Learning from Relevant Subgoals in Successful Dialogs using Iterative Training for Task-oriented Dialog Systems

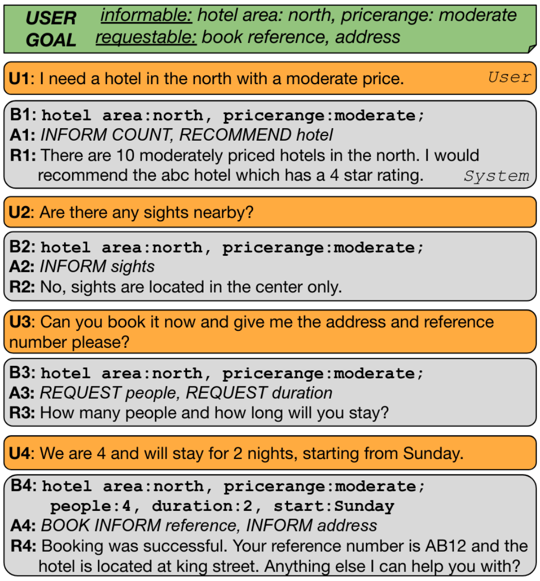
Magdalena Kaiser[1], **Patrick Ernst**[2], **Gyuri Szarvas**[2]
[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
[2] Amazon, Berlin, Germany
mkaiser@mpi-inf.mpg.de, {peernst, szarvasg}@amazon.de

Abstract

Task-oriented Dialog (ToD) systems have to solve multiple subgoals to accomplish user goals, whereas feedback is often obtained only at the end of the dialog. In this work, we propose SUIT (= SUbgoal-aware ITerative Training), an iterative training approach for improving ToD systems. We sample dialogs from the model we aim to improve and determine subgoals that contribute to dialog success using distant supervision to obtain high quality training samples. We show how this data improves supervised fine-tuning or, alternatively, preference learning results. Performance improves when applying these steps over several iterations: SUIT reaches new state-of-the-art performance on a popular ToD benchmark.

1 Introduction

Task-oriented Dialog (ToD) systems nowadays converse with users in natural language conversations and assist them in various tasks, such as booking restaurants, querying weather forecasts and resolving customer service issues. Fig. 1 shows a sample conversation for making a hotel reservation, where the user's goal is defined as a set of constraints (*informable slots*, e.g. pricerange) and information needs (*requestable slots*, e.g. hotel address) that should be satisfied at the end of the dialog. For accomplishing such goals, ToD systems must be able to solve multiple subproblems: (1) dialog state tracking (DST) – understanding user utterances and keeping track of the conversation by storing relevant information in a structured representation of the dialog progress (belief states b_i, e.g. *area* or *price range* in Fig. 1), (2) inferring how to react by selecting some dialog actions like database lookups or requesting more information from the user (actions a_i like *REQUEST number*, *INFORM address* in Fig. 1), (3) formulating a natural language response based on the dialog state and actions (like asking about the length of the stay and the number of people in Fig. 1, r_i). Specialized approaches focus on solving specific problems, e.g., amongst others, (Lee et al., 2021), AG-DST (Tian et al., 2021), D3ST (Zhao et al., 2022) focus on DST, LarL (Zhao et al., 2019), TCUP (Vlastelica et al., 2023) concentrate on response generation. End-to-end (E2E) systems try to leverage complex models to solve all problems jointly. Modern ToD systems in that category are based on pre-trained Large Language Models (LLMs) and cast dialog state, action and response generation into sequence prediction problems. SimpleTOD (Hosseini-Asl et al., 2020) was the first approach which successfully applied this paradigm, by training a causal language model using Supervised Fine-tuning (SFT). A challenge for ToD systems is the fact that ultimate success with respect to the user's goal is observed at the end of the dialog. While Reinforcement Learning (RL) approaches (Zhao et al., 2019; Lubis et al., 2020; Vlastelica et al., 2023) optimize for such sparse rewards, most LLM-based systems neglect these signals and only optimize next turn predictions.



informable: hotel area: north, pricerange: moderate
requestable: book reference, address

U1: I need a hotel in the north with a moderate price.

B1: `hotel area:north, pricerange:moderate;`
A1: *INFORM COUNT, RECOMMEND hotel*
R1: There are 10 moderately priced hotels in the north. I would recommend the abc hotel which has a 4 star rating.

U2: Are there any sights nearby?

B2: `hotel area:north, pricerange:moderate;`
A2: *INFORM sights*
R2: No, sights are located in the center only.

U3: Can you book it now and give me the address and reference number please?

B3: `hotel area:north, pricerange:moderate;`
A3: *REQUEST people, REQUEST duration*
R3: How many people and how long will you stay?

U4: We are 4 and will stay for 2 nights, starting from Sunday.

B4: `hotel area:north, pricerange:moderate; people:4, duration:2, start:Sunday`
A4: *BOOK INFORM reference, INFORM address*
R4: Booking was successful. Your reference number is AB12 and the hotel is located at king street. Anything else I can help you with?

