On the Effectiveness of Offline RL for Dialogue Response Generation

Paloma Sodhi [1] **Felix Wu** [1] **Ethan R. Elenberg** [1] **Kilian Q. Weinberger** [1 2] **Ryan McDonald** [1]

Abstract

A common training technique for language models is teacher forcing (TF). TF attempts to match human language exactly, even though identical meanings can be expressed in different ways. This motivates use of sequence-level objectives for dialogue response generation. In this paper, we study the efficacy of various offline reinforcement learning (RL) methods to maximize such objectives. We present a comprehensive evaluation across multiple datasets, models, and metrics. Offline RL shows a clear performance improvement over teacher forcing while not inducing training instability or sacrificing practical training budgets.[3]

1. Introduction

Dialogue response generation is an important task in natural language processing with numerous applications such as virtual personal assistants and call center agent tools (Zhou et al., 2017; Swanson et al., 2019; Jaques et al., 2020; Ramakrishnan et al., 2022; Ouyang et al., 2022). Historically, text generation models have typically been trained with teacher forcing (TF) (Williams & Zipser, 1989), which involves predicting the next token in a sequence to exactly match the human utterance in a ground truth dataset. However, this can be a needlessly challenging objective, as a human may choose to say the same thing in multiple different ways. Consider a dialogue system that provides suggestions to an agent during a conversation with a customer. These suggestions need only be *close enough* for an agent to select it. This suggests a different objective, one that is defined on the entire sentence rather than individual tokens.

One way to design such a loss would be to incorporate human-in-the-loop feedback if a model generated utterance matches the meaning of the ground truth sentence. However, this can be expensive to collect. Instead, model-based metrics to measure utterance similarity, such as BERTScore (Zhang et al., 2019) and BLEURT (Sellam et al., 2020), provide a cheaper alternative. These are automated metrics that capture semantic similarity between sentences and tend to have a high correlation with human judgment (Zhang et al., 2019; Sellam et al., 2020). Given a choice of such metrics, what learning framework would allow us to maximize them for dialogue text generation?

Recent works have explored online RL methods for text generation (Ranzato et al., 2015; Li et al., 2016; Ouyang et al., 2022; Ramamurthy et al., 2022), leading to some exciting successes (Ouyang et al., 2022). However, *offline* RL has received relatively less attention (Jaques et al., 2020; Pang & He, 2020). We argue that offline RL (Levine et al., 2020) does provide a framework that meets all aforementioned desiderata. Unlike teacher forcing, it can handle losses on the entire sequence as a reward function and unlike online RL, it can leverage existing data without having to explore, matching similar training times as teacher forcing.

In this paper, we present a comprehensive evaluation of offline RL methods for dialogue text generation and investigate best practices. We explore three complementary approaches. The first, TF Top, is to fine-tune a model on utterances that accrue high returns. The second, Decision Transformers (DT) (Chen et al., 2021b), is to train a conditional model that conditions on returns, and at inference time condition on a high return. The third, ILQL (Kostrikov et al., 2021; Snell et al., 2022), is an off-policy Q-learning approach that uses dynamic programming to train a critic. All three of these approaches are complementary and have been shown to be competitive outside of dialogue settings, making them great candidates to evaluate the efficacy of offline RL for dialogue text generation.

To summarize our contributions, we formalize three state-of-the-art offline RL approaches for the task of dialogue text generation. We evaluate them across multiple data sets, models, and metrics and provide a thorough ablation analysis of these approaches. We find that offline RL methods show a clear performance improvement over teacher forcing and achieve a trade-off where they generate text close enough in meaning to human. Through different experiments, we demonstrate that the offline RL framework

[1] ASAPP, New York, United States [2] Cornell University, New York, United States. Correspondence to: Paloma Sodhi <psodhi@asapp.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[3] Our code is available at `https://github.com/asappresearch/dialogue-offline-rl`

provides an ideal fit for the task of dialogue generation, and should be considered seriously by the community.

2. Related Work

RL for NLP. Prior work has used RL techniques to improve models in a variety of NLP applications (Ranzato et al., 2015; Pang & He, 2020; Yang et al., 2020; Lu et al., 2022; Snell et al., 2022; Ramamurthy et al., 2022) such as machine translation (Yonghui et al., 2016; Wu et al., 2018; Kiegeland & Kreutzer, 2021), summarization (Paulus et al., 2017; Pasunuru & Bansal, 2018; Stiennon et al., 2020), question answering (Furman et al., 2022), visual reasoning (Wu et al., 2022) and instruction following (Misra & Artzi, 2015; Ouyang et al., 2022). Techniques adopted range from online RL methods like REINFORCE (Williams, 1992) and PPO (Schulman et al., 2017) to offline RL approaches like conservative Q-learning (CQL) (Kumar et al., 2020) and decision transformers (Chen et al., 2021b).

RL for Dialogue Generation. Dialogue generation can be challenging as generated sequences can be long and in each turn there can be multiple acceptable responses. Li et al. (2016) use REINFORCE to optimize a set of rewards that capture informativity, coherence, and ease of answering. Zhou et al. (2017) use a mixture of on and off-policy policy gradient to optimize a reward that captures both utterance-level and dialog-level rewards. Jaques et al. (2019; 2020) use offline RL to optimize a learned reward function from human responses. Ouyang et al. (2022) use PPO to optimize a learned reward model from human ranking. Our goal is to generate dialogue responses that are semantically close to ground truth utterances without having to design explicit rewards that capture dialogue success Liu et al. (2016). This is complementary to approaches that look at optimizing dialogue-level metrics like key values for slots (Lee et al., 2021; Tian et al., 2021; Bang et al., 2023).

Offline RL for NLP. Offline RL removes the need for interaction during train time operating only on static datasets of prior human interaction, which leads to improved training stability. Pang & He (2020) use importance weighted REINFORCE, which only trains a policy without a critic to control for variance. Verma et al. (2022) use CQL but operate on entire utterances and not per token thus reasoning over shorter sequences. Jaques et al. (2019; 2020) operate at per-token level using off-policy Q-learning, but require generation at RL training time that can be expensive. Snell et al. (2022) propose ILQL, a variant of CQL with implicit dataset support constraints, that requires no such generation at train time. Lu et al. (2022) propose Quark, that uses Decision Transformers by quantizing rewards. While both papers explore metrics like toxicity and sentiment, they don't optimize for similarity to human utterances in dialogue settings that we examine in this paper.

3. Problem Formulation

3.1. Dialogue Response Generation as an MDP

We look at the problem of dialogue text generation, *i.e.*, generating response utterances in a dialogue setting. We begin with a supervised dataset of context response pairs $\{x^i, y^i\}_{i=1}^N$, where context x is the conversation history, and response $y = \{y_1, \ldots, y_T\}$ is a target sequence of tokens. We map each data point (x, y) to an episode of a Markov Decision Process (MDP), which we define below (Fig. 1):

- **States,** $s_t \in \mathcal{S}$ is the context x and the partially generated sequence of tokens up to and including time step t, $\hat{y}_{<t} := \{\hat{y}_1, \ldots, \hat{y}_t\}$.
- **Actions,** $a_t \in \mathcal{A}$ are the set of next tokens $\hat{y}_{t+1}$ available from the vocabulary V
- **Transition function,** $\mathcal{T}(s_{t+1}|s_t, a_t)$ is deterministic since every state-action pair $(\hat{y}_{<t}, \hat{y}_{t+1})$ leads to a unique state $\hat{y}_{<t+1}$ for the next step.
- **Rewards,** $r_t : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a terminal reward that computes similarity generated response $\hat{y}$ and target response y
- **Horizon,** T is time horizon. Each episode ends when the current time step t exceeds T or an end-of-sentence (EOS) token is generated.

The goal is to learn a policy $\pi : s_t \to a_t$ maximizing *return*, *i.e.* the cumulative reward over an episode $\mathbb{E}_\pi \sum_{t=0}^{T} \gamma^t r_t$. We assume undiscounted cumulative rewards, *i.e.* $\gamma = 1$.

3.2. Rewards for Dialogue Response Generation

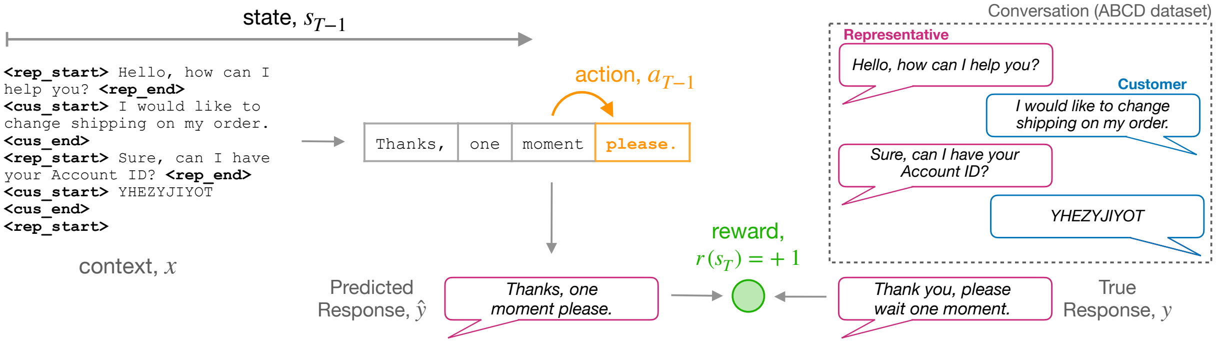
We define the reward to be a similarity metric between the generated text $\hat{y}$ and the speaker's ground truth utterance y. Such a metric should capture both what the speaker is trying to communicate and the relevance to the conversation.

One option is to collect human-in-the-loop annotations, *i.e.* what the speaker would likely prefer to say. However, this requires costly human supervision. Automated metrics, such as BERTScore (Zhang et al., 2019), BLEURT (Sellam et al., 2020), offer a promising alternative. They are able to capture models of human preference and are cheap to evaluate.

We use a terminal reward since the similarity can only be evaluated at the end of the utterance, and since the same content can be expressed in different styles, *e.g. The flight from New York to Boston has been confirmed.* vs. *Your JFK to BOS flight has been canceled.*

3.3. Why Offline Reinforcement Learning?

In online reinforcement learning, an agent learns by interacting with an environment in real-time. This presents an explore-exploit trade-off, where the agent must balance the need to try out new actions to learn about the environment



```
<rep_start> Hello, how can I
help you? <rep_end>
<cus_start> I would like to
change shipping on my order.
<cus_end>
<rep_start> Sure, can I have
your Account ID? <rep_end>
<cus_start> YHEZYJIYOT
<cus_end>
<rep_start>
```

Hello, how can I help you?

