Improving Contextual Query Rewrite for Conversational AI Agents through User-preference Feedback Learning

Zhongkai Sun[1] **Yingxue Zhou**[1] **Jie Hao**[1] **Xing Fan**[1]
Yanbin Lu[1] **Chengyuan Ma**[1] **Wei Shen**[1] **Chenlei Guo**[1]
[1]Amazon Alexa AI
{zhongkas,zyingxue,jieha,fanxing,luyanbin,mchengyu,sawyersw,guochenl}@amazon.com

Abstract

Contextual query rewriting (CQR) is a crucial component in Conversational AI agents, leveraging the contextual information from previous user-agent conversations to improve the comprehension of current user intent. However, traditional CQR methods often concentrate on supervised fine-tuning only, neglecting the opportunities to learn from user feedback to align with user preferences. Inspired by recent advances in learning from human feedback (LHF), this paper proposes a novel Preference Aligned Contextual Query Rewriting (PA-CQR) framework to enhance the CQR model's capability in generating user preference-aligned rewrites. This paper also investigates the efficacy of various state-of-the-art feedback learning algorithms on the CQR task, and proposes a novel Dynamic Direct Preference Optimization (Dynamic DPO) algorithm to better adapt the DPO algorithm to large-scale CQR training. Experiments on large-scale real-world CQR data set demonstrate the superiority of the proposed PA-CQR framework and the Dynamic DPO.

1 Introduction

Conversational AI agents, such as Alexa, Siri, and Google Assistant, play a crucial role in the daily lives of individuals. To comprehend multi-turn spoken dialogues effectively, it is imperative to address the challenges of referring expressions resolution and entity tracking across the conversation, known as the "contextual carryover" problem (Naik et al., 2018; Anantha et al., 2020). Specifically, in a multi-turn conversation, users may omit or reference entities discussed earlier, causing ambiguity for the AI agent. Contextual query rewriting (CQR) (Zhou et al., 2023; Liu et al., 2021; Zuo et al., 2022; Sun et al., 2022), which rewrites the incomplete/ambiguous user query based on contextual information, have been widely utilized to address the contextual carryover problem.

Recent research have proposed various advanced

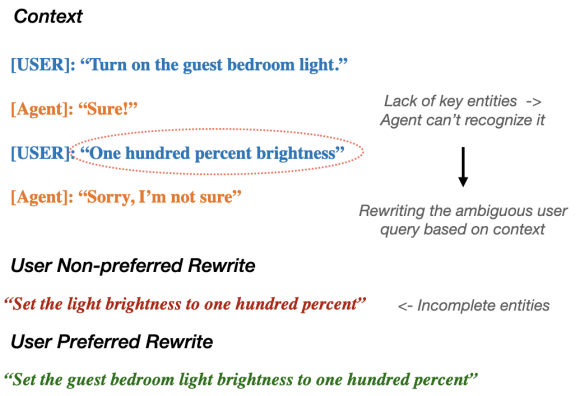

Figure 1: Contextual Query Rewrite (CQR) example, with both user-preferred and non-preferred rewrites.

CQR approaches (Naik et al., 2018; Chen et al., 2019; Yu et al., 2020). However, these methods typically only involve the supervised fine-tuning (SFT) stage, thereby missing some opportunities to further enhance the model from user-preference feedback. Figure 1 illustrates a CQR example. Recently, LHF (learning from human feedback) (Ouyang et al., 2022; Ziegler et al., 2019; Rafailov et al., 2023) has shown promising performance in leading language models to generate human-preferred content, which has been demonstrated as a key factor in the success of LLMs (large-language models) (Ouyang et al., 2022; Bai et al., 2022). Inspired by RLHF frameworks in (Ouyang et al., 2022; Bai et al., 2022), this paper proposes a user **P**reference **A**ligned **C**ontextual **Q**uery **R**ewrite framework, named as PA-CQR. PA-CQR consists of three stages: 1) the SFT stage fine-tunes a pre-trained language model (PLM) on the CQR data (in which the context with imperfect user query is the input and the ground truth rewrite is the target output); 2) the SFT model from stage 1 is applied to conduct inference on provided contexts and the generated rewrites are then fed into a reward model to obtain the feedback that indicates users' preference; 3) the obtained user-preference feedback