Figure 1: Successful dialog example.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 6236–6246
November 12-16, 2024 ©2024 Association for Computational Linguistics

Contributions. We introduce SUIT (= SUbgoal-aware ITerative Training), an E2E ToD system based on LLMs, which contrary to prior work learns from dialog-level success signals. Due to the sparseness of these signals, it is unclear which turns, states, actions and responses contribute to the overall success of the dialog. For example, the second turn in Fig. 1 is irrelevant for the success of the dialog. The user's goal does not depend on the availability of sights in close vicinity to the hotel, contrary to the respective price range, which will affect the system's success. We aim to identify these important subparts, which we call *subgoals*, from multiple dialog variants generated by an LLM. A naive approach would consider all generations from successful dialogs for SFT, or pair all generations from successful with unsuccessful dialogs for preference learning algorithms, such as RLHF (Christiano et al., 2017) or Direct Preference Optimization (DPO) (Rafailov et al., 2024). However, these naive approaches cannot distinguish between subgoals that are relevant for the final goal from those that are not. We employ an iterative distant supervision approach to identify these subgoals that play a major role in dialog success to obtain relevant training samples. Our contributions are as follows:

1. We propose a sampling approach for finding subgoals using distant supervision without relying on external feedback.
2. We introduce an iterative training procedure for ToD systems.
3. Our simple but effective approach surpasses state-of-the-art performance on a popular ToD benchmark.

2 SUIT Training Approach

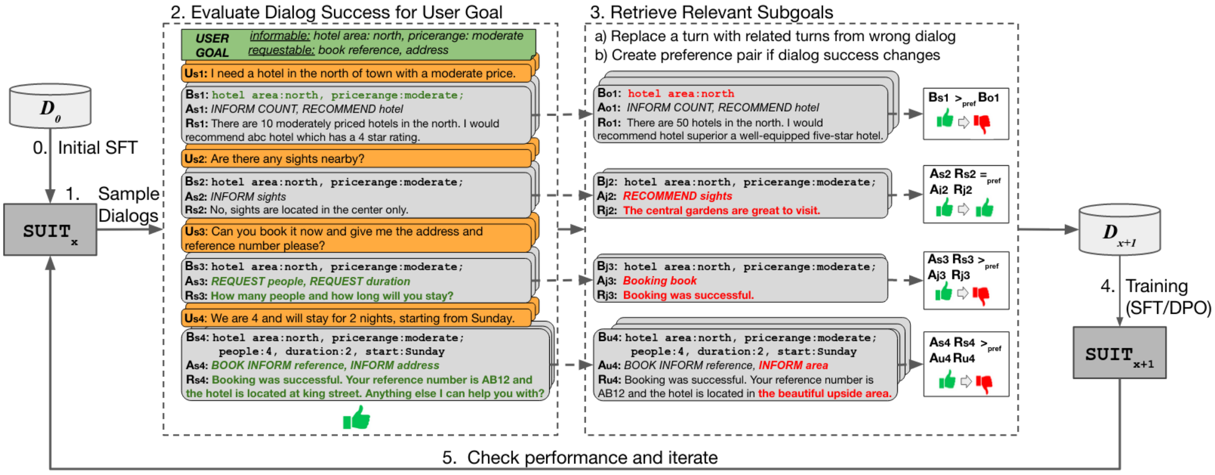
Fig. 2 provides an overview of the SUIT approach. As input we rely on a ToD dataset $\mathbb{D}$, where each dialog is associated with a user goal. This goal describes the user's information need and a set of constraints, that should be fulfilled at the end of the dialog. First, an initial LLM is trained on $\mathbb{D}$ with Supervised Fine-tuning (SFT), using turn-level supervision (Step 0). Then, we sample from this model to create dialog variants for each user goal in the training data (Step 1). Next, we determine dialog success for the newly created dialogs (Step 2). For each successful dialog, we apply distant supervision to identify subgoals that contribute to the ultimate success of the dialog, by comparing them with generations coming from unsuccessful dialogs (more details in Sec. 2.2). The subgoals considered relevant comprise further training samples (Step 3). The SUIT approach can apply SFT or preference learning based on a dataset which pairs subgoals with negative examples from unsuccessful dialogs (Step 4). This procedure is repeatable by sampling from the newly obtained model once again (Step 5). Using this effective training paradigm, SUIT improves SOTA performance (see Sec. 3). Compared to prior E2E ToD systems, SUIT is not based on model customization and does not require feedback from reward models or annotators. The fact, that any off-the-shelf LLM can be plugged into SUIT, makes it simple to set up and use in large scale applications. For preference learning we apply DPO due to its efficiency, low complexity and stable training.