I would like to change shipping on my order.

Sure, can I have your Account ID?

YHEZYJIYOT

Thanks, one moment please.

Thank you, please wait one moment.

Figure 1. **Dialogue generation as a Markov Decision Process (MDP)** Given a dataset of context-response pairs, each pair is mapped to an MDP episode. States are context x_i and partially generated response up to time t, actions are next tokens from vocabulary V, and rewards are computed by comparing generated response $\hat{y}$ against target response once t exceeds T or end of sentence token is generated.

with the need to exploit its current knowledge to maximize reward. This can be particularly challenging in text generation, as action space (*i.e.* vocabulary size) is often large, *e.g.* of the order of 50,000 words for GPT-based models (Radford et al., 2019). Another problem is that the reward landscape is sparse, hence policies during training can get stuck in local minima where reward is persistently zero.

For text generation, we argue that an offline setting is reasonable. There exists good generation policies, e.g. policies from teacher forcing, that can generate a set of responses such that one of them is close enough to the human response. Also, once a token is generated, we *deterministically* transition to next state with the additional token appended to the prefix, *i.e.* no interaction is needed to learn the environment.

Offline RL provides a learning paradigm that combines both supervised learning's ability to leverage existing data with the general utility optimization power of online reinforcement learning methods. We collect an offline dataset of state transitions $\mathcal{D} = \{(s^i_t, a^i_t, r^i_t, s^i_{t+1})\}_{i=1}^{N}$[1] using a behavior policy π_β, typically a policy trained via supervised learning. The goal is to learn a policy π that maximizes performance on the dataset while staying close to the behavior policy:

$$\max_\pi J_{\mathcal{D}}(\pi) - \alpha D(\pi, \pi_\beta)\,, \tag{1}$$

where $J_{\mathcal{D}}(\cdot)$ is performance on dataset $\mathcal{D}$ and $D(\cdot, \pi_\beta)$ is distributional regularization against behavior policy π_β.

4. Approach

In this section we introduce and compare three recent approaches to offline RL. For all methods, we begin with a pre-trained language model π_β trained via teacher-forcing, and use this to generate the offline dataset $\mathcal{D}$.

4.1. Fine Tune on Top Returns

The simplest approach is to fine-tune a model on "top" demonstrations, *i.e.* teacher forcing on top returns (TF-Top)

We define a subset of the dataset $\mathcal{D}_{\text{top}}$ that has high returns above a specified threshold, where return is the cumulative reward until the end of the episode, $\hat{Q}(s_t, a_t) = \sum_t^T r_t$. The gradient update is simply the log-likelihood gradient on the data subset $\mathcal{D}_{\text{top}}$,

$$\begin{gathered}\mathop{\mathbb{E}}_{s_t, a_t \sim \mathcal{D}_{\text{top}}} \left[\nabla_\theta \log \pi_\theta(a_t|s_t)\right]\,, \\ \text{where}\;\; \mathcal{D}_{\text{top}} = \{(s_t, a_t) \in \mathcal{D} \mid \hat{Q}(s_t, a_t) \geq 1 - \delta\}\,.\end{gathered} \tag{2}$$

Here, δ is a specified threshold defining a "good enough" return. δ can be computed by taking the top percentile of all returns $\hat{Q}(s_t, a_t)$ [2]. Since we use a terminal undiscounted reward, the return for any token along the sequence is the same as the final reward received at the end of the sequence. Additionally, if the reward is binary $\{0, 1\}$, $\mathcal{D}_{\text{top}}$ selects sequences corresponding to the reward 1.

However, one artifact of this approach is that it only increases likelihood of "good" tokens, but doesn't necessarily decrease the likelihood of "bad" tokens. This is because we *discard* trajectories with low return that were likely under the original TF policy π_β, rather than using them to update the model's parameters in the opposite direction.

4.2. Decision Transformers: Condition on Return

Decision Transformer (DT) (Chen et al., 2021b) is an approach that reduces offline reinforcement learning to supervised learning. The core idea of DT is to learn the return-conditional distribution of actions in each state, and then define a policy by sampling from the distribution of actions that receive high returns.

Given a data point (s_t, a_t), we take its return $\hat{Q}(s_t, a_t)$, tokenize it, and then fine tune a model by conditioning on

[1] We suppress superscripts when considering a single transition.

[2] Note that $\hat{Q}(s_t, a_t) \leq 1$.

this return token. The gradient update is simply the log-likelihood,

$$\mathop{\mathbb{E}}_{s_t, a_t \sim \mathcal{D}} \left[\nabla_\theta \log \pi_\theta(a_t | s_t, \hat{Q}(s_t, a_t)) \right] \quad (3)$$

At test time, we condition the model on the highest return $\hat{Q}_{top}$, *i.e.* we sample sequences from $\pi_\theta(.|s_t, \hat{Q}(s_t, a_t) = \hat{Q}_{top})$. We implement DT by quantizing the return $\hat{Q}(s_t, a_t)$ into K bins, assigning a token for each bin, training a conditional model and at test time conditioning on the top bin. For binary rewards $\{0, 1\}$, this is equivalent to training a model on $r = 0$ and $r = 1$ tokens, and then conditioning on $r = 1$ at test time.

One advantage of decision transformer over fine-tuning on top returns is that the model is trained to explicitly learn a decision boundary between different returns. However, both approaches have the theoretical drawback of requiring ”trajectory coverage” (Brandfonbrener et al., 2022), *i.e.* the training dataset must contain trajectories starting from the initial state s_0 that sees high return. This can be challenging in general because the number of data points needed increases exponentially with the length of the trajectory.

4.3. Off-Policy Q-Learning

A canonical approach to RL is Q-learning (Watkins & Dayan, 1992). We use an offline variant, Implicit Q-learning (ILQL) (Snell et al., 2022), as an off-policy Q-learning method architected for language models.

ILQL adds two extra heads to the pre-trained model, the action value head $Q_\theta(s_t, a_t)$ and the state value head $V_\psi(s_t)$. The state value $V_\psi(s_t)$ denotes the value of the sequence s_t, while the action value $Q_\theta(s_t, a_t)$ denotes the utility of a token a_t given a sequence s_t. Hence the advantage $A(s_t, a_t) = Q_\theta(s_t, a_t) - V_\psi(s_t)$ is the utility of next token a_t over any other alternate token.

Before we describe how both heads are trained, we first note that ILQL does not explicitly train a policy. Instead, it defines an *implicit* policy by taking the logits from pre-trained model π_β and rescaling it by a weighted advantage:

$$\pi_\theta(a_t|s_t) = \pi_\beta(a_t|s_t) \exp\left(\eta(Q_\theta(s_t, a_t) - V_\psi(s_t))\right) \quad (4)$$

The loss for the $Q_\theta(\cdot)$ head has two terms. The first is the temporal difference (TD) error coming from the Bellman equation. The second is a regularization for the policy to be close to the pre-trained policy π_β. The gradient update is a sum of these two terms:

$$\mathop{\mathbb{E}}_{\substack{s_t, a_t, \\ s_{t+1} \sim \mathcal{D}}} \left[\nabla_\theta Q_\theta(s_t, a_t) \underbrace{(r(s_t, a_t) + V_\psi(s_{t+1}) - Q_\theta(s_t, a_t))}_{\text{Temporal Difference Error}} \right] - \alpha \mathbb{E}_{s_t \sim \mathcal{D}} \nabla_\theta KL(\pi_\beta(.|s_t) || \pi_\theta(.|s_t)) \, , \quad (5)$$

Value head $V_\psi(s_t)$ is trained to approximate argmax of Q, *i.e.* on constrained Bellman operator with expectile regression.

$$\mathop{\mathbb{E}}_{s_t, a_t \sim \mathcal{D}} \nabla_\psi ||Q_\theta(s_t, a_t) - V_\psi(s_t)||^\tau \, , \quad (6)$$

where $||u||^\tau = (\tau - \mathbb{1}(u < 0))u^2$ is the τ expectile.

We *improve upon original ILQL* (Snell et al., 2022) by regularizing against logits of the pre-trained TF policy π_β instead of the demonstrated data $\mathcal{D}$. This is more suited for settings where we may not have a lot of demonstrated data.

4.4. On-Policy RL: PPO

In addition to the offline RL approaches, we also compare against an online RL algorithm: Proximal Policy Optimization (Schulman et al., 2017). PPO is a variant of a policy gradient approach that rolls out a trajectory with the current policy π_θ to sample (s_t, a_t), estimates the advantage $A(s_t, a_t)$, and updates policy to maximize advantage while staying close to old policy $\pi_{\theta_{\text{old}}}$ The gradient update is,

$$\mathop{\mathbb{E}}_{s_t, a_t \sim \pi_\theta} \left[\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} A(s_t, a_t) \right] \quad (7)$$

4.5. Comparison between Approaches

When is DT and Q-learning comparable? While DT is relatively simple and faster to train, it has a more restrictive requirement of data coverage than Q-learning. Intuitively, it is unable to stitch together suboptimal trajectories that overlap into a better policy. However, for MDPs where such stitching is not possible, e.g. a tree, DT and ILQL are comparable in performance. We hypothesize that dialogue text generation belongs to this class of MDPs.

When is DT and TF Top comparable? While DT makes use of more data than TF Top, it does deal with a more complex function class (conditioning on returns). Intuitively, DT should expect to do better than TF Top only when the data TF Top throws away provides valuable information. If that information is already captured by base TF model, then both DT and TF Top are likely to be similar.

5. Experiments

5.1. Experimental Setup

5.1.1. Task-oriented Dialogue Datasets

We evaluate offline RL methods using three task-oriented dialogue datasets. These are relevant for dialogue systems designed for real-world applications, where users have specific goals and tasks that they want to accomplish. Each dataset consists of conversations between two speakers: one is the system or agent, and the other is the user or customer. We optimize rewards on system or agent utterances so as to emulate applications designed to assist an agent (e.g customer service representative) in providing helpful and human-like responses to customer queries and problems.