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 432–439
December 6-10, 2023

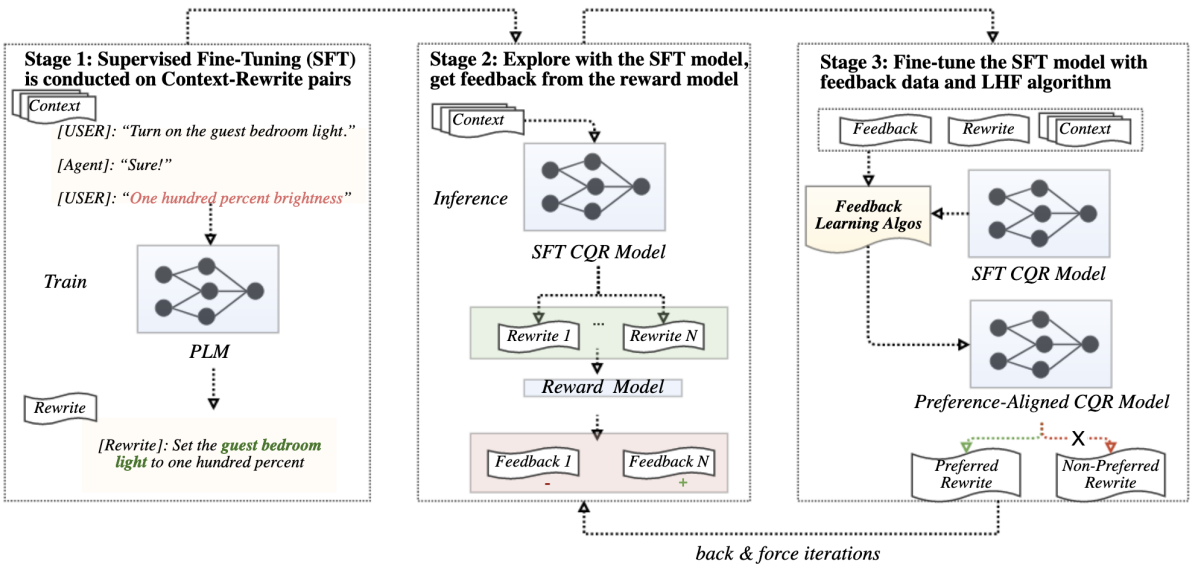

Figure 2: Overview of the proposed user preference-aligned CQR (PA-CQR) framework, which consists of the SFT, feedback collection, and feedback learning stages.

is utilized to fine-tune the SFT model through a feedback learning algorithm. Figure 2 illustrates the details of the proposed PA-CQR framework.

In the proposed PA-CQR framework, we also investigate the effectiveness of state-of-the-art feedback learning algorithms used in open-ended language generation tasks. To the best of our knowledge, this paper is the first to investigate effective feedback learning algorithms tailored for the CQR task. Specifically, we have studied the straightforward best-of-n feedback learning Expert Iteration (Anthony et al., 2017), the Preference Guided Feedback Learning inspired by (Lu et al., 2022) and Contrastive Feedback Learning inspired by Chain-of-Hindsight (Liu et al., 2023), the popular reinforcement learning algorithm PPO (Schulman et al., 2017), and the direct preference optimization DPO (Rafailov et al., 2023). To relieve the reward *distribution-shift* issue arises in the DPO algorithm, we also proposes a novel Dynamic DPO algorithm which gradually weaken the reference model's impact and switch from DPO objective to Maximum likelihood estimation objective. Extensive experiments on large-scale real-world user-agent CQR datasets demonstrate the effectiveness of our proposed PA-CQR and the Dynamic DPO.

2 Related Work

Contextual query rewriting (CQR) Contextual query rewriting (CQR) (Elgohary et al., 2018; Regan et al., 2019) is a crucial aspect in conversational AI as it involves reformulating the original query with additional or substitute terms that capture the true information need of the user based on the conversational context. Recently, language model based methods such as (Regan et al., 2019; Yu et al., 2020; Zuo et al., 2022) have been widely leveraged to conduct query rewriting by capturing necessary information from the context. Such techniques have also been successfully deployed to conversational AI systems (Rastogi et al., 2019; Zhou et al., 2023) to improve user experience. However, these works typically focus on the supervised-fine tuning stage while ignores the continually improvement procedure to generate better rewrites that can be aligned with user preference.