2.1 Initial LLM-based ToD Model

For each turn index t in a dialog D_i, we are given a dialog context $C_{it} = [U_{i0}, S_{i0}, \ldots, S_{it-1}, U_{it}]$ consisting of the current U_{it}, and previous user utterances $U_{ij_{j<t}}$, as well as system turns $S_{ij_{j<t}}$. The goal is to train an initial LLM (SUIT_0) for generating the system turn S_{it}, which contains belief states B_{it}, system actions A_{it}, and a response R_{it}. We split the problem into two prediction tasks:
(1) We predict the belief state $B_{it} = \text{SUIT}_0(C_{it})$;
(2) Actions and responses are jointly generated $T_{it} = \text{SUIT}_0([C_{it}, B_{it}])$, where $T_{it} = [A_{it}, R_{it}]$;
We prompt the model twice and introduce special tokens indicating different parts in the generation (see Appendix A.1). The model is trained to minimize the negative log-likelihood over the training dataset. We provide ground truth belief for action and response prediction at training time. During inference, SUIT generates belief states conditioned on input dialog contexts. Actions and responses are then predicted conditioned on the context and this generated belief.

2.2 Subgoal Candidate Generation

We split each training dialog D_i into all possible dialog contexts C_{it} and sample a *set* of k states $\mathbb{B}_{it}$, and per state, another k actions $\mathbb{A}_{it}$ and responses



USER GOAL informable: hotel area: north, pricerange: moderate
requestable: book reference, address

Us1: I need a hotel in the north of town with a moderate price.

Bs1: `hotel area:north, pricerange:moderate;`
As1: INFORM COUNT, RECOMMEND hotel
Rs1: There are 10 moderately priced hotels in the north. I would recommend abc hotel which has a 4 star rating.

Us2: Are there any sights nearby?

Bs2: `hotel area:north, pricerange:moderate;`
As2: INFORM sights
Rs2: No, sights are located in the center only.

Us3: Can you book it now and give me the address and reference number please?

Bs3: `hotel area:north, pricerange:moderate;`
As3: REQUEST people, REQUEST duration
Rs3: How many people and how long will you stay?

Us4: We are 4 and will stay for 2 nights, starting from Sunday.

Bs4: `hotel area:north, pricerange:moderate; people:4, duration:2, start:Sunday`
As4: BOOK INFORM reference, INFORM address
Rs4: Booking was successful. Your reference number is AB12 and the hotel is located at king street. Anything else I can help you with?

a) Replace a turn with related turns from wrong dialog
b) Create preference pair if dialog success changes

Bo1: `hotel area:north`
Ao1: INFORM COUNT, RECOMMEND hotel
Ro1: There are 50 hotels in the north. I would recommend hotel superior a well-equipped five-star hotel.

Bj2: `hotel area:north, pricerange:moderate;`
Aj2: RECOMMEND sights
Rj2: The central gardens are great to visit.

Bj3: `hotel area:north, pricerange:moderate;`
Aj3: Booking book
Rj3: Booking was successful.

Bu4: `hotel area:north, pricerange:moderate; people:4, duration:2, start:Sunday`
Au4: BOOK INFORM reference, INFORM area
Ru4: Booking was successful. Your reference number is AB12 and the hotel is located in the beautiful upside area.

Figure 2: Overview of training procedure in SUIT. We sample multiple dialogs for one user goal, where each dialog D_s consists of user turns U_{st}, and system turns, which are split into dialog states B_{st}, system actions A_{st} and responses R_{st}. We evaluate dialog success at the end of each generated dialog. For every successful dialog D_s, we replace parts of system turns (subgoals) with the respective parts coming from wrong dialogs $D_{o,j,u}$. If the dialog success flips to unsuccessful, we add the successful subgoal as training data.

$\mathbb{R}_{it}$ from a given SUIT model[1]:

$$\mathbb{B}_{it} = \bigcup_{B \sim \mathrm{SUIT}(C_{it})}^{k} B, \mathbb{T}_{it} = \bigcup_{B \in \mathbb{B}_{it}, T \sim \mathrm{SUIT}(C_{it}, B)}^{k} T$$

Alg. 1 shows the precise steps of the SUIT training loop. We obtain k^2 candidate dialogs $\mathbb{D}_c$ by replacing the turns S_{it} with samples S'_{it} (1 in Alg. 1 and Fig. 2).