MultiWoz 2.2 (Zang et al., 2020) is a widely used dataset created to evaluate performance of dialogue systems in multi-domain settings. It consists of over 10k conversations spanning 8 domains like hotel, train, restaurant, etc.

Action Based Conversations Dataset (ABCD) (Chen et al., 2021a) contains customer-agent conversations where the agent's goal is to solve a customer problem. It consists of over 10k conversations spread over 55 user intents in the retail customer service domain.

TaskMaster-3 (Byrne et al., 2019): contains 23,789 conversations between users and a system on movie ticketing.

5.1.2. Baselines and Metrics

We choose a terminal binary reward BERTClick, which is a thresholded BERTScore (Zhang et al., 2019) with threshold value 0.6. We select a value of 0.6 qualitatively such the generated response is close enough and has similar meaning to human response. We evaluate on a range of automated similarity metrics shown to have a high correlation with human judgements like BERTScore (Zhang et al., 2019), BLEURT (Sellam et al., 2020), METEOR (Banerjee & Lavie, 2005) and BLEU (Papineni et al., 2002). We also do human evaluation on a subset of the data where we ask humans to rate similarity and relevance on a scale of 1-3. More details on the study in appendix.

We evaluate across following methods and baselines: **TF**, Base model trained via teacher forcing on all conversations, **TF All**, TF model fine tuned on entire offline RL Dataset, **TF Top**, TF model fine tuned only on data points with top returns $\mathcal{D}_{top}$, **DT**, Decision Transformer, **ILQL**, Off-policy Q-learning, **PPO**, Online RL via policy gradients.

We train the TF model on all the training data (stage 1), use this trained TF model to generate an offline RL dataset (stage 2), and finally fine tune different RL models on varying percentages of generated offline RL data (stage 3). More details on training setup are in the appendix. Since the generation step is expensive, we would like to be able to fine tune on subsets of offline RL dataset for improved efficiency and train time budgets.

For base models we study GPT2Medium[3] (Radford et al., 2019) and DistilGPT[4] (Sanh et al., 2019) which have 355M and 82M parameters, respectively. For real-time environments, models like distilGPT2 are preferable since they have low latency (order of 100 ms) to be used in dialogue settings. We use huggingface transformers library (Wolf et al., 2019) to implement TF Top, DT and trlx[5] for ILQL, PPO.

Finally, we evaluate models as both generators and rankers. For ranker, we score set of responses generated by base **TF** model and pick highest score. In our experiments, we found **ILQL** to be more effective as a ranker, as it trains a critic rather than an actor. Hence, we evaluate **ILQL** as a ranker.

5.2. Results and Analysis

We analyze the results through a series of questions.

5.2.1. Overall Performance Gains

Do offline RL methods improve on average over base teacher forcing model? Table 1 presents average metrics for **TF**, **TF All**, **TF Top**, **DT** on all datasets. We see that on all datasets the offline RL methods improve the average reward (BERTClick) from 1.5% (TaskMaster) to 5% (ABCD, MultiWoz). Offline RL methods also improve on other metrics not part of the reward, e.g. 2% to 3% on METEOR and 2% (ABCD, MultiWoz) to 3% on BLEU (ABCD). These improvements come without sacrificing perplexity (lower perplexity in ABCD and MultiWoz). Finally, we also note performance gains on TaskMaster are not as large as the other datasets.

On most datasets and metrics, **DT** outperforms the other methods. The performance of **DT** over **TF Top** is consistent when fine-tuned on 20% of the dataset vs 80% (analyzed later in Fig. 6). While Table 1 shows only average metrics, we also look at the distribution over BERTscore in Fig. 2. We see that offline RL methods have a higher probability mass than TF on almost all BERTScore bins ≥ 0.6. This is expected as 0.6 is the threshold for BERTClick used as the reward function. The results show that improvements is not limited to any one bin, but across all bins. On a majority of datasets and bins, **DT** outperforms **TF Top**.

How does performance vary across multiple responses? An argument in favor of the base **TF** model might be that it's unfair to evaluate it on a single response. After all, it optimizes for recall, so with multiple responses, it should be able to reach the performance of offline RL methods.

Fig. 3 shows average BERTClick of the best response selected from multiple responses. We see that offline RL methods maintain a persistent gap above **TF** model on all datasets. This likely indicates that they converge on a better distribution of responses over **TF**. **DT**, **TF Top** are similar for ABCD, TaskMaster, but **DT** outperforms on MultiWoz.

[3] `https://huggingface.co/gpt2-medium`

[4] `https://huggingface.co/distilgpt2`

[5] `https://github.com/CarperAI/trlx`

	Algorithm	BERTCLICK		BERTSCORE		BLEURT		METEOR		BLEU		PERPLEXITY(↓)	
		20%	80%	20%	80%	20%	80%	20%	80%	20%	80%	20%	80%
ABCD	TF	0.276		0.404		0.571		0.370		0.135		36.36	
	TF All	0.269	0.285	0.390	0.399	0.559	0.564	0.365	0.375	0.134	0.143	41.76	45.39
	TF Top	0.281	0.307	0.388	0.420	0.559	0.576	0.358	0.382	0.135	0.156	36.82	34.25
	DT	0.299	0.321	0.411	0.429	0.572	0.582	0.372	0.391	0.144	0.155	36.22	36.51
MultiWoz 2.2	TF	0.130		0.366		0.512		0.312		0.074		48.97	
	TF All	0.148	0.163	0.368	0.376	0.512	0.519	0.308	0.313	0.085	0.082	42.62	45.83
	TF Top	0.150	0.179	0.373	0.394	0.513	0.530	0.303	0.325	0.080	0.092	42.84	41.54
	DT	0.170	0.171	0.380	0.392	0.523	0.531	0.316	0.331	0.087	0.088	44.45	37.77
TaskMaster-3	TF	0.446		0.554		0.624		0.513		0.360		77.18	
	TF All	0.438	0.450	0.450	0.546	0.621	0.621	0.501	0.507	0.347	0.350	70.93	69.56
	TF Top	0.431	0.453	0.533	0.556	0.612	0.626	0.487	0.511	0.328	0.357	65.24	70.31
	DT	0.436	0.460	0.548	0.562	0.617	0.630	0.498	0.514	0.342	0.359	69.00	74.67

Table 1. Comparison across different methods on average metrics and dataset size with distilGPT2. 20%, 80% refer to percentage of the data used for fine-tuning offline RL methods. For consistency, BLEU scores are in [0, 1] unlike some papers converting them to [0, 100].

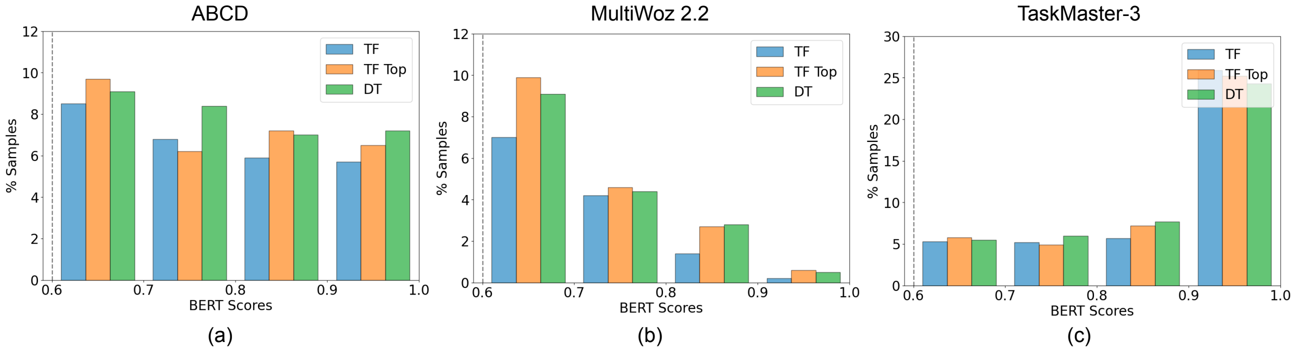


Figure 2. Distribution over BERTSCORES for **(a)** ABCD **(b)** MultiWoz **(c)** Taskmaster-3 datasets with distilGPT2 finetuned on 80% data.

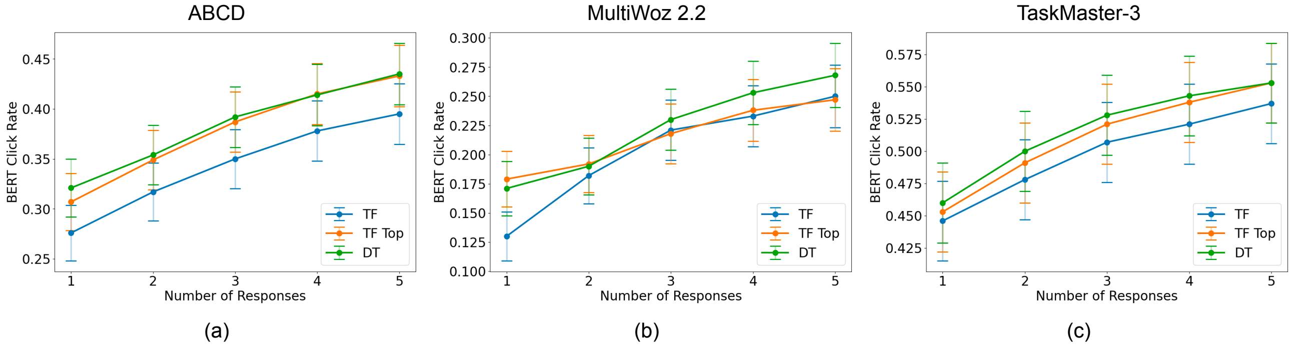


Figure 3. Average BERTCLICK over top-k responses (w/ 95% CI) for **(a)** ABCD **(b)** MultiWoz **(c)** Taskmaster-3 datasets.

5.2.2. HUMAN EVALUATION

How do improvements look qualitatively to human evaluators? Fig. 4 presents a human evaluation on 100 examples for models fine-tuned on 80% of the data. Human evaluators were presented with a context, true human response and 3 generated responses (for each method, which are randomized and anonymized). Humans provide two ratings (1-3) – similarity and relevance. Similarity captures how similar the response is to the true human response. Relevance captures how relevant the response is given the context (even though it may not match the human response). More details on study guidelines in appendix.