Aligning User Preference through Feedback Learning It has been a vital and challenging task to align content generated by the language model with human-preference through feedback learning. Given the fact that human preference feedback can be in arbitrary format and usually intrackable in model training, reinforcement learning (RL) algorithms such as PPO (Schulman et al., 2017) has widely adopted in training preference-aligned language models (Ouyang et al., 2022; Bai et al., 2022). However, reinforcement learning algorithms are often unstable, difficult to train, and expensive. Therefore, recently a variety of non-RL alternative feedback learning algorithms have been developed: Quark (Lu et al., 2022) first quantile generated content by reward and then re-train the

language model to generate corresponding content conditioned on the its reward; Chain-of-Hindsight (CoH) (Liu et al., 2023) encourages the model to generate both preferred and non-preferred content so that learn the key disparity among them, Direct Preference Optimization (DPO) (Rafailov et al., 2023) converts the reward maximization problem to a single stage of a classification training on the human preference data. In this paper, we have investigated both RL and non-RL feedback learning algorithms in the PA-CQR framework.

3 Preference Aligned CQR

In the context of a conversational AI system, we first introduce the concept of contextual query rewriting, which is more evident in the case of a multi-turn dialogue. For instance, in a multi-turn dialogue "[USER]: Turn on the guest bedroom light [Agent]: Sure [USER]: One hundred percent brightness", the user's entity slot "guest bedroom light" require carryover to facilitate the generation of a contextual query rewrite. Therefore, we can pose this scenario as a specific rewriting task, aiming to generate a contextually rewritten query, such as "[USER]: Set the guest bedroom light to one hundred percent brightness".

Despite recent advancements in LLMs, the importance of CQR is still pronounced, particularly for enhancing conversational AI agents in industrial scenarios: 1) Implementing LLMs for every user entails high costs and latency; 2) LLMs may still make carryover mistakes 3) it's more straightforward to achieve customized CQR to serve diverse users. Besides, the concept of CQR can be adapted to fit within future LLM scenarios. For example, when multiple LLM agents manage user/system interactions, CQR is essential for ensuring context continuity across the agents. Consequently, CQR retains a pivotal role in maintaining coherence and a seamless user experience, even in the LLM era.

In this section, we present the proposed PA-CQR framework, which consists of three stages: SFT for CQR, feedback collection, and feedback learning for CQR. We discuss each of these stages in the subsequent parts.

3.1 SFT for CQR

A pre-trained language model (PLM) is adopted for the SFT for CQR. For every training point, the previous dialogue turns (including both user requests and agent responses) and the current user request are flatten into a single sequence and fed input to the PLM, and the PLM is fine-tuned to generate the corresponding contextual rewrite.

Formally, the CQR task is cast as a text generation problem: given a flatten dialogue context sequence [1] $\mathbf{c} = \{c_1, ..., c_M\}$, where c_i for $i \in \{1, ..., M\}$ denotes a token in the sequence, and the corresponding rewrite $\mathbf{r} = \{r_1, ..., r_N\}$, the ultimate goal of the rewrite generation problem is to learn a probability distribution $P_\theta(\mathbf{r})$ over the variable-length text sequence $\mathbf{r}$, where θ is the parameter of the transformer model. Maximum likelihood estimation (MLE) objective is adopted to train the language model, which is defined as:

$$\mathcal{L}_\theta^{MLE}(\mathbf{c}, \mathbf{r}) = -\frac{1}{|\mathbf{r}|} \sum_{j=1}^{|\mathbf{r}|} \log P_\theta(r_j | \mathbf{r}_{<j}, \mathbf{c}) \ . \quad (1)$$

Typically, given finite training examples, i.e., T pairs of contextual query and rewrite $S = \{\mathbf{q}_t, \mathbf{c}_t\}_{t=1}^T$, the model is trained by minimizing the empirical finite sample objective loss function $\mathcal{L}_\theta^{MLE}(S) = \frac{1}{T} \sum_{t=1}^T \mathcal{L}_\theta^{MLE}(\mathbf{c}_t, \mathbf{r}_t)$.