Algorithm 1: SUIT Training Approach

```
Input: dialogs $\mathbb{D}_x = \mathbb{D}$, $x = 0$, user goal partitioning
       over dialogs $g(\mathbb{D})$, initial model $\mathrm{SUIT}_x$,
       $train \in \{DPO, SFT\}$, success criteria $succ$
repeat
    $\mathbb{D}_c \leftarrow \bigcup_{D_i \in \mathbb{D}} \{[U_{i0}, \dots, S'_{it}, \dots, S'_{i|D_i|}]|$
        $S'_{it} = [B_{it}, T_{it}] \sim \mathrm{SUIT}_x\}$   (1)
    $\mathbb{D}_{x+1} \leftarrow \emptyset$
    foreach dialog set with same goal $\mathbb{D}_G \in g(\mathbb{D}_c)$ do
        foreach dialog $D_s \in \mathbb{D}_G$: $succ(D_s)$   (2)  do
            $\mathbb{D}_{x+1} \leftarrow \mathbb{D}_{x+1} \cup \{(C_{st}, S'_{st})|$   (3)
            $\exists D_o \in \mathbb{D}_G, S'_{ot} \in D_o : \neg succ(D_o) \wedge$
            $\neg succ([U_{s0}, \dots, S'_{ot}, \dots, S'_{s|D_s|}])\}$
    $\mathrm{SUIT}_{x+1} \leftarrow train(\mathrm{SUIT}_x, \mathbb{D}_{x+1})$   (4)
    $x \leftarrow x + 1$
until $\mathrm{SUIT}_x$ does not improve   (5)
```

2.3 Distant Supervision for Subgoal Detection

We only consider successful dialogs as source of potential training data (2 in Alg. 1 and Fig. 2). To determine dialog success, we use the evaluation function from (Nekvinda and Dušek, 2021) by checking if both INFORM and SUCCESS metrics are fulfilled after the last turn. More precisely, a dialog is considered successful if the last offered entity satisfies the user's goal constraints and the system mentioned all *requestable* slots defined in the user's goal in its response. In Fig. 2 and Alg. 1, dialog D_s is successful. For each successful dialog, we search for unsuccessful dialogs sharing the same user goal (3 in Alg. 1 and Fig. 2 (j, o, u)). If found, we go over the successful dialogs turn-wise and replace state B_{st} (and action/response, $A_{st} R_{st}$) with the respective state (action/response) in the unsuccessful one. After each replacement, we once again evaluate the modified dialog. If the dialog is now unsuccessful, the replaced subgoal was indeed crucial for making it successful. If the dialog is still successful, we cannot make any judgement, since the replacement from the unsuccessful dialog might be correct (there can be correct subgoals in unsuccessful dialogs). Therefore, we sample replacements from different unsuccessful dialogs to see whether the evaluation changes. Please note, that we only make *one* replacement at a time, while the other turns of the successful dialog remain unchanged. State replacements are done separately, while actions and responses are replaced jointly. Replacements for a respective turn t come from another dialog (with same user goal) at the same turn level t. This makes sense in our setup, since samples share the same ground truth dialog context. Nevertheless, our method is robust to different dialog flows. For high variations in dialog flow, one

---

[1] We enforce that the samples contain greedy generations

could additionally apply a similarity based scoring to find the most suitable turn for replacement first. In Fig. 2, replacing the state at turn 1, as well as replacing action/responses at turn 3 and 4 each change the evaluation of dialog s from successful to unsuccessful, whereas the replacement with A_{j2}/R_{j2} results in no change. Therefore, B_{s1}, A_{s3}, R_{s3}, A_{s3}, R_{s3} are considered as relevant subgoals and will be used for training, while A_{s2} and R_{s2} are *not* used as training data, since no replacement was found that changed the evaluation of the dialog. This procedure creates a small, high-quality training set, $\mathbb{D}_{x+1}$ (4 in Alg. 1 and Fig. 2), which is dense in samples that are critical to the final dialog success. For SFT, we use the selected subgoals, while for DPO, we take the selected subgoals as preferred samples (like B_{s1}) and the replacements, which made the dialog change from successful to unsuccessful, as dispreferred samples (like B_{o1}). In summary, SUIT's iterative training approach consists of the following steps:

(1) Given an LLM SUIT$_x$, generate more dialog variations $\mathbb{D}_c$ by sampling (see Sec. 2.2);
(2) Evaluate $success$ for all dialogs using the evaluation function from (Nekvinda and Dušek, 2021);
(3) Identify relevant subgoals by replacing successful ones using distant supervision (see Sec. 2.3);
(4) Apply SFT or DPO to derive model SUIT$_{x+1}$;
(5) Assess model performance, repeat or stop;

3 Experiments

Dataset. We use MultiWOZ 2.2 (Zang et al., 2020), which is a popular ToD benchmark. It contains 10k human-human dialogs over 7 domains.

Metrics. We follow the standardized evaluation from (Nekvinda and Dušek, 2021) to allow for a better comparability. A delexicalized BLEU score measures response coherence, while INFORM and SUCCESS rates express how much a user's goal is fulfilled at the end of the dialog. It is common to assess the overall performance with a COMBINED score $= BLEU + \frac{INFORM+SUCCESS}{2}$.

Model. We use an encoder-decoder Flan-T5 large model, which is trained for 1 epoch per iteration. We verbalize states and actions to be more suitable for generative models. Examples for this verbalization and hyperparameters can be found in App. A.1.