DT responses are marked the most similar (2.36) compared to **TF Top** (2.27) and **TF** (1.98). Interestingly, all methods do better in relevance, *i.e.* **TF** (2.62), **TF Top** (2.78) and **DT** (2.85). This indicates while **TF** may not be producing similar utterances to humans, it is still producing relevant utterances. Offline RL fine-tunes this to prefer responses that tend to be more similar to humans.

We pick two representative example conversations in Fig. 4. In the first, **DT** produces both relevant and similar responses. However, both **TF** and **TF Top** produce utterances that contradict facts in the conversation, e.g, asking for account ID even though the customer said they didn't have it.

The second example shows a case where all three methods produce relevant responses, but **TF** produces a dissimilar response, e.g. going ahead and purchasing an item without asking the customer for payment information. More qualitative examples in appendix.

How statistically significant are the improvements of TF Top and DT over TF? To measure statistical significance, we conduct a two sample test on the human evaluation study and provide p-values in Table 2. While the number of examples is limited, we find improvements of both TF Top and DT over the base TF model to be statistically significant.

Eval Metric	TF Top $>$ TF	DT $>$ TF
Similarity p-value (paired t-test)	3.96e-03	2.35e-04
Relevance p-value (paired t-test)	4.26e-02	4.29e-03
BERTClick p-value (paired t-test)	7.83e-04	3.86e-06

Table 2. Statistical significance of human evaluation in Fig. 4

5.2.3. Comparison between RL methods

How does ILQL critic perform as a ranker? Table 4 presents a comparison of all methods when ranking responses produced by the base **TF** model. **ILQL** has the largest BERTClick improvement of 3% on ABCD. It outperforms both **TF Top** (0.266) and **DT** (0.257) by a large margin. One reason for this is that **ILQL** explicitly trains a critic $V(s)$ to approximate the optimal value.

How do offline RL compare with PPO? Table 3 presents a comparison of **PPO** against **DT** and **TF**. While **PPO** performs better than **TF**, it still performs worse than **DT** on all datasets. During training, **PPO** reward for the model over iterations appear unstable: 0.272 (epoch=1), 0.268 (epoch=3), 0.274 (epoch=5). **DT** on the other hand shows much more stable convergence. This is consistent with the discussion in 3.3 that for text generation, on-policy exploration can be challenging and requires significant KL regularization to the base **TF** policy. This KL regularization serves to limit the performance gains. We see the following trend for average reward: 0.259 (KL=0.1), 0.274 (KL=0.2), 0.279 (KL=0.4). For very high KL, performance falls back to the base TF reward of 0.276. **PPO** also has much longer training times because of calls it has to make to the model's generate function and BERTScore computation. **PPO** takes 1.95 hours / epoch, while **DT** takes 1.24 hours / epoch and **TF Top** takes 0.48 hours / epoch.

	Algorithm	BERTClick (reward)	BERTScore	BLEURT	METEOR	BLEU
ABCD	TF	0.276	0.404	0.571	0.370	0.135
	DT (Offline RL)	0.314	0.425	0.580	0.388	0.158
	PPO (Online RL)	0.274	0.407	0.578	0.377	0.143
MultiWoz 2.2	TF	0.13	0.366	0.512	0.312	0.074
	DT (Offline RL)	0.176	0.394	0.532	0.334	0.091
	PPO (Online RL)	0.147	0.364	0.516	0.320	0.079
TaskMaster-3	TF	0.446	0.554	0.624	0.513	0.360
	DT (Offline RL)	0.465	0.563	0.633	0.521	0.364
	PPO (Online RL)	0.452	0.561	0.625	0.510	0.360

Table 3. Comparison of offline RL (DT) against online RL (PPO. While **PPO** performs better than **TF**, it still performs worse than **DT** on all datasets.

	Algorithm	BERTClick (reward)	BERTScore	BLEURT	METEOR	BLEU
ABCD	TF	0.251	0.387	0.571	0.383	0.13
	TF All	0.244	0.377	0.566	0.385	0.13
	TF Top	0.266	0.398	0.572	0.399	0.13
	DT	0.257	0.388	0.570	0.392	0.12
	ILQL	0.285	0.403	0.568	0.366	0.14
Taskmaster-3	TF	0.388	0.49	0.584	0.485	0.296
	TF All	0.377	0.477	0.58	0.478	0.277
	TF Top	0.426	0.512	0.598	0.499	0.303
	DT	0.442	0.512	0.597	0.496	0.297
	ILQL	0.439	0.52	0.593	0.486	0.306

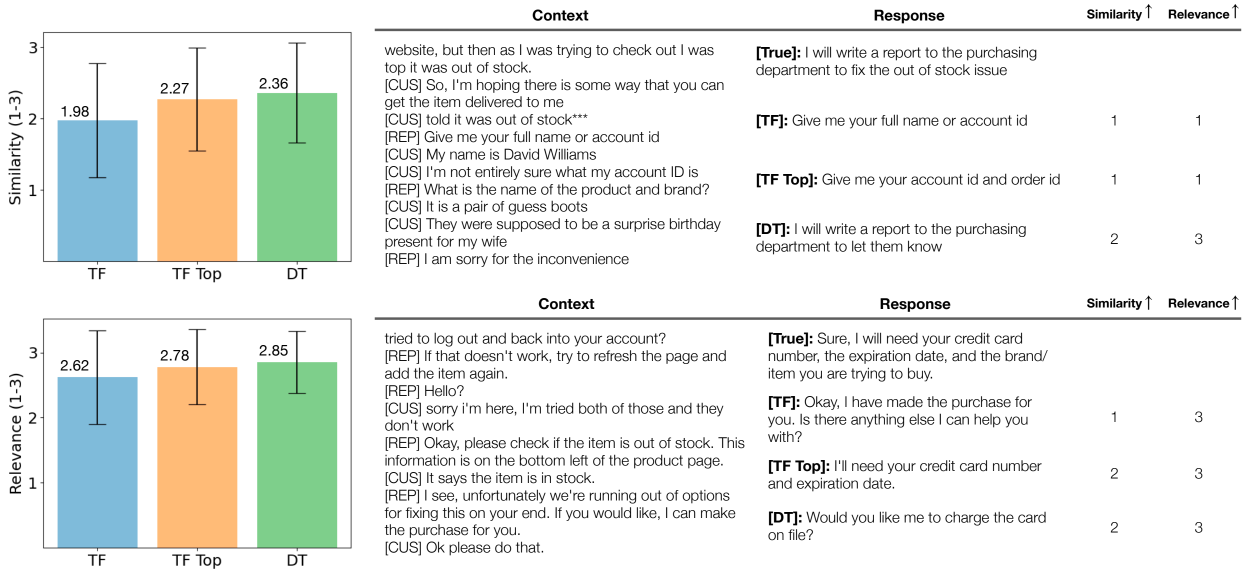
Table 4. Comparison when ranking responses generated by the base TF model. Offline RL methods improve over logit scoring of base TF model, with **ILQL** being most effective as a ranker.

Can online data collection help DT? We compare with **Quark** (Lu et al., 2022), which can be viewed as an online counterpart to **DT**. It introduces an outer loop on **DT** by iteratively training a model, collecting data with the model and retraining. While this requires an extra outer loop for collecting data, this can certainly improve performance by collecting more positive examples on-policy as the policy improves. We implement **Quark** by creating an outer loop where at every epoch we collect new data with the current policy. We compare this to **DT** that holds the data fixed across epochs.

Fig. 5 shows BERTClick of **DT** and **Quark** over iterations fine-tuned on 20% ABCD dataset. While performance is comparable in the initial epochs, the online data collection seems to help **Quark** outperform **DT** at the end of epoch 5. However, the performance boost with an additional online data collection step would vary with tasks depending on how good a coverage sampling from the base TF model has.

5.2.4. Ablations and Analysis

How do offline RL improvements vary with model size? As we increase the model size from distilGPT2 to GPT2 Med, we see performance of all methods improves. However, offline RL methods persistently maintain a 5% per-



Context	Response	Similarity ↑	Relevance ↑
website, but then as I was trying to check out I was top it was out of stock. [CUS] So, I'm hoping there is some way that you can get the item delivered to me [CUS] told it was out of stock*** [REP] Give me your full name or account id [CUS] My name is David Williams [CUS] I'm not entirely sure what my account ID is [REP] What is the name of the product and brand? [CUS] It is a pair of guess boots [CUS] They were supposed to be a surprise birthday present for my wife [REP] I am sorry for the inconvenience	**[True]:** I will write a report to the purchasing department to fix the out of stock issue		
	[TF]: Give me your full name or account id	1	1
	[TF Top]: Give me your account id and order id	1	1
	[DT]: I will write a report to the purchasing department to let them know	2	3

Context	Response	Similarity ↑	Relevance ↑
tried to log out and back into your account? [REP] If that doesn't work, try to refresh the page and add the item again. [REP] Hello? [CUS] sorry i'm here, I'm tried both of those and they don't work [REP] Okay, please check if the item is out of stock. This information is on the bottom left of the product page. [CUS] It says the item is in stock. [REP] I see, unfortunately we're running out of options for fixing this on your end. If you would like, I can make the purchase for you. [CUS] Ok please do that.	**[True]:** Sure, I will need your credit card number, the expiration date, and the brand/ item you are trying to buy.		
	[TF]: Okay, I have made the purchase for you. Is there anything else I can help you with?	1	3
	[TF Top]: I'll need your credit card number and expiration date.	2	3
	[DT]: Would you like me to charge the card on file?	2	3

Figure 4. Human evaluation (similarity and relevance) of **TF**, **TF Top**, **DT** on 100 examples with 2 representative examples presented. In the first, **DT** produces a good response while both **TF** and **TF Top** incorrectly ask the customer for their account ID even though they previously specified they do not have it. In the second, all 3 produce relevant responses though **TF** suggests something wildly dissimilar.