3.2 Feedback Collection

The SFT CQR model is then applied to additional context to collect feedback. Following recent LHF work (Ouyang et al., 2022; Lu et al., 2022), contrastive feedback are gathered for every specific context. Specifically, a context is fed into the SFT CQR model and the N-best outputs are considered as N rewrite candidates. A reward model, which is capable of representing user preferences concerning the generated rewrites, is subsequently applied to every set of <*context, query, rewrite candidate*> to obtain user-preference feedback. Thus, this approach facilitates the collection of contrastive rewrite candidates (*user-preferred rewrite* v.s. *non-preferred rewrite*) for a specific context.

3.3 Feedback Learning Algorithms for CQR

Reinforcement learning (RL) algorithms (e.g., PPO) have been widely used to fine tune the SFT model with feedback (i.e., RLHF). However, such reinforcement learning algorithms on large-scale industrial data usually faces issues such as high complexity, high instability, high sensitivity to hyperparameters, and extremely expensive training costs.

[1] In the given example, we have the flatten dialogue context as "[USER] Turn on the guest bedroom light [AGENT] Sure [USER] One hundred percent brightness", where the last turn "USER] One hundred percent brightness" is the query that neesds rewrite.

Recently, alternative feedback learning methods (Lu et al., 2022; Rafailov et al., 2023; Liu et al., 2023) for language generation has been proposed to achieve a similar impact as RLHF with simpler implementation, better stability and lower cost. In this paper, we extensively explore four state-of-the-arts feedback learning algorithms for the proposed PA-CQR framework.

Learning from Positive Feedback. The most efficient approach for utilizing feedback data is direct fine-tuning the SFT model on positive feedback. This paper employs a common method known as *Expert-Iteration* (Anthony et al., 2017), specifically designed to learn from positive feedback. Initially, the model generates N rewrites given the context, then the model is subsequently fine-tuned on the <*context, best positive generated rewrite*> pair that holds the highest positive feedback reward score among the total N pairs.

Preference Guided Feedback Learning. Exclusively learning from positive feedback limits the model's awareness of undesirable content, potentially restricting its ability to utilize negative feedback in avoiding non-preferred content. A recent reward conditioning algorithm Quark (Lu et al., 2022) enforces the model to unlearn the misaligned generation by fine-tuning the SFT model conditioned on reward quantile. Inspired by Quark, we apply the similar *preference guided feedback learning* method that leverages both preferred and non-preferred feedback rewrites to fine-tune the model. Specifically, we first collect pairs of $(\mathbf{c}, \hat{\mathbf{r}})$, where $\mathbf{c}$ is the context, and $\hat{\mathbf{r}}$ is the generated rewrite of the SFT CQR model, assigned with a user preferred or non preferred feedback using the reward model (denoted as $+$ and $-$). Next, an indicator prompt is added to the context $\mathbf{c}$ based on the feedback of $\hat{\mathbf{r}}$ to create new fine-tuning data for the SFT CQR model. The learning instance is of the format $([\mathbf{p}, \mathbf{c}], \hat{\mathbf{r}})$, where $\mathbf{p}$ is "generate good rewrite:" when $\hat{\mathbf{r}}$ is $+$ and "generate bad rewrite" when $\hat{\mathbf{r}}$ is $-$. Formally, the Preference Guided Feedback Learning (PGFL) objective is

$$\mathcal{L}_\theta^{PGFL}(\mathbf{c}, \hat{\mathbf{r}}) = -\frac{1}{|\hat{\mathbf{r}}|}\sum_{j=1}^{|\hat{\mathbf{r}}|} \log P_\theta(\hat{r}_j|\hat{\mathbf{r}}_{<j}, [\mathbf{p}, \mathbf{c}]).$$

The model is trained by minimizing the empirical finite sample loss function $\mathcal{L}_\theta^{PGFL}(S) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_\theta^{PGFL}(\mathbf{c}_t, \hat{\mathbf{r}}_t)$.