Baselines. We compare with state-of-the-art E2E systems from the MultiWOZ leaderboard. MARS (Sun et al., 2023) uses a contrastive loss to differentiate dialog contexts with the same states from dissimilar ones. KRLS (Yu et al., 2023) applies RL with a special reward, sensitive to important keywords. DIACTTOD (Wu et al., 2023) uses an action encoder to perform nearest neighbor search over latent representations of ground truth actions.

3.1 Main Results

Tab. 1 shows SUIT models, trained for up to two iterations with SFT/DPO. It compares their performances with competitors and baselines, including models trained for one round with all successful dialogs instead of using relevant subgoals only.

SUIT improves SOTA performance. Our initial model reaches the highest BLEU score (19.94). While further iterations decrease this metric slightly, they reach the highest INFORM (90.0), SUCCESS (87.1), and COMBINED (105.02) scores and surpass all E2E competitors on the MultiWOZ leaderboard. The decreasing BLEU score is unavoidable. Due to sampling, the generations may deviate from the fixed reference responses. The responses, however, are still accurate and fluent. Both training paradigms (DPO/SFT) show improvements. In fact, best results are reached with a combination (DPO-SFT).

SUIT identifies the most helpful training data. Training with all successful dialogs, even though comprising a much larger set (31586 vs. 4838 and 2493/2166), performs worse than training with relevant subgoals only. This shows the quality degradation of training data, when not selecting relevant subgoals carefully and confirms the benefits of SUIT's approach (Sec. 2.2).

Sampling new dialogs greatly improves the initial model SUIT$_0$. The INFORM metric improves from 80.4 to 89.8 and SUCCESS from 72.5 to 84.0 after training for one iteration. A second iteration further improves SUCCESS from 84.0 up to 87.1. After the second iteration, the COMBINED score is not increasing any further, thus we stop.

3.2 Domain-wise Results

Tab. 2 shows the domain-wise results for INFORM and SUCCESS of our best SUIT models (per iteration). SUIT performs well on all domains. While results for the train domain are already quite high for SUIT$_0$, results of the other domains are improved a lot over iterations. Especially, there are large jumps for the attraction domain (INFORM from 84.3 to 97 and SUCCESS from 68.9 to 89.4).

Model	BLEU	INFORM	SUCCESS	COMBINED	#samples
MARS (Sun et al., 2023)	19.90	88.9	78.0	103.4	-
KRLS (Yu et al., 2023)	19.00	89.2	80.3	103.8	-
DIACTTOD (Wu et al., 2023)	17.50	89.5	84.2	104.4	-
SUIT$_0$ (initial SFT)	**19.94**	80.4	72.5	96.39	-
+ all subgoals (SFT)	19.50	87.0	79.4	102.70	31586
+ all subgoals (DPO)	17.79	86.9	80.6	101.54	31586
SUIT$_1$ (SFT)	17.75	89.8	84.0	104.65	4838
SUIT$_1$ (DPO)	17.44	88.5	82.7	103.04	4838
SUIT$_2$ (SFT-SFT)	15.11	89.7	85.9	102.91	2493
SUIT$_2$ (SFT-DPO)	17.17	89.5	84.4	104.12	2493
SUIT$_2$ (DPO-SFT)	16.47	**90.0**	**87.1**	**105.02**	2166
SUIT$_2$ (DPO-DPO)	16.92	88.8	84.4	103.52	2166

Table 1: SUIT results compared to other SOTA systems on MultiWOZ 2.2.

	INFORM					SUCCESS				
Model ↓ / Domain →	train	attr.	rest.	taxi	hotel	train	attr.	rest.	taxi	hotel
SUIT$_0$	94.5	84.3	88.3	100.0	83.0	78.8	68.9	75.3	72.8	74.9
SUIT$_1$ (SFT)	92.1	97.5	96.3	100.0	89.8	86.7	85.4	87.2	86.7	83.5
SUIT$_2$ (DPO-SFT)	92.3	97.2	96.1	100.0	90.4	86.7	89.4	89.2	89.7	84.0

Table 2: Domain-wise results for INFORM and SUCCESS metrics of best SUIT model per iteration (*attr.* is short for *attraction* and *rest.* for *restaurant* domain).

4 Related Work

End-to-end ToD Systems. SimpleTOD (Hosseini-Asl et al., 2020) optimizes all sub-tasks jointly using causal language modeling. Prior work uses specialized losses (MTTOD (Lee, 2021), Mars (Sun et al., 2023)) or focus on special encoders (DiactTOD (Wu et al., 2023)) or learn adapters for the individual tasks (TOATOD (Bang et al., 2023)). KRLS (Yu et al., 2023), CASPI (Ramachandran et al., 2022), CALM (Snell et al., 2022) and RewardNet (Feng et al., 2023) apply RL with special reward functions. GALAXY (He et al., 2022) applies semi-supervised learning and in (Steindl et al., 2024) a data augmentation approach by mixing existing dialogs is proposed, whereas our sampling-based approach additionally enriches lexical variety and carefully selects the most helpful training data using distant supervision.