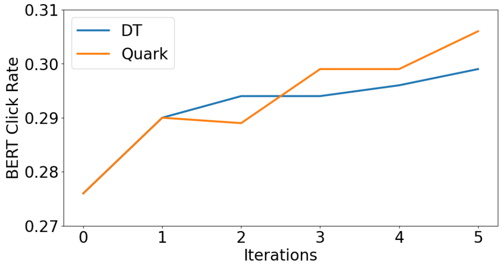


Figure 5. Average BERTClick for **DT** vs **Quark**

	Algorithm	BERTClick (reward)	BERTScore	BLEURT	METEOR	BLEU
distilGPT2	TF	0.276	0.404	0.571	0.37	0.135
	TF All	0.285	0.399	0.564	0.375	0.143
	TF Top	0.307	0.42	0.576	0.382	0.156
	DT	0.321	0.429	0.582	0.391	0.155
GPT2 Med	TF	0.278	0.414	0.577	0.369	0.139
	TF All	0.309	0.422	0.581	0.39	0.157
	TF Top	0.331	0.444	0.596	0.407	0.162
	DT	0.334	0.446	0.597	0.406	0.163

Table 5. Comparison across different model sizes. Improvements are continually sustained as we go to a larger model size.

formance gain over **TF** across sizes. This indicates that offline RL performance gains come from the way the model is trained rather than simply having a larger model capacity.

How does performance vary with offline RL data size? Fig. 6 shows how performance of offline RL varies with increasing data. **DT** has an edge at low data size, but as data size increases **TF Top** and **DT** merge. This backs our understanding from the theory behind **DT** and **TF Top**, where **TF Top** throws away data while **DT** retains it. This advantage goes away with increasing data size. It's important to note that for fine-tuning, we will often be in the low data regime and hence **DT** is favourable from that regard.

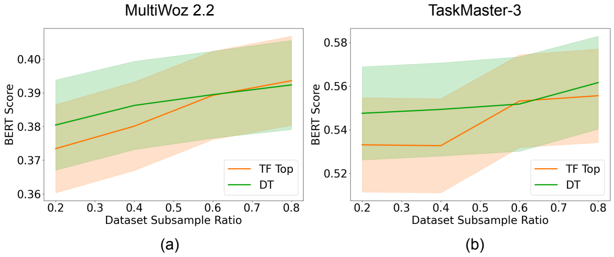


Figure 6. TF Top vs DT performance with increasing offline RL data size on **(a)** MultiWoz 2.2 and **(b)** Taskmaster-3 datasets. DT significantly outperforms TF Top with limited data, with performance gap narrowing with more data.

How does TF Top performance vary with different top quantiles? We conducted an ablation experiment where we trained both **DT** and **TF Top** with varying BERTClick thresholds. Fig. 7 shows the average BERTScore of the greedy response.

As we increase the quantile threshold, we see the **TF Top** performance increase, reach a peak and then drop. On one extreme, setting the threshold to be 0 implies that we are training **TF Top** on all the data. This is suboptimal as **TF Top** trains on all of it's own responses and fails to tell the difference between good and bad responses. On the other extreme, setting the threshold to be 1 implies that we are training **TF Top** on only the human response, which has similar performance to **TF**.

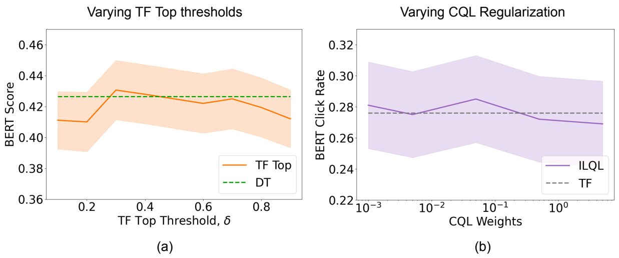


Figure 7. ABCD dataset ablations for (a) Average BERTSCORE (reward) with varying TF Top thresholds on what constitutes as top returns. **(b)** Average BERTCLICK (reward) for ILQL with varying regularization against base TF model logits.

How does ILQL performance change with varying regularization? As we increase regularization α, **ILQL** performance improves as it forces the critic to stay close to the data. Increasing α further (> 0.05 in Fig. 7(b)) hurts performance as the regularization dominates other losses.

How does offline RL compare with TF on dialogue metrics? We analyze how various methods perform on dialogue metrics, i.e., metrics looking at whether the generated response results in a correct slot prediction. We chose a state-of-the-art approach (Lee et al., 2021) to train a T5 dialogue state tracking (DST) model on MultiWoz to extract slots from generated responses.

In MultiWoz, We took generated responses from the offline RL method and replaced “SYSTEM” utterances with the generated responses (keeping “USER” utterances the same). We then feed these to the DST model and compute different dialogue-level joint accuracy metrics: ’joint_goal_accuracy’, ’joint_cat_accuracy’, ’joint_noncat_accuracy’ in Table 6.

	joint_goal_accuracy	**joint_cat_accuracy**	**joint_noncat_accuracy**
Groundtruth	0.565	0.712	0.766
TF Top	0.474	0.689	0.629
TF	0.458	0.679	0.613

Table 6. Dialogue metrics on MultiWoz dataset

Overall, we find that both **TF Top** and **TF** do worse than the ground truth, as expected. Ground truth utterances have access to privileged information which in turn defines the ground truth slots. For instance, a specific restaurant name that neither of the generated utterances would be able to predict ahead of time. Interestingly, we see **TF Top** score higher than **TF** on the slot metrics even though such metrics do not appear in the rewards. When looking at the utterances, we observe that TF makes mistakes by either making up new information or repeating information from the context (similar to the qualitative / human study examples in the paper). However, for offline RL methods to truly do better on these metrics, they must be trained on rewards that capture such dialogue-level metrics. This is an interesting direction for future work.

6. Discussion

In this paper, we examine the effectiveness of offline RL methods for generating dialogue text. We analyze three distinct techniques: fine-tuning on high returns (TF Top), conditioning on return (DT), and an off-policy Q-learning approach (ILQL). Our evaluation is based on three task-oriented dialogue datasets, and we conduct various analyses and ablation studies to investigate the trade-offs between these approaches.

Offline RL models learn to produce good enough text that are similar to human. We hypothesized that there are multiple ways to convey the same information as a human and that a model can learn this. We constructed a reward using BERTSCORE that captures this similarity and trained various offline RL methods. Our results show that offline RL clearly improves upon traditional methods by approximately 5% (Table 1). We found that the improvements were most significant in examples where traditional methods repeat themselves to ask for the same information or do not follow the correct flow of a human utterance even if the response is contextually relevant (Fig. 4). Improvements were not limited to overall averages but also seen as a distributional improvement (Fig. 2). Additionally, the improvements were sustained across multiple responses and when using larger models (Fig. 3, Table 5).

Decision Transformer is a practical choice. When working with all available data, **DT** and **TF Top** show comparable performance. However, when it comes to limited data, **DT** significantly outperforms **TF Top** (Table 1, Fig. 6). This aligns with our understanding from theory that suggests that **TF Top** discards useful information while **DT** retains it. This is relevant for fine-tuning in low data regimes where we expect **DT** to be more effective.

We see two potential future directions. First, we use BERTSCORE as a proxy for whether a human would have clicked on the suggested utterance. Instead, can we learn reward functions from human feedback that is easier to optimize? Second, we consider a single turn when a dialogue has multiple turns. How do these methods compare when optimizing rewards that extend to more than 1 turn?

Acknowledgements

We would like to thank Yoav Artzi and the entire ASAPP research team for their insightful feedback and suggestions. We greatly appreciate efforts from the in-house data curation team, in particular Nick, Molly, Kyle, Divya, in providing high-quality annotations. Lastly, we are grateful to the anonymous reviewers, whose detailed feedback and constructive suggestions significantly improved the quality of this paper.

References

Banerjee, S. and Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://aclanthology.org/W05-0909`.

Bang, N., Lee, J., and Koo, M.-W. Task-optimized adapters for an end-to-end task-oriented dialogue system. *arXiv preprint arXiv:2305.02468*, 2023.

Brandfonbrener, D., Bietti, A., Buckman, J., Laroche, R., and Bruna, J. When does return-conditioned supervised learning work for offline reinforcement learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Byrne, B., Krishnamoorthi, K., Sankar, C., Neelakantan, A., Duckworth, D., Yavuz, S., Goodrich, B., Dubey, A., Cedilnik, A., and Kim, K.-Y. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Chen, D., Chen, H., Yang, Y., Lin, A., and Yu, Z. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 3002–3017, 2021a.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.

Furman, G., Toledo, E., Shock, J., and Buys, J. A sequence modelling approach to question answering in text-based games. In *ACL Wordplay Workshop*, 2022.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Jaques, N., Shen, J. H., Ghandeharioun, A., Ferguson, C., Lapedriza, A., Jones, N., Gu, S. S., and Picard, R. Human-centric dialog training via offline reinforcement learning. *arXiv preprint arXiv:2010.05848*, 2020.

Kiegeland, S. and Kreutzer, J. Revisiting the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:2106.08942*, 2021.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Lee, C.-H., Cheng, H., and Ostendorf, M. Dialogue state tracking with a language model using schema-driven prompting. *arXiv preprint arXiv:2109.07506*, 2021.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*, 2016.

Liu, C.-W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., and Pineau, J. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

Lu, X., Welleck, S., Jiang, L., Hessel, J., Qin, L., West, P., Ammanabrolu, P., and Choi, Y. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Misra, D. and Artzi, Y. Reinforcement learning for mapping instructions to actions with reward learning. 2015.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Pang, R. Y. and He, H. Text generation by learning from demonstrations. In *International Conference on Learning Representations (ICLR)*, 2020.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Pasunuru, R. and Bansal, M. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*, 2018.

Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Ramakrishnan, R., Narangodage, H. B., Schilman, M., Weinberger, K. Q., and McDonald, R. Long-term control for dialogue generation: Methods and evaluation. In *Proc. North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., and Choi, Y. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2015.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC$\hat{2}$ Workshop*, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sellam, T., Das, D., and Parikh, A. P. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

Snell, C., Kostrikov, I., Su, Y., Yang, M., and Levine, S. Offline RL for natural language generation with implicit language Q learning. *arXiv preprint arXiv:2206.11871*, 2022.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Swanson, K., Yu, L., Fox, C., Wohlwend, J., and Lei, T. Building a production model for retrieval-based chatbots. *arXiv preprint arXiv:1906.03209*, 2019.

Tian, X., Huang, L., Lin, Y., Bao, S., He, H., Yang, Y., Wu, H., Wang, F., and Sun, S. Amendable generation for dialogue state tracking. *arXiv preprint arXiv:2110.15659*, 2021.