Contrastive Generation Feedback Learning. In PGFL, the preference information is introduced in the input end. Alternatively, inspired by the work of Chain of Hindsight (CoH) (Liu et al., 2023) , the preference can also be introduced in the output end via a contrastive generation, which learns to generate both preferred and non-preferred rewrite simultaneously. Specifically, the model can be fine-tuned by taking the specific context as input and generating both the user-preferred and non-preferred rewrite pair $\hat{\mathbf{r}} = (\hat{\mathbf{r}}^+, \hat{\mathbf{r}}^-)$. This motivation is to allow the model to recognize the key disparities between positive and negative patterns through the generation of comparative forms, therefore to enhance the model's capacity of identifying and differentiating desirable and undesirable patterns. Formally, the loss of this Contrastive Generation Feedback Learning (CGFL) algorithm is:

$$\mathcal{L}_\theta^{CGFL}(\mathbf{c}, \hat{\mathbf{r}}) = -\frac{1}{|\hat{\mathbf{r}}|}\sum_{j=1}^{|\hat{\mathbf{r}}|} \log P_\theta(\hat{r}_j|\hat{\mathbf{r}}_{<j}, \mathbf{c}).$$

Similarly, the model is trained by minimizing the empirical finite sample loss function $\mathcal{L}_\theta^{CGFL}(S) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_\theta^{CGFL}(\mathbf{c}_t, \mathbf{r}_t)$.

DPO and Dynamic DPO. Recently (Rafailov et al., 2023) proposed a Direct Preference Optimization (DPO) algorithm that implicitly optimizes the same objective as existing RLHF. DPO directly optimizes the model by a straightforward contrastive loss to boosting the reward of preferred generation and penalizing that of the non-preferred generation. The DPO loss is

Algorithm 1 Dynamic DPO

Input: Initial policy model parameters θ_r, feedback dataset $\hat{S} = \{(\mathbf{c}_i,\ \hat{\mathbf{r}}_i^+, \hat{\mathbf{r}}_i^-)\}_{i=1}^T$

Set: Total iteration N_t, DPO iteration N_d, batch size b

for step n in $1, 2, .., N_t$ **do**
  Sample batch $B = \{(\mathbf{c}_i,\ \hat{\mathbf{r}}_i^+, \hat{\mathbf{r}}_i^-)\}_{i=1}^b$ from $\hat{S}$
  $\mathcal{L}_\theta^{MLE} = \frac{1}{b}\sum_{i=1}^b \mathcal{L}_\theta^{MLE}(\mathbf{c}_i, \hat{\mathbf{r}}_i^+)$
  if $n <= N_d$ **then**
    $\mathcal{L}_\theta^{DDPO} = \frac{1}{b}\sum_{i=1}^b \mathcal{L}_\theta^{DDPO}(\mathbf{c}_i,\ \hat{\mathbf{r}}_i^+, \hat{\mathbf{r}}_i^-)$
    $\mathcal{L}_{total} = \mathcal{L}_\theta^{MLE} + \mathcal{L}_\theta^{DDPO}$
  else
    $\mathcal{L}_{total} = \mathcal{L}_\theta^{MLE}$
  end if
  Update θ using gradient descent on loss $\mathcal{L}_{total}$
end for

$$\mathcal{L}_\theta^{DPO}(\mathbf{c}, \hat{\mathbf{r}}^+, \hat{\mathbf{r}}^-) = -\log\sigma\left(\beta\log\frac{P_\theta(\hat{\mathbf{r}}^+|\mathbf{c})}{P_{\theta_r}(\hat{\mathbf{r}}^+|\mathbf{c})} - \beta\log\frac{P_\theta(\hat{\mathbf{r}}^-|\mathbf{c})}{P_{\theta_r}(\hat{\mathbf{r}}^-|\mathbf{c})}\right)$$

where σ represents logistic function and β is a weight hyper-parameter, θ_r is the reference model (SFT CQR model in our case). Intuitively, the DPO loss function implicitly increases the *reward* of the positive rewrite $\hat{\mathbf{r}}^+$ and decreases the *reward* of negative rewrite $\hat{\mathbf{r}}^-$, where the *reward* is approximated by the likelihood re-weighted by the reference model θ_r, i.e., $\beta\log\frac{P_\theta(\hat{\mathbf{r}}^+|\mathbf{c})}{P_{\theta_r}(\hat{\mathbf{r}}^+|\mathbf{c})}$.