Preference Learning. Stiennon et al. (2020) use RLHF for fine-tuning. Kaufmann et al. (2024) provide an overview of work applying RLHF. DPO (Rafailov et al., 2024), PRO (Song et al., 2024), IPO (Gheshlaghi Azar et al., 2024), and RRHF (Yuan et al., 2023) optimize for preferences with supervised learning. Guo et al. (2024) study these methods in online setups using LLM annotators. Xu et al. (2023) adopt Cringe Loss (Adolphs et al., 2023) to preference data. Contrary to our work, these approaches require external feedback.

5 Conclusion

SUIT is an iterative training approach for ToD systems, which couples sampling to derive new dialogs, with distant supervision to determine subgoals that impact the final dialog success. This coupling enables SUIT models to improve INFORM and SUCCESS metrics and advance the SOTA.

6 Limitations

One limitation of the current approach is the fact that we rely on evaluating dialog success based on ground truth user goals. We do not consider generating new goals, for example by simulating users. By only performing turn-wise replacements, the newly generated training samples will closely follow the flow of the ground truth dialogs. However, for some subgoals order does not matter, e.g., in which order to ask for user preferences is most of the time not crucial for success, and generating them in arbitrary order may increase the diversity of the generated dialogs. The experiments in this paper focus on MultiWOZ, since it is the most common dataset of task-oriented human-human conversations. Due to its adoption and range over multiple domains, MultiWOZ provides a general benchmark for ToD systems. However, transferring the learning and methods to a different dataset could further strengthen the generalizability aspect of this work. We consider the aforementioned limitations for future work.

7 Ethical Considerations

There are no negative ethical and societal concerns arising from this work. The used data is provided by (Zang et al., 2020) and no further human intervention was required. We train models based on FlanT5-large ($783M$ parameters), which can be considered as lightweight compared to much larger GPT/LAMA models, keeping the environmental impact comparatively small.

References

Leonard Adolphs, Tianyu Gao, Jing Xu, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2023. The CRINGE loss: Learning what language not to model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8854–8874, Toronto, Canada. Association for Computational Linguistics.

Namo Bang, Jeehyun Lee, and Myoung-Wan Koo. 2023. Task-optimized adapters for an end-to-end task-oriented dialogue system. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7355–7369.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yihao Feng, Shentao Yang, Shujian Zhang, Jianguo Zhang, Caiming Xiong, Mingyuan Zhou, and Huan Wang. 2023. Fantastic rewards and how to tame them: A case study on reward learning for task-oriented dialogue systems. *Preprint*, arXiv:2302.10342.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. 2024. Direct language model alignment from online ai feedback. *Preprint*, arXiv:2402.04792.

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. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2024. A survey of reinforcement learning from human feedback. *Preprint*, arXiv:2312.14925.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Nurul Lubis, Christian Geishauser, Michael Heck, Hsien-chin Lin, Marco Moresi, Carel van Niekerk, and Milica Gasic. 2020. LAVA: Latent action spaces via variational auto-encoding for dialogue policy optimization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 465–479, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, flavours of success: The case of MultiWOZ. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online. Association for Computational Linguistics.

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.

Charlie Snell, Sherry Yang, Justin Fu, Yi Su, and Sergey Levine. 2022. Context-aware language modeling for goal-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2351–2366.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18990–18998.

Sebastian Steindl, Ulrich Schäfer, and Bernd Ludwig. 2024. Counterfactual dialog mixing as data augmentation for task-oriented dialog systems. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4078–4087.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Haipeng Sun, Junwei Bao, Youzheng Wu, and Xiaodong He. 2023. Mars: Modeling context & state representations with contrastive learning for end-to-end task-oriented dialog. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11139–11160, Toronto, Canada. Association for Computational Linguistics.

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021. Amendable generation for dialogue state tracking. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 80–92, Online. Association for Computational Linguistics.

Marin Vlastelica, Patrick Ernst, and Gyuri Szarvas. 2023. Taming continuous posteriors for latent variational dialogue policies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13673–13681.

Qingyang Wu, James Gung, Raphael Shu, and Yi Zhang. 2023. DiactTOD: Learning generalizable latent dialogue acts for controllable task-oriented dialogue systems. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 255–267, Prague, Czechia. Association for Computational Linguistics.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2023. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*.

Xiao Yu, Qingyang Wu, Kun Qian, and Zhou Yu. 2023. KRLS: Improving end-to-end response generation in task oriented dialog with reinforced keywords learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12338–12358, Singapore. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 10935–10950. Curran Associates, Inc.

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

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling. *CoRR*, abs/2201.08904.

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.