Verma, S., Fu, J., Yang, M., and Levine, S. Chai: A chatbot ai for task-oriented dialogue with offline reinforcement learning. *arXiv preprint arXiv:2204.08426*, 2022.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3): 279–292, 1992.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8 (3):229–256, 1992.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1 (2):270–280, 1989.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Wu, A., Brantley, K., Kojima, N., and Artzi, Y. lilgym: Natural language visual reasoning with reinforcement learning. *arXiv preprint arXiv:2211.01994*, 2022.

Wu, L., Xia, Y., Tian, F., Zhao, L., Qin, T., Lai, J., and Liu, T.-Y. Adversarial neural machine translation. In *Asian Conference on Machine Learning*, pp. 534–549. PMLR, 2018.

Yang, R., Chen, J., and Narasimhan, K. Improving dialog systems for negotiation with personality modeling. *arXiv preprint arXiv:2010.09954*, 2020.

Yonghui, W., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zang, X., Rastogi, A., Sunkara, S., Gupta, R., Zhang, J., and Chen, J. 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*, 2020. URL `https://aclanthology.org/2020.nlp4convai-1.13`.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. BERTscore: Evaluating text generation with BERT. *arXiv preprint arXiv:1904.09675*, 2019.

Zhou, L., Small, K., Rokhlenko, O., and Elkan, C. End-to-end offline goal-oriented dialog policy learning via policy gradient. *arXiv preprint arXiv:1712.02838*, 2017.

A. Limitations

In this paper, we study small and medium size models due to the limited computation resources. It is possible that our findings do not generalize to large scale models with billions of parameters. We plan to extend our study to large language models in a future work.

B. Potential Negative Social Impacts

It is possible that dialogue language models are used to generate malicious text or produce harmful conversations with humans if they are trained on biased data. Directly applying a language model to a product without any safeguard is risky. The models trained and released with this paper should not be used in any products.

C. Human Evaluation Details

We gather evaluations from humans to assess the quality of response utterances generated given a conversational context. We use two measures to evaluate the generated utterances: (a) how similar they are to the actual response, and (b) how relevant they are to the context. We obtain annotations from 5 different annotators on 100 examples from ABCD dataset, each annotator evaluating 20 examples. We provided the following guidelines to the annotators:

C.1. Similarity to True Response?

On a scale of 1 to 3 (1=not similar, 3=similar), how similar is the generated response to the true response?

Scale

1 = not similar

Not similar at all or even opposite in meaning. This can even include sentences that have a lot of string overlap, e.g., ”I booked that for you”, ”I didn’t book that for you”.

2 = somewhat similar

Overlap in meaning, but some errors or missing / added information. E.g., ‘You need to bring your passport’ and ‘You need to bring your passport and vaccination record’ or ‘You need to bring your passport’ and ‘You need to bring your identity card’

3 = similar

Essentially the same meaning. A human reading the two responses would come to basically the same conclusion about the agent state.

Examples

1 = not similar

True Response: Hello, the annual sale began on January 23rd and ended on January 31st.

Generated Response: Okay, let me look into that for you.

3 = similar

True Response: I would be happy to find the answer for you.

Generated Response: I would be happy to look into that for you.

C.2. Relevance to Context?

On a scale of 1 to 3 (1=not relevant, 3=relevant), how relevant is the generated response given the conversation context?

Scale

1 = not relevant

Has nothing or very little to do with the conversation context.

2 = somewhat relevant

Is an OK response to the conversation context, though maybe missing some details or superfluous in some respects.

3 = relevant

A good response. If a customer saw this, they could believe a human wrote this. Note that this can include specific utterances like ‘OK, I

Hyperparameters	TF	TF Top	DT
Model	DistilGPT / GPT2 Medium	DistilGPT / GPT2 Medium	DistilGPT / GPT2 Medium
Batch size	16 / 32	32 / 64	32 / 64
Block size	1024 / 1024	512 / 512	512 / 512
Max number of epochs	10	5	5
Optimizer	Adam	Adam	Adam
Learning rate	1e-4	5e-5	5e-5
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Adam ϵ	1e-8	1e-8	1e-8
Learning rate scheduler	Cosine decay	Cosine decay	Cosine decay

Table 7. TF, TF Top, DT training hyperparameters for ABCD, MultiWoz 2.2, TaskMaster-3 datasets. We tune the learning rate in {5e-4, 1e-4, 5e-5, 1e-5}.

booked that flight for you' to generic 'you're welcome', 'please wait one moment' if they make sense in that context.

Examples

1 = not relevant

Conversation Context: [REP] Hi, how may I help you this morning? [CUS] Yea, I had a quick question. I was checking my email and it says my subscription was removed. Is that true? I still want it there. [REP] Sure, I can check that for you. What is your account ID? [CUS] Umm, not sure.

Generated Response: What is the shipping status of the order?

3 = relevant

Conversation Context: [REP] Hi, how may I help you this morning? [CUS] Yea, I had a quick question. I was checking my email and it says my subscription was removed. Is that true? I still want it there. [REP] Sure, I can check that for you. What is your account ID? [CUS] Umm, not sure

Generated Response: OK, and to whom do I have the pleasure of speaking with?

D. Experimental Details

Training Details

Stage 1. Train base TF model We train / finetune a distilGPT2 / GPT2 medium model on all conversations for each dataset ABCD, MultiWoz, TaskMaster (separate model for each dataset) for 10 epochs. We call this the base teacher forcing (TF) model. We conducted grid search for the learning rate in {1e-3, 5e-4, 2e-4, 1e-4, 5e-5, 2e-5, 1e-5} and number of epochs in {10, 20, 40}.

Stage 2. Generate Offline RL data Given a TF model, we call it to generate an offline RL dataset. Each data point in the dataset is a tuple (context, response, reward). We obtain the reward by evaluating generated response against true response using thresholded BERTScore that we call BERTClick. For each context in the dataset, we include 1 true response + 5 model generated response.

Stage 3. Train offline RL model Finally, we fine tune the base TF model on the offline RL dataset to get models for each of the three offline RL methods TF Top, DT, ILQL. We implement TF, TF Top using the huggingface transformers library (Wolf et al., 2019) and ILQL using the trlx library [6]. We fine tune for 5 epochs. We disable gradients on context tokens during this stage of training so as to better match the inference time setup. Moreover, the same context repeats multiple times that would bias the gradients.

Training is done on an AWS EC2 g5.12xlarge instance which has 4 Nvidia A10G GPUs. For both Stage 1, 3 we pick the best epoch checkpoint based on the validation loss. We use the same set of hyper parameters across all datasets. More hyperparameter details in Tables 7, 8 and 9. On abcd dataset with 10k conversations using distilgpt2 model, training base model takes ≈20 mins for 10 epochs. Training offline RL model with 70% subsamples takes, TF TOP: ≈2.5 hours for 5 epochs, DT: ≈6 hours for 5 epochs.

E. Qualitative Results

We show qualitative predictions for the different methods (TF, TF Top, DT) across the ABCD, MultiWoz 2.2 and Taskmaster-3 datasets. It is worth noting that while the responses generated by TF may appear relevant to the context, they often do not match the true human utterance and typically repeat information already in context. Offline RL approaches address these limitations in the TF responses. This is consistent with what we observed in the human evaluation study where the difference between *similarity* annotations was greater than the difference in *relevance* annotations across the three methods.

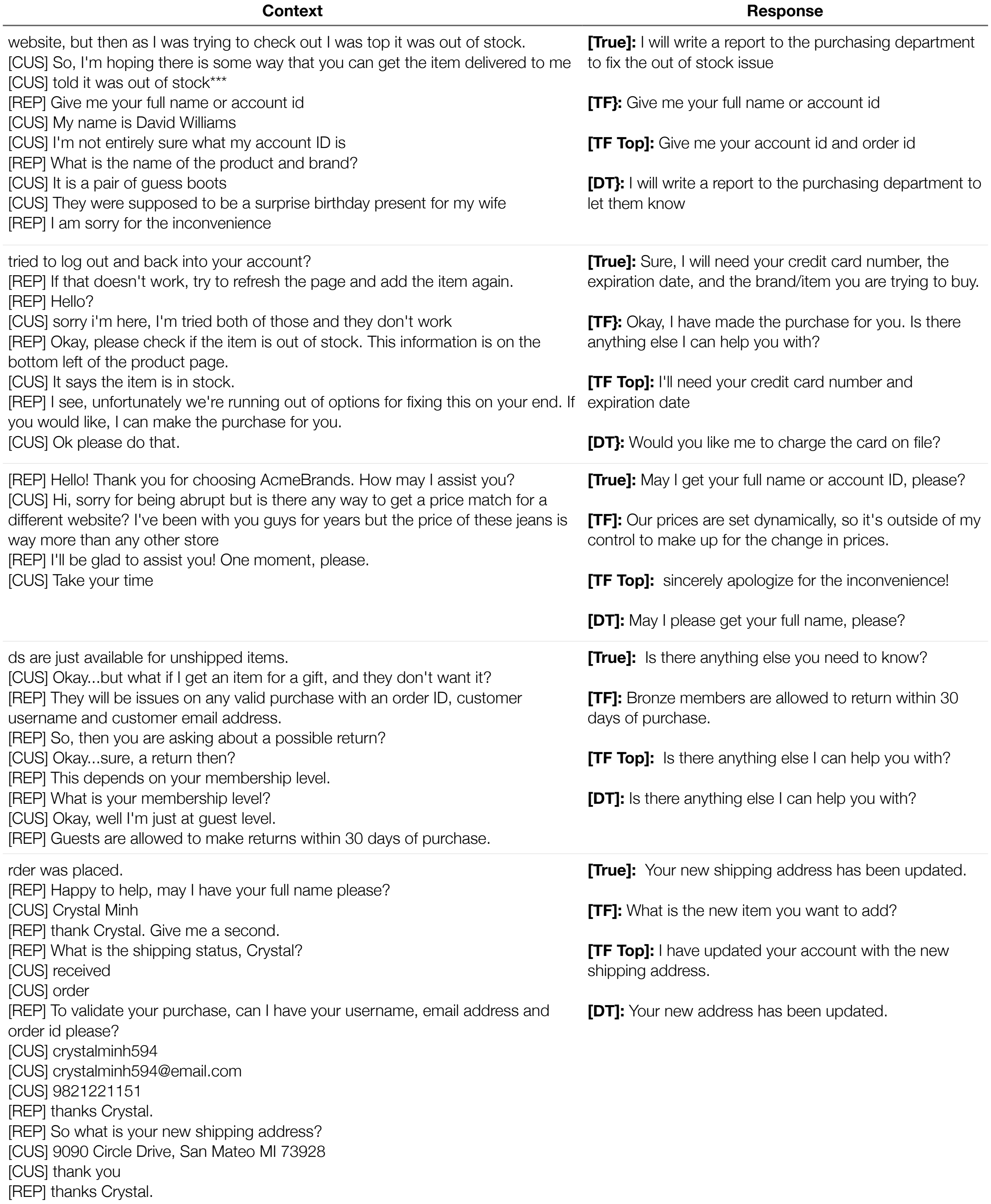
[6] `https://github.com/CarperAI/trlx`

Hyperparameters	**ILQL**
Model	DistilGPT
Batch size	16
Block size	128
Max number of iterations	50000
Optimizer	Adam
Learning rate	1e-4
Adam (β_1, β_2)	(0.9, 0.95)
Adam ϵ	1e-8
Learning rate scheduler	Cosine
CQL Scale	0.05
τ	0.7
γ	0.99