However, as training progresses, the policy model gradually diverges from the initial reference model, and aligning more closely with a distribution that is consistent with the feedback data. Consequently, the reward approximated from the reference model may substantially deviate from the distribution of the current policy model, causing an impact on the training of the policy model. This is identified as the *reward distribution-shift* issue. To mitigate this problem, we propose a Dynamic DPO algorithm which adds a decaying factor in the reference model and interpolates between normal MLE training and DPO training. The intuition is to gradually weaken the weight of the reference model in DPO and smoothly transit from the DPO objective to MLE eventually. The proposed dynamic DPO (DDPO) loss is define as

$$\mathcal{L}_\theta^{DDPO}(\mathbf{c}, \hat{\mathbf{r}}^+, \hat{\mathbf{r}}^-) = -\log\sigma\left(\beta\log\frac{P_\theta(\hat{\mathbf{r}}^+|\mathbf{c})}{P_{\theta_r}(\hat{\mathbf{r}}^+|\mathbf{c})^{\epsilon_n}} - \beta\log\frac{P_\theta(\hat{\mathbf{r}}^-|\mathbf{c})}{P_{\theta_r}(\hat{\mathbf{r}}^-|\mathbf{c})^{\epsilon_n}}\right)$$

where $\epsilon_n = \min(1, \frac{C}{n})$ is the decaying factor which decays as iteration steps n increases when step n is larger than a threshold C. Given a batch of data $B = \{(\mathbf{c}_i, \hat{\mathbf{r}}_i^+, \hat{\mathbf{r}}_i^-)\}_{i=1}^b$, the loss on the batch is $\mathcal{L}_\theta^{DDPO}(B) = \sum_{i=1}^b \mathcal{L}_\theta^{DDPO}(\mathbf{c}_i, \hat{\mathbf{r}}_i^+, \hat{\mathbf{r}}_i^-))$. After a certain iteration steps (i.e., N_d in Algorithm 1 line 4), we switch the training objective of combined DDPO loss and MLE loss ((line 6 in Algorithm 1)) to only MLE loss (line 8 in Algorithm 1). The detailed algorithm is described in Algorithm 1.

4 Experiments

We conduct experiment on a large-scale real-word industry CQR to validate the PA-CQR framework and evaluate the feedback learning methods discussed in section 3. Note that all data used in this paper has been de-identified therefore no user information is remained.

Name	# trigger	# non-trigger
Real-world Train	1M	1M
Real-world Test	4k	16k

Table 1: CQR training and test set statistics.

4.1 Experiment Setup

Dataset Training data for the CQR task contains 2M de-identified real world user-agent contextual conversations. Among the 2M data, 1M is the should-trigger CQR data, i.e., the last user query in each context has a corresponding ground truth rewrite. Thus the model needs to take the context (includes the last user query) as input and predict the corresponding ground truth rewrite; The remaining 1M data are selected from non-triggered CQR traffic, in which the last user query is either accurate enough or not be able to rewritten. The model then needs to take the context as input and predicts "NULL" as the rewrite output. The CQR model is trained on both of the should-trigger and not-triggered CQR data, so that to learn to determine when should it provide the query rewrite and generate correct rewrite simultaneously.

For test, a 20k human-annotated dataset on sampled real-world traffic, include both should-trigger and non-trigger data, is used. Table 1 demonstrates the statistics of train and test sets.

Evaluation Metrics Three evaluation metrics are utilized: 1) **Rewrite Accuracy**: Given the fact that only high confidence rewrites will be triggered in practical, the utterance-level precision at a set 20% trigger rate is used as the rewrite accuracy; 2) **Entity Omission Rate**: The utterance-level precision can be limited as it requires a strict match. Thus, the percentage of cases where predicted rewrite misses a key entity in the ground truth rewrite label is also examined. The key entities are identified as the non stop-words entities in the ground truth rewrite. 3) **Trigger F1**: The F1 score of the trigger prediction is calculated, which is used to measure the model's performance in determining when should trigger the rewrite given the context.

Model Set-up The FLAN-T5-Large (Chung et al., 2022) serves as the base PLM for all experiments. All experiments are executed on eight A100 GPUs. The epoch number is set as 10 for all experiments. The learning rates for all methods are set as $3e-5$. The batch size is set as 32 for DPO/DDPO, 8 for PPO, and 128 for all other methods.