A Appendix

A.1 Experimental Details

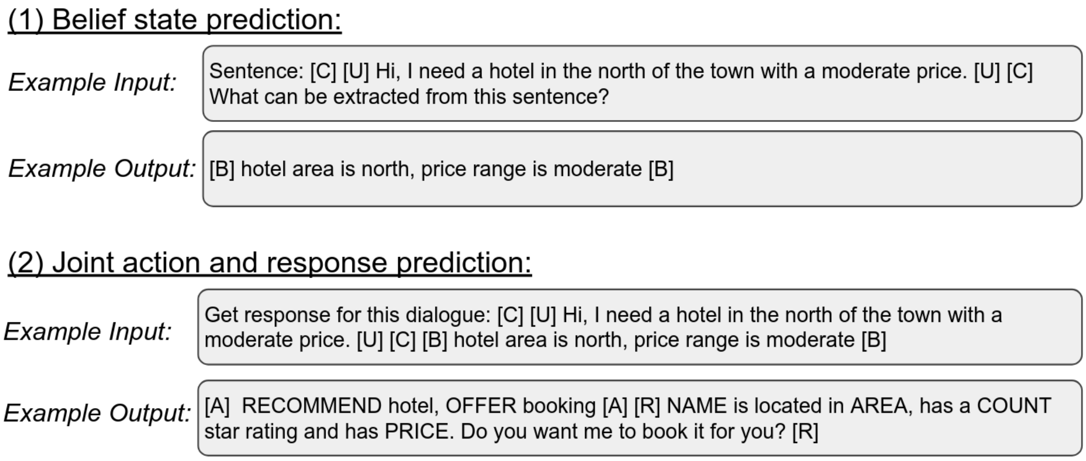
We use a Flan-T5 large model from Hugging Face[2] in our experiments.

Input/Output Representations. As described in Sec. 2.1, we split the generation into two separate prediction tasks:
(1) Predicting the belief state: $B_{it} = \textsc{Suit}_0(C_{it})$
(2) Jointly predicting actions and responses: $[A_{it}, R_{it}] = \textsc{Suit}_0([C_{it}, B_{it}])$
Fig. 3 shows an example for these predictions. Special tokens $[C], [U], [R], [B], [A]$ are used to indicate dialog context, user utterances, system responses, belief states and actions accordingly.

Hyperparameters. After initial model training ($\textsc{Suit}_0$), SUIT models are trained for up to two rounds of SFT/DPO. As stopping criteria we use the COMBINED score. For efficiency, we sample half of the user goals in the training data per iteration for creating new dialogs. We use $k = 2$ for sampling these new dialogs and additionally take

---

[2] `https://huggingface.co/google/flan-t5-large`



(1) Belief state prediction:

Example Input: Sentence: [C] [U] Hi, I need a hotel in the north of the town with a moderate price. [U] [C] What can be extracted from this sentence?

Example Output: [B] hotel area is north, price range is moderate [B]

(2) Joint action and response prediction:

Example Input: Get response for this dialogue: [C] [U] Hi, I need a hotel in the north of the town with a moderate price. [U] [C] [B] hotel area is north, price range is moderate [B]

Example Output: [A] RECOMMEND hotel, OFFER booking [A] [R] NAME is located in AREA, has a COUNT star rating and has PRICE. Do you want me to book it for you? [R]

Figure 3: Example for input/ouput representation in SUIT.

sampled from	# Goals	# Dialogs # suc.	# Dialogs # unsuc.	# Successful Dialogs per Goal 0	1	2	3	4	5
SUIT$_0$	4218	7720	13370	1510	590	599	593	477	449
SUIT$_1$ (SFT)	4218	11983	9107	1212	212	248	407	641	1498
SUIT$_1$ (DPO)	4218	11831	9259	1278	242	236	356	481	1625

Table 3: Sampling statistics for SUIT models.

the greedy generation (resulting in $k^2 + 1$ different dialogs).

For supervised learning, the batch size was set to 2 and learning rate to $5e - 5$. The maximal input length of 512 was used, the maximal target length was 256 and for generation, beam search with a beam size of 5 was used. We use default parameters when sampling from the model (top-k was set to 0).

For DPO training, we set hyperparameters as follows: $\beta = 0.1$, batch size $= 2$, learning rate $= 1e - 6$, warmup steps $= 150$, maximal input length $= 512$, and target length $= 256$.

A.2 Data Statistics

We use the MultiWoZ version 2.2[3] and for pre-processing and evaluation we follow `https://github.com/Tomiinek/MultiWOZ_Evaluation/tree/master`. Tab. 4 shows some statistics about this dataset.

Tab. 3 shows some statistics on sampling new dialogs in SUIT. For each iteration, half of the user goals (4218) in the train set are sampled. Then, 5 new dialogs are created for each sampled user goal, resulting in 21090 newly generated dialogs in total (see amount of dialogs evaluated as succ/unsucc in the table). The right side in Tab. 3 shows the number of user goals for which there were 0/5 successful dialogs, 1/5 successful dialog, ..., up to 5 out of 5 successful dialogs. Entries in 1-4, which contain at least one successful and one unsuccessful dialog, are the candidates used in SUIT, since our approach requires comparison between successful and unsuccessful dialogs for the same user goal.