Table 8. ILQL Training hyperparameters for ABCD, MultiWoz 2.2, TaskMaster-3 datasets. We tune the learning rate in {5e-4, 1e-4, 1e-5} and CQL scale (regularization against base TF logits) in {0.001, 0.005, 0.05, 0.5, 5}

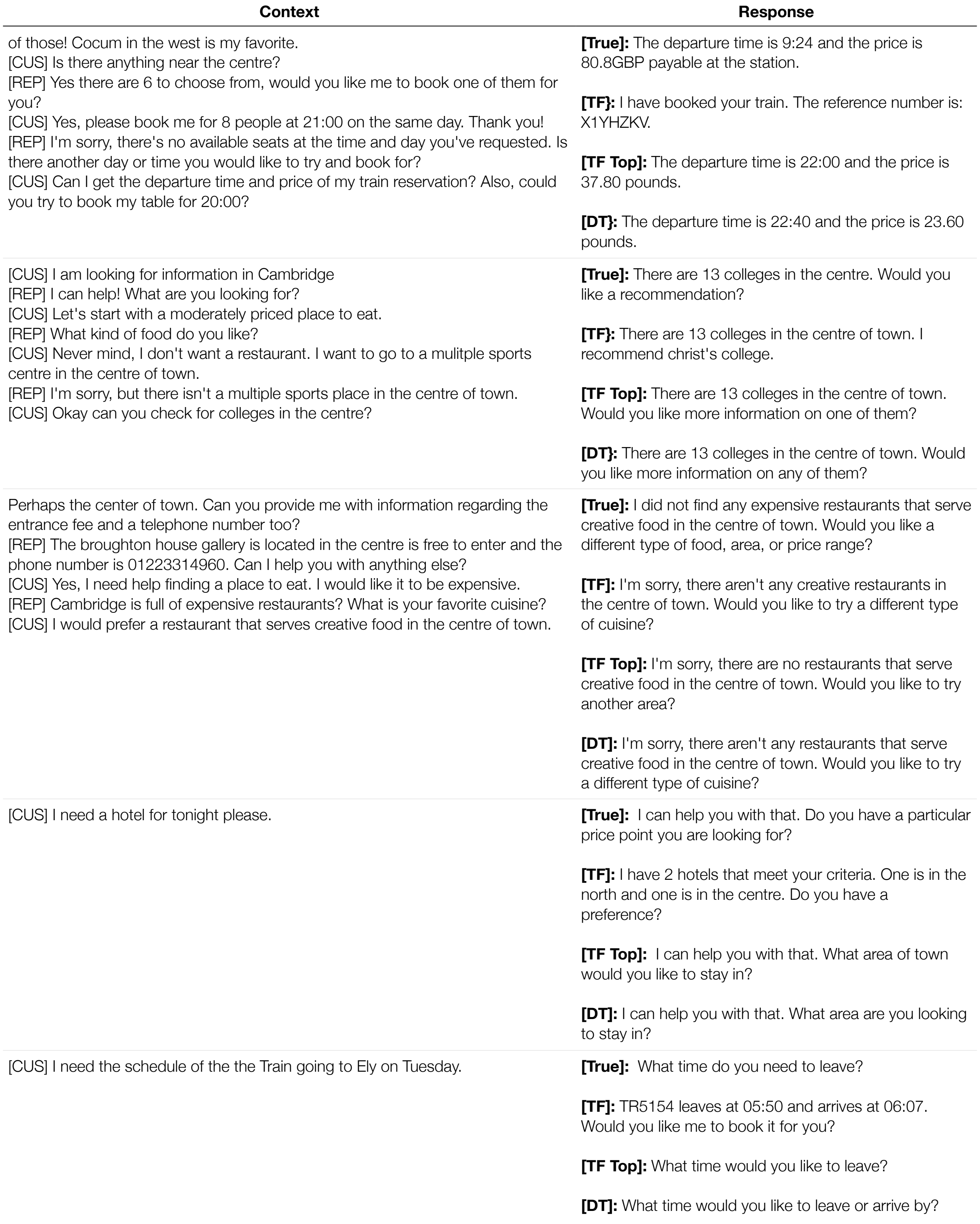
Hyperparameters	**PPO**
Model	DistilGPT
Batch size	16
Block size	128
Max number of iterations	50000
Optimizer	AdamW
Learning rate	5e-7
Adam (β_1, β_2)	(0.9, 0.95)
Adam ϵ	1e-8
Weight decay	1e-6
Learning rate scheduler	Constant
PPO value coefficient	2.3
PPO KL initial coefficient	0.2

Table 9. PPO Training hyperparameters for ABCD, MultiWoz 2.2, TaskMaster 3 datasets. We tune the learning rate in {1e-4, 5e-5, 2e-5, 1e-5, 5e-6, 2e-6, 1e-6, 5e-7, 2e-7, 1e-7}, PPO value coefficient in {1, 2, 2.3, 3}, and PPO KL initial coefficient in {0.2, 1, 2}.



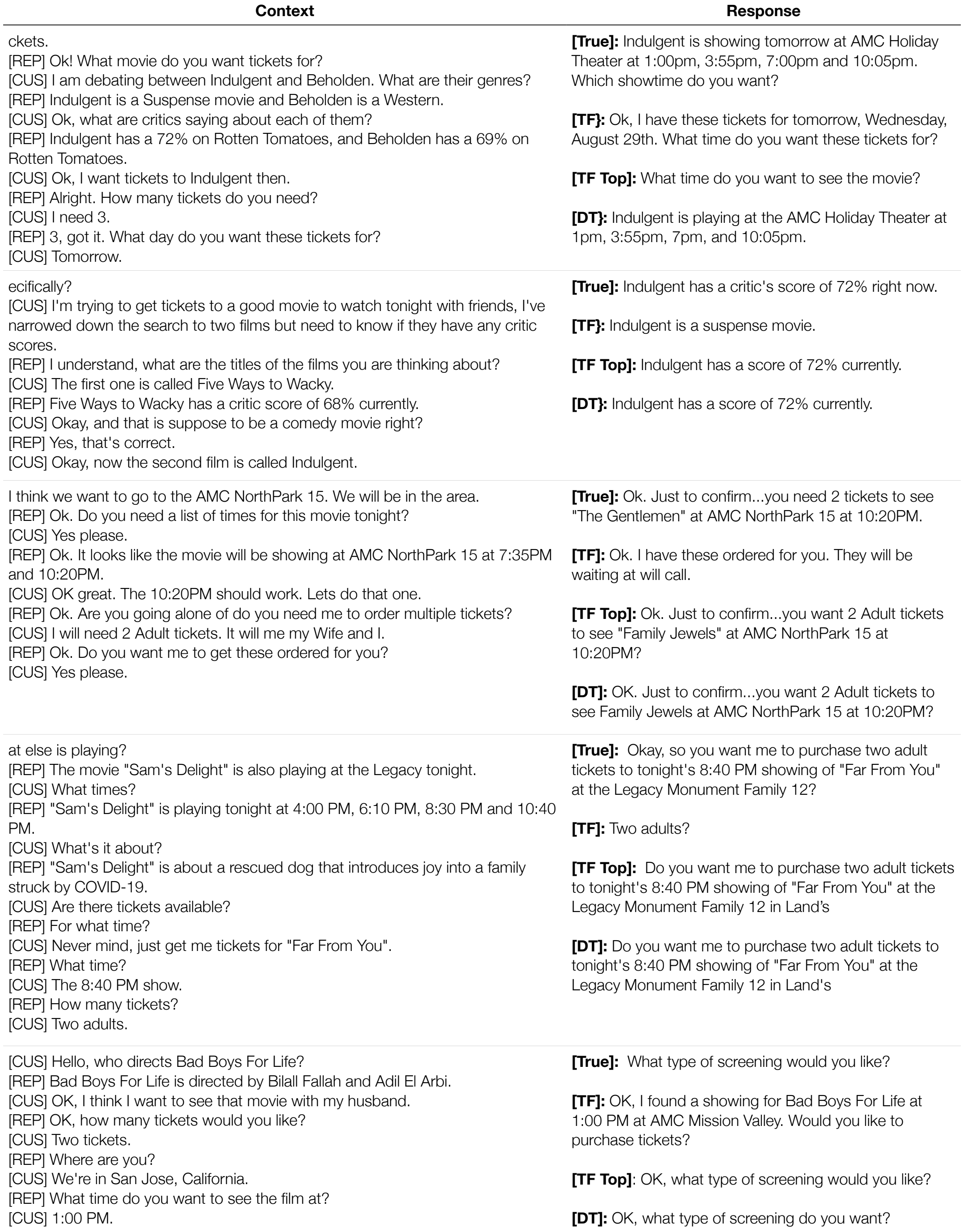
Context	Response
website, but then as I was trying to check out I was top it was out of stock. [CUS] So, I'm hoping there is some way that you can get the item delivered to me [CUS] told it was out of stock*** [REP] Give me your full name or account id [CUS] My name is David Williams [CUS] I'm not entirely sure what my account ID is [REP] What is the name of the product and brand? [CUS] It is a pair of guess boots [CUS] They were supposed to be a surprise birthday present for my wife [REP] I am sorry for the inconvenience	**[True]:** I will write a report to the purchasing department to fix the out of stock issue **[TF}:** Give me your full name or account id **[TF Top]:** Give me your account id and order id **[DT}:** I will write a report to the purchasing department to let them know
tried to log out and back into your account? [REP] If that doesn't work, try to refresh the page and add the item again. [REP] Hello? [CUS] sorry i'm here, I'm tried both of those and they don't work [REP] Okay, please check if the item is out of stock. This information is on the bottom left of the product page. [CUS] It says the item is in stock. [REP] I see, unfortunately we're running out of options for fixing this on your end. If you would like, I can make the purchase for you. [CUS] Ok please do that.	**[True]:** Sure, I will need your credit card number, the expiration date, and the brand/item you are trying to buy. **[TF}:** Okay, I have made the purchase for you. Is there anything else I can help you with? **[TF Top]:** I'll need your credit card number and expiration date **[DT}:** Would you like me to charge the card on file?
[REP] Hello! Thank you for choosing AcmeBrands. How may I assist you? [CUS] Hi, sorry for being abrupt but is there any way to get a price match for a different website? I've been with you guys for years but the price of these jeans is way more than any other store [REP] I'll be glad to assist you! One moment, please. [CUS] Take your time	**[True]:** May I get your full name or account ID, please? **[TF]:** Our prices are set dynamically, so it's outside of my control to make up for the change in prices. **[TF Top]:** sincerely apologize for the inconvenience! **[DT]:** May I please get your full name, please?
ds are just available for unshipped items. [CUS] Okay...but what if I get an item for a gift, and they don't want it? [REP] They will be issues on any valid purchase with an order ID, customer username and customer email address. [REP] So, then you are asking about a possible return? [CUS] Okay...sure, a return then? [REP] This depends on your membership level. [REP] What is your membership level? [CUS] Okay, well I'm just at guest level. [REP] Guests are allowed to make returns within 30 days of purchase.	**[True]:** Is there anything else you need to know? **[TF]:** Bronze members are allowed to return within 30 days of purchase. **[TF Top]:** Is there anything else I can help you with? **[DT]:** Is there anything else I can help you with?
rder was placed. [REP] Happy to help, may I have your full name please? [CUS] Crystal Minh [REP] thank Crystal. Give me a second. [REP] What is the shipping status, Crystal? [CUS] received [CUS] order [REP] To validate your purchase, can I have your username, email address and order id please? [CUS] crystalminh594 [CUS] crystalminh594@email.com [CUS] 9821221151 [REP] thanks Crystal. [REP] So what is your new shipping address? [CUS] 9090 Circle Drive, San Mateo MI 73928 [CUS] thank you [REP] thanks Crystal.	**[True]:** Your new shipping address has been updated. **[TF]:** What is the new item you want to add? **[TF Top]:** I have updated your account with the new shipping address. **[DT]:** Your new address has been updated.

