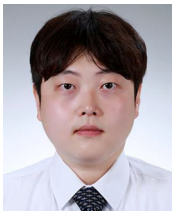
Hyeok-Min Gwon https://orcid.org/0009-0003-1111-7079
Yohan Lee https://orcid.org/0000-0001-8015-9609
Jin-Xia Huang https://orcid.org/0000-0002-2851-5124

REFERENCES

1. Y. Lee, *Improving end-to-end task-oriented dialog system with a simple auxiliary task*, (Findings of the Association for Computational Linguistics: EMNLP 2021), 2021, pp. 1296–1303.
2. Z. Lin, A. Madotto, G. I. Winata, and P. Fung, *MinTL: Minimalist transfer learning for task-oriented dialogue systems*, (Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)), 2020, pp. 3391–3405.
3. T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, *A network-based end-to-end trainable task-oriented dialogue system*, (Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain), 2017, pp. 438–449.
4. Y. Yang, Y. Li, and X. Quan, *UBAR: Towards fully end-to-end task-oriented dialog system with GPT-2*, (Proceedings of the AAAI Conference on artificial Intelligence), 2021, pp. 14230–14238.
5. E. Chung, H. W. Kim, B. Yoo, R. Han, J. Yang, and H. J. Song, *Dialog-based multi-item recommendation using automatic evaluation*, ETRI J. **46** (2024), no. 2, 277–289.
6. Y. Zhang, Z. Ou, and Z. Yu, *Task-oriented dialog systems that consider multiple appropriate responses under the same context*, (Proceedings of the AAAI Conference on Artificial Intelligence), 2020, pp. 9604–9611.
7. P.-H. Su, D. Vandyke, M. Gašić, D. Kim, N. Mrkšić, T.-H. Wen, and S. Young, *Learning from real users: rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems*, (Interspeech 2015, Dresden, Germany), 2015, pp. 2007–2011.
8. A. Papangelis, Y.-C. Wang, P. Molino, and G. Tür, *Collaborative multi-agent dialogue model training via reinforcement learning*, (Proceedings of the 20th Annual Sigdial Meeting on Discourse and Dialogue, Stockholm, Sweden), 2019, pp. 92–102.
9. B. Peng, X. Li, J. Gao, J. Liu, and K.-F. Wong, *Deep Dyna-Q: Integrating planning for task-completion dialogue policy learning*, (Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia), 2018, pp. 2182–2192.
10. R. Takanobu, R. Liang, and M. Huang, *Multi-agent task-oriented dialog policy learning with role-aware reward decompositionc*, (Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics), 2020, pp. 625–638.
11. Y.-C. Wu, B.-H. Tseng, and M. Gašić, *Actor-double-critic: incorporating model-based critic for task-oriented dialogue systems*, (Findings of the Association for Computational Linguistics: EMNLP 2020), 2020, pp. 854–863.
12. T. Zhao, K. Xie, and M. Eskenazi, *Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models*, (Proceedings of NAACL-HLT 2019, Minneapolis, MN, USA), 2019, pp. 1208–1218.
13. C. Musto, A. F. M. Martina, A. Iovine, F. Narducci, M. de Gemmis, and G. Semeraro, *Tell me what you like: introducing natural language preference elicitation strategies in a virtual assistant for the movie domain*, J. Intell. Infor. Syst. **62** (2024), no. 2, 575–599.
14. S. Kim, S. Bae, J. Shin, S. Kang, D. Kwak, K. Yoo, and M. Seo, *Aligning large language models through synthetic feedback*, (Proceedings of the 2023 Conference on empirical methods in Natural Language Processing, Singapore), 2023, pp. 13677–13700.
15. G. S. Ramachandran, K. Hashimoto, and C. Xiong, *CASPI: Causal-aware safe policy improvement for task-oriented dialogue*, (Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland), 2022, pp. 92–102.
16. E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher, *A simple language model for task-oriented dialogue*, (Advances in Neural Information Processing Systems, Vancouver, Canada), 2020, pp. 20179–20191.