	# Goals	#Turns	avg. #Turns/Dialog
train set	8437	56776	6.70
dev set	1000	7374	7.37
test set	1000	7372	7.37

Table 4: Data statistics for MultiWOZ 2.2.

sampled from	# Goals	# Subgoal Samples states	# Subgoal Samples actions/responses
SUIT$_0$	2238	836	4002
SUIT$_1$ (SFT)	1461	617	1876
SUIT$_1$ (DPO)	1278	746	1420

Table 5: Subgoal train data statistics for SUIT models.

Tab. 5 shows statistics about the new subgoal-based training data obtained by our approach. We show the number of different user goals present in the new train data as well as the number of subgoals that were considered relevant, split into subgoals

---

[3] `https://github.com/budzianowski/multiwoz/tree/master/data/MultiWOZ_2.2`

representing states and subgoals representing action/response generations. Additionally, we found that the most relevant turns are turn 2-5 and the top-5 relevant dialog acts are: *inform NAME*, *book REFERENCE*, *inform PRICE*, *inform AREA*, *inform PHONE*.

A.3 Example Dialogs

Tab. 6 and 7 show excerpts from sample dialogs, where relevant subgoals were detected, along with the negative example used in DPO. Highlighted in green one can see the relevant information from a subgoal that contributes in reaching dialog success, whereas in red the information from the unsuccessful dialog is shown, that changed the evaluation of the successful dialog when used as a replacement.

Domain: Hotel

Context: User: Hello! Can you tell me about places to stay in the north area of town? I'll be on a business trip, so I do need free wifi. **System:** I have several options that meet your needs. I would recommend the Avalon, unless you need free parking.
User: I do not need parking, is the Avalon moderately priced?

Positive Action: *booking hotel inform NAME; inform PRICE;*
Negative Action: *booking hotel inform PRICE; inform AREA; inform COUNT;*
Positive Response: *NAME is PRICE. would you like me to book it for you?*
Negative Response: *very good! it is in the AREA and has COUNT stars. would you like me to book it for you?* missing PRICE

Domain: Attraction

Context: User: Hi, I am planning my Cambridge trip and could use some help with a particular attraction. **System:** I sure can, what is the name of the attraction you are seeking information about?
User: I'm looking for tenpin I want the address and entrance fee.

Positive Action: *attraction inform ADDRESS; inform PRICE; inform NAME; inform POST; general*
Negative Action: *attraction inform AREA; inform PRICE; inform NAME; general*
Positive Response: NAME is a nearby attraction, admission is PRICE, and they are located at ADDRESS, postcode POST. is there anything else?
Negative Response: NAME is in AREA and it's PRICE to enter. can i help you with anything else? missing ADDRESS

Table 6: Dialog examples with positive/negative subgoals.

Domain: Train

Context: ... **User:** Okay, sounds good. Also I need to get a train from Liverpool street to Cambridge. **System:** What day would you like to make this trip? Do you have any other specifications?

User: The train should leave after 12:45 and should leave on friday. for 2 people please book.

Positive State: `train departure: london liverpool street; destination: cambridge;`

Negative State: `train departure: cambridge; destination: london liverpool street;` swapped departure and destination

Domain: Restaurant

Context: []

User: I'm looking for a restaurant with mediterranean food.

Positive Action: *booking restaurant inform AREA; inform COUNT; inform FOOD; inform NAME; inform PRICE;*

Negative Action: *restaurant inform COUNT;*

Positive Response: i have COUNT PRICE FOOD restaurants in the AREA. NAME and NAME. would you like me to book one for you?

Negative Response: there are COUNT. for booking do you have a preferred information is there a specific time of the day or time of day you would like to arrive by? noisy/less concrete

Domain: Taxi

Context: **User:** I am traveling to Cambridge and excited about seeing location attractions. Could you help me find a place to go, like a college? **System:** Certainly. I have many available. There is corpus christi college, fore example ... **User:** I'm also looking for a hotel in the centre. ... **System:** Your booking at university arms hotel was successful with a reference number of S1HTVV32. Is there anything else I can do for you today?

User: I need a taxi please between the 2 places. I want to leave the attraction by 2:30.

Positive State: `taxi departure: corpus christi; destination: university arms hotel; leave is 02:30; hotel area: centre; bookday: tuesday; bookstay: 1; name: university arms hotel; attraction type: college;`

Negative State: `hotel area: centre; bookday: tuesday; bookstay: 1; name: university arms hotel; stars: 4; attraction type: college;`

Postive Action: *taxi inform PHONE; inform TYPE;*

Negative Action: *taxi request PLACE;*

Postive Response: booking completed! booked car type: TYPE contact number: PHONE

Negative Response: sure, i can provide you a car if you like. where are you departing from? info from context not considered

Table 7: Dialog examples with positive/negative subgoals.