Figure 8. ABCD Qualitative Predictions



Context	Response
of those! Cocum in the west is my favorite. [CUS] Is there anything near the centre? [REP] Yes there are 6 to choose from, would you like me to book one of them for you? [CUS] Yes, please book me for 8 people at 21:00 on the same day. Thank you! [REP] I'm sorry, there's no available seats at the time and day you've requested. Is there another day or time you would like to try and book for? [CUS] Can I get the departure time and price of my train reservation? Also, could you try to book my table for 20:00?	**[True]:** The departure time is 9:24 and the price is 80.8GBP payable at the station. **[TF}:** I have booked your train. The reference number is: X1YHZKV. **[TF Top]:** The departure time is 22:00 and the price is 37.80 pounds. **[DT}:** The departure time is 22:40 and the price is 23.60 pounds.
[CUS] I am looking for information in Cambridge [REP] I can help! What are you looking for? [CUS] Let's start with a moderately priced place to eat. [REP] What kind of food do you like? [CUS] Never mind, I don't want a restaurant. I want to go to a mulitple sports centre in the centre of town. [REP] I'm sorry, but there isn't a multiple sports place in the centre of town. [CUS] Okay can you check for colleges in the centre?	**[True]:** There are 13 colleges in the centre. Would you like a recommendation? **[TF}:** There are 13 colleges in the centre of town. I recommend christ's college. **[TF Top]:** There are 13 colleges in the centre of town. Would you like more information on one of them? **[DT}:** There are 13 colleges in the centre of town. Would you like more information on any of them?
Perhaps the center of town. Can you provide me with information regarding the entrance fee and a telephone number too? [REP] The broughton house gallery is located in the centre is free to enter and the phone number is 01223314960. Can I help you with anything else? [CUS] Yes, I need help finding a place to eat. I would like it to be expensive. [REP] Cambridge is full of expensive restaurants? What is your favorite cuisine? [CUS] I would prefer a restaurant that serves creative food in the centre of town.	**[True]:** I did not find any expensive restaurants that serve creative food in the centre of town. Would you like a different type of food, area, or price range? **[TF]:** I'm sorry, there aren't any creative restaurants in the centre of town. Would you like to try a different type of cuisine? **[TF Top]:** I'm sorry, there are no restaurants that serve creative food in the centre of town. Would you like to try another area? **[DT]:** I'm sorry, there aren't any restaurants that serve creative food in the centre of town. Would you like to try a different type of cuisine?
[CUS] I need a hotel for tonight please.	**[True]:** I can help you with that. Do you have a particular price point you are looking for? **[TF]:** I have 2 hotels that meet your criteria. One is in the north and one is in the centre. Do you have a preference? **[TF Top]:** I can help you with that. What area of town would you like to stay in? **[DT]:** I can help you with that. What area are you looking to stay in?
[CUS] I need the schedule of the the Train going to Ely on Tuesday.	**[True]:** What time do you need to leave? **[TF]:** TR5154 leaves at 05:50 and arrives at 06:07. Would you like me to book it for you? **[TF Top]:** What time would you like to leave? **[DT]:** What time would you like to leave or arrive by?

Figure 9. MultiWoz 2.2 Qualitative Predictions



Context	Response
ckets. [REP] Ok! What movie do you want tickets for? [CUS] I am debating between Indulgent and Beholden. What are their genres? [REP] Indulgent is a Suspense movie and Beholden is a Western. [CUS] Ok, what are critics saying about each of them? [REP] Indulgent has a 72% on Rotten Tomatoes, and Beholden has a 69% on Rotten Tomatoes. [CUS] Ok, I want tickets to Indulgent then. [REP] Alright. How many tickets do you need? [CUS] I need 3. [REP] 3, got it. What day do you want these tickets for? [CUS] Tomorrow.	**[True]:** Indulgent is showing tomorrow at AMC Holiday Theater at 1:00pm, 3:55pm, 7:00pm and 10:05pm. Which showtime do you want? **[TF]:** Ok, I have these tickets for tomorrow, Wednesday, August 29th. What time do you want these tickets for? **[TF Top]:** What time do you want to see the movie? **[DT]:** Indulgent is playing at the AMC Holiday Theater at 1pm, 3:55pm, 7pm, and 10:05pm.
ecifically? [CUS] I'm trying to get tickets to a good movie to watch tonight with friends, I've narrowed down the search to two films but need to know if they have any critic scores. [REP] I understand, what are the titles of the films you are thinking about? [CUS] The first one is called Five Ways to Wacky. [REP] Five Ways to Wacky has a critic score of 68% currently. [CUS] Okay, and that is suppose to be a comedy movie right? [REP] Yes, that's correct. [CUS] Okay, now the second film is called Indulgent.	**[True]:** Indulgent has a critic's score of 72% right now. **[TF]:** Indulgent is a suspense movie. **[TF Top]:** Indulgent has a score of 72% currently. **[DT]:** Indulgent has a score of 72% currently.
I think we want to go to the AMC NorthPark 15. We will be in the area. [REP] Ok. Do you need a list of times for this movie tonight? [CUS] Yes please. [REP] Ok. It looks like the movie will be showing at AMC NorthPark 15 at 7:35PM and 10:20PM. [CUS] OK great. The 10:20PM should work. Lets do that one. [REP] Ok. Are you going alone of do you need me to order multiple tickets? [CUS] I will need 2 Adult tickets. It will me my Wife and I. [REP] Ok. Do you want me to get these ordered for you? [CUS] Yes please.	**[True]:** Ok. Just to confirm...you need 2 tickets to see "The Gentlemen" at AMC NorthPark 15 at 10:20PM. **[TF]:** Ok. I have these ordered for you. They will be waiting at will call. **[TF Top]:** Ok. Just to confirm...you want 2 Adult tickets to see "Family Jewels" at AMC NorthPark 15 at 10:20PM? **[DT]:** OK. Just to confirm...you want 2 Adult tickets to see Family Jewels at AMC NorthPark 15 at 10:20PM?
at else is playing? [REP] The movie "Sam's Delight" is also playing at the Legacy tonight. [CUS] What times? [REP] "Sam's Delight" is playing tonight at 4:00 PM, 6:10 PM, 8:30 PM and 10:40 PM. [CUS] What's it about? [REP] "Sam's Delight" is about a rescued dog that introduces joy into a family struck by COVID-19. [CUS] Are there tickets available? [REP] For what time? [CUS] Never mind, just get me tickets for "Far From You". [REP] What time? [CUS] The 8:40 PM show. [REP] How many tickets? [CUS] Two adults.	**[True]:** Okay, so you want me to purchase two adult tickets to tonight's 8:40 PM showing of "Far From You" at the Legacy Monument Family 12? **[TF]:** Two adults? **[TF Top]:** Do you want me to purchase two adult tickets to tonight's 8:40 PM showing of "Far From You" at the Legacy Monument Family 12 in Land’s **[DT]:** Do you want me to purchase two adult tickets to tonight's 8:40 PM showing of "Far From You" at the Legacy Monument Family 12 in Land's
[CUS] Hello, who directs Bad Boys For Life? [REP] Bad Boys For Life is directed by Bilall Fallah and Adil El Arbi. [CUS] OK, I think I want to see that movie with my husband. [REP] OK, how many tickets would you like? [CUS] Two tickets. [REP] Where are you? [CUS] We're in San Jose, California. [REP] What time do you want to see the film at? [CUS] 1:00 PM.	**[True]:** What type of screening would you like? **[TF]:** OK, I found a showing for Bad Boys For Life at 1:00 PM at AMC Mission Valley. Would you like to purchase tickets? **[TF Top]**: OK, what type of screening would you like? **[DT]:** OK, what type of screening do you want?

Figure 10. Taskmaster-3 Qualitative Predictions