Set-up		Annotated Test		
Method	Data-size	Rew Acc @ 20% ↑	Entity Omission @ 20% ↓	Trigger F1 ↑
SFT	2M	0.0%	0.0 %	0.0%
PPO	(2M) + 400k	+ 0.9%	- 1.06%	- 0.24%
Exp-Iteration	(2M) + 400k	+ 2.56%	- 8.48%	**+ 1.43**%
Preference Guided	(2M) + 400k	- 22.0%	+ 55.1%	- 4.76%
Contrastive Generation	(2M) + 400k	- 17.3%	- 37.1%	-2.97%
DPO	(2M) + 400k	+ 4.51%	- 14.1%	+ 1.18%
Dynamic DPO	(2M) + 400k	**+ 6.62**%	**-18.4** %	+ 0.36%

Table 2: Overall result table of different methods. Relative improvements compared to the SFT model are reported for each feedback learning algorithm. The SFT model represents fine-tuning Flan-t5-large on the 2M data. The feedback is collected by applying the SFT model to the 2M data again. For a fair comparison, 400k feedback data are collected for every setting.

4.2 Experiment Results

To evaluate the performance of different feedback types and LHF algorithms, following the pipeline illustrated in Figure 2, the raw FLAN-T5-large model is first fine-tuned using the 2M training data to obtain the SFT model. The SFT model is then utilized to perform inference on the same 2M data points, and the resulting rewrites are processed by the reward model, which is trained using data from human annotation and heuristic rules. We selected 400k feedback data from the reward results, in which each context has one user-preferred rewrite and one non-preferred rewrite. (note: the reward model is only deployed to cases where rewrite triggers). Then, the SFT model is fine-tuned with additional feedback data using different methods described in section 3.3. PPO (Schulman et al., 2017) is also applied as a RL baseline.

The primary results of the experiment are shown in Table 2. An initial observation reveals that feedback learning techniques such as *Expert-Iteration, DPO, Dynamic DPO* outperform both the SFT model and the PPO method in terms of CQR-related metrics, thereby verifies the efficacy of the PA-CQR framework we propose. Moreover, the *Expert-Iteration* feedback learning exhibits a significant enhancement in rewrite accuracy and entity omission rate. *Expert-Iteration* can be perceived as an approach of seeking the optimal rewrite from an expansive array of self-generated candidates, thus facilitating the SFT model's feedback learning towards improved performance. This result further demonstrates the necessity of feedback-learning and the great potential of improving the model by examining and learning from its own generated content. However, it is notable that the *Preference Guided* and *Contrastive Generation* methods show worse performance on the CQR task. These two methods integrate feedback learning information via text-format by either modifying the input text or target text. However, different from general generation tasks used in (Lu et al., 2022; Liu et al., 2023), the key disparity between preferred and non-preferred rewrites in CQR tasks could be subtle (e.g., *"Set light to green"* versus *"Set bedroom light to green"*). Hence, training the SFT on text-level feedback could potentially overlook key factors, leading to model confusion. Lastly, both the *DPO* and the proposed *Dynamic PPO* show promising results, and the *Dynamic DPO* showing superior results in terms of rewrite accuracy and entity omission metrics. This verifies the effectiveness of our proposed *Dynamic DPO* algorithm.

5 Conclusion

This paper introduces the PA-CQR framework, which is inspired by the recent achievements of human feedback learning, to continually improve the industry CQR model to generate better rewrites that are aligned with user preference. Besides, to mitigate the limitations of the DPO algorithm in large-scale CQR training, the paper also proposes a novel Dynamic DPO algorithm which gradually weaken the impact of the reference model in training. Extensive experiments conducted on real-world user-agent CQR dataset experiments have demonstrated the effectiveness of the proposed PA-CQR framework, certain feedback learning algorithms such as *Expert-Iteration, DPO, Dynamic DPO*. The research also reveals that feedback learning methods such as *Preference Guided, Contrastive Generatio* exhibit limited performance when applied to the CQR task , highlighting the potential for further research and development in this area.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. *arXiv preprint arXiv:2010.04898*.

Thomas Anthony, Zheng Tian, and David Barber. 2017. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30.

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

Tongfei Chen, Chetan Naik, Hua He, Pushpendre Rastogi, and Lambert Mathias. 2019. Improving long distance slot carryover in spoken dialogue systems. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 96–105.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Ahmed Elgohary, Chen Zhao, and Jordan Boyd-Graber. 2018. Dataset and baselines for sequential open-domain question answering. In *Empirical methods in natural language processing*.