17. J. Sun, J. Kou, W. Hou, and Y. Bai, *A multi-agent curiosity reward model for task-oriented dialogue systems*, Pattern Recog. **157** (2025), 110884.
18. J. Chen, B. Zeng, Z. Du, H. Deng, M. Xu, Z. Gan, and M. Ding, *RFM: Response-aware feedback mechanism for background based conversation*, Appl. Intell. **53** (2023), no. 9, 10858–10878.
19. Z. Liu, R. Pang, and Z. Dong, *Task-based dialogue policy learning based on diffusion models*, Appl. Intell. **54** (2024), no. 22, 11752–11764.
20. V. Srinivasan, S. Santhanam, and S. Shaikh, *Using reinforcement learning with external rewards for open-domain natural language generation*, J. Intell. Inform. Syst. **56** (2021), no. 1, 189–206.
21. Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, and N. Joseph, *Training a helpful and harmless assistant with reinforcement learning from human feedback*, arXiv preprint (2022). DOI 10.48550/arXiv:2204.05862
22. P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, *Deep reinforcement learning from human preferences*, (Advances in Neural Information Processing Systems, Long Beach, CA, USA), 2017, pp. 4302–4310.
23. R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, and X. Jiang, *WebGPT: browser-assisted question-answering with human feedback*, arXiv preprint (2021). DOI 10.48550/arXiv:2112.09332
24. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, and J. Schulman, *Training language models to follow instructions with human feedback*, (Advances in Neural information Processing Systems, New Orleans, LA, USA), 2022, pp. 27730–27744.
25. N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, *Learning to summarize from human feedback*, (Advances in Neural Information Processing Systems, Vancouver, Canada), 2020, pp. 3008–3021.
26. D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, *Fine-tuning language models from human preferences*, arXiv preprint (2019). DOI 10.48550/arXiv:1909.08593
27. J. Majkutewicz and J. Szymański, *Aligning large language models with human preferences using historical text edits*, Knowld.-Based Syst. **322** (2025), 113566.
28. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, arXiv preprint (2017). DOI 10.48550/arXiv:1707.06347
29. Q. Zhu, Z. Zhang, Y. Fang, X. Li, R. Takanobu, J. Li, B. Peng, J. Gao, X. Zhu, and M. Huang, *ConvLab-2: an open-source toolkit for building, evaluating, and diagnosing dialogue systems*, (Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations), 2020, pp. 142–149.
30. M. Eric, R. Goel, S. Paul, A. Sethi, S. Agarwal, S. Gao, A. Kumar, A. Goyal, P. Ku, and D. Hakkani-Tur, *Multiwoz 2.1: a consolidated multi-domain dialogue dataset with state corrections and state tracking baselines*, (Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 422–428.
31. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language models are unsupervised multitask learners*, OpenAI (2019).
32. S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, and A. Skowron, *Pythia: a suite for analyzing large language models across training and scaling*, (Proceedings of the 40th International Conference on Machine learning, Honolulu, HI, USA), 2023, pp. 2397–2430.
33. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, *Bleu: a method for automatic evaluation of machine translation*, (Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics), 2002, pp. 311–318.
34. S. Mehri, T. Srinivasan, and M. Eskenazi, *Structured fusion networks for dialog*, (Proceedings of the 20th Annual Sigdial Meeting on Discourse and Dialogue, Stockholm, Sweden), 2019, pp. 165–177.

AUTHOR BIOGRAPHIES



Hyeok-Min Gwon received the BS and MS degrees in Artificial Intelligence from Kyungpook National University, Daegu, Republic of Korea, in 2023 and 2024, respectively. Since 2024, he has been with the Language Intelligence Research Section, Artificial Intelligence Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. His research interests include natural language processing and dialogue systems.

Yohan Lee received the BS and MS degrees in electrical engineering from Korea University, Seoul, Republic of Korea, in 2015 and 2017, respectively. Since 2017, he has been with the Language Intelligence Research Section, Artificial Intelligence Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. His research interests include natural language processing, machine translation, and machine learning.

Jin-Xia Huang received a BS degree in physics from Jilin University, China, in 1991, an MS degree in computer science from KAIST, Republic of Korea, in 2001, and a PhD degree in computer science from Jeonbuk National University, Republic of Korea, in 2018. From 1994 to 1997, she worked as an Engineer at Yanbian University of

Science and Technology (YUST), China, and from 2001 to 2003 as a Researcher at Microsoft Research Asia (MSRA), China. Since 2008, she has been with the Electronics and Telecommunications Research Institute (ETRI), Republic of Korea, where she is currently a Principal Researcher in the Language Intelligent Research Section. Since March 2024, she has also served as an Associate Professor in the AI Major at UST ETRI School. Her research has focused on natural language processing, including dialogue systems and intelligent tutoring agents, and her current research interests include intelligent multi-agent orchestration and AI-driven scientific discovery.

Jonghyuk Lee received the BS degree from the Catholic University of Korea, Bucheon, Republic of Korea, in 2025. Since 2024, he has been with the ITS Division, Korea Expressway Corporation (KEC), Gimcheon, Republic of Korea. His research interests include natural language processing and knowledge distillation.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: H.-M. Gwon, Y. Lee, J.-X. Huang, and J. Lee, *Evaluating reinforcement learning from human feedback for task-oriented dialogue systems*, ETRI Journal (2026), 1–12, DOI 10.4218/etrij.2025-0313