Hang Liu, Meng Chen, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Conversational query rewriting with self-supervised learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7628–7632. IEEE.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Chain of hindsight aligns language models with feedback. *arXiv preprint arXiv:2302.02676*, 3.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609.

Chetan Naik, Arpit Gupta, Hancheng Ge, Lambert Mathias, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. *arXiv preprint arXiv:1806.01773*.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Pushpendre Rastogi, Arpit Gupta, and Lambert Mathias. 2019. Contextual query rewriting (cqr): natural language as interface for dialog state tracking. In *NAACL 2019*.

Michael Regan, Pushpendre Rastogi, Arpit Gupta, and Lambert Mathias. 2019. A dataset for resolving referring expressions in spoken dialogue via contextual query rewrites (cqr). *arXiv preprint arXiv:1903.11783*.

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

Zhongkai Sun, Sixing Lu, Chengyuan Ma, Xiaohu Liu, and Chenlei Guo. 2022. Query expansion and entity weighting for query reformulation retrieval in voice assistant systems. *arXiv preprint arXiv:2202.13869*.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1933–1936.

Yingxue Zhou, Jie Hao, Mukund Rungta, Yang Liu, Eunah Cho, Xing Fan, Yanbin Lu, Vishal Vasudevan, Kellen Gillespie, Zeynab Raeesy, et al. 2023. Unified contextual query rewriting.

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

Simiao Zuo, Qingyu Yin, Haoming Jiang, Shaohui Xi, Bing Yin, Chao Zhang, and Tuo Zhao. 2022. Context-aware query rewriting for improving users' search experience on e-commerce websites. *arXiv preprint arXiv:2209.07584*.

A Appendix

A.1 Additional Analysis on un-generating Negative Patterns

To further verify if the feedback learning algorithm is effective in fine-tune the SFT CQR model in un-learning non-preferred patterns, additional <context, non-preferred rewrite> pairs are collected. Specifically, the non-preferred rewrites are obtained from the inference result of the SFT

model that are labeled as non-preferred by the reward model. Next, each model's likelihood (represented as the $loss$ of model(context, non-preferred rewrite)) of generating the given non-preferred rewrite for the given context can be calculated to represent the model's capability in un-generating non-preferred patterns.

Table 3 demonstrates the result. It is observed that the effective feedback learning algorithms such as *Expert-Iteration, PPO, DPO, Dynamic DPO* all have lower likelihood (higher loss) in generating non-preferred patterns. Besides, methods like *PPO, DPO, Dynamic DPO* have a higher performance than *Expert Iteration* because they are directly optimized using both preferred and non-preferred patterns.

Method	Non-preferred Loss
SFT	0.65
PPO	0.87
Expert-Iteration	0.71
Preference Guided	0.72
Contrastive Generation	0.66
DPO	**1.13**
Dynamic DPO	1.09

Table 3: Comparisons of different methods' capabilities in un-generating non-preferred patterns, represented using each model's loss value of <context, non-preferred rewrite>.

A.2 Training Speed

Table 4 illustrates the training speed for every feedback learning algorithms. The training speed is represented as the average number of tokens processed every second, on 8 A100 GPUs. The numbers show that the Expert Iteration is the fastest option while the PPO requires a large training cost.

Method	Training Seed (# tokens / s)
SFT	6340
PPO	310
Expert-Iteration	6200
Preference Guided	6170
Contrastive Generation	6250
DPO	2140
Dynamic DPO	1190

Table 4: Training speed for every feedback learning algorithm.

B Ethical Discussion

This work aims at enhancing the performance of Contextual Query Rewriting (CQR) for conversational AI agents through feedback learning. However, implementing such feedback and corresponding feedback learning algorithms may involve ethical considerations in privacy and data protection.

For example, the training of the reward model and CQR model requires real-world user-agent dialogues. Therefore, it's critical to guarantee that the acquisition and processing of this data are conducted in a manner that user privacy information is well protected. In this work, all data for training and testing are from sources where user identification and privacy information have been removed. This procedure ensures that users' private details have been omitted and are not input into the models. Moreover, in the realistic production pipeline, several additional safety examinations are employed to assure that both the training data collection and output rewriting comply with appropriate content standards.