User Simulator Assisted Open-ended Conversational Recommendation System

Qiusi Zhan[1], **Xiaojie Guo**[2], **Heng Ji**[1], **Lingfei Wu**[3]
[1]University of Illinois Urbana-Champaign
[2]IBM Thomas J. Watson Research Center, [3]Pinterest
{qiusiz2, hengji}@illinois.edu
xguo7@gmu.edu, teddy.lfwu@gmail.com

Abstract

Conversational recommendation systems (CRS) have gained popularity in e-commerce as they can recommend items during user interactions. However, current open-ended CRS have limited recommendation performance due to their short-sighted training process, which only predicts one utterance at a time without considering its future impact. To address this, we propose a User Simulator (US) that communicates with the CRS using natural language based on given user preferences, enabling long-term reinforcement learning. We also introduce a framework that uses reinforcement learning (RL) with two novel rewards, i.e., recommendation and conversation rewards, to train the CRS. This approach considers the long-term goals and improves both the conversation and recommendation performance of the CRS. Our experiments show that our proposed framework improves the recall of recommendations by almost 100%. Moreover, human evaluation demonstrates the superiority of our framework in enhancing the informativeness of generated utterances. [1]

1 Introduction

Conversational Recommendation Systems (CRS) (Li et al., 2018; Chen et al., 2019; Zhou et al., 2020; Liang et al., 2021; Lei et al., 2020b; Deng et al., 2021; Yang et al., 2022) are of growing interest. Unlike traditional recommendation systems, CRS extract user preferences directly and recommend items during their interaction with users. Traditional CRS (Deng et al., 2021; Lei et al., 2020b,a) recommend an item or ask about the user preference of a specific attribute at a turn and use predefined question templates with item/attribute slots in practical applications, which are denoted as attribute-centric CRS. In addition, they often use reinforcement learning to learn a policy of recommending items and asking about attributes. Although such attribute-centric CRS are popular in industry due to its easy implementation, the user experience is unsatisfactory due to its lack of flexibility and interactivity. In addition, limited user information is collected by the CRS due to the constrained interaction format. To this end, open-ended CRS(Li et al., 2018; Chen et al., 2019; Zhou et al., 2020; Liang et al., 2021; Yang et al., 2022) are proposed to provide more flexible interactions with users. Such CRS can interact with the user like a real human-being, which focus on understanding user preferences according to their utterances and generating fluent responses to recommend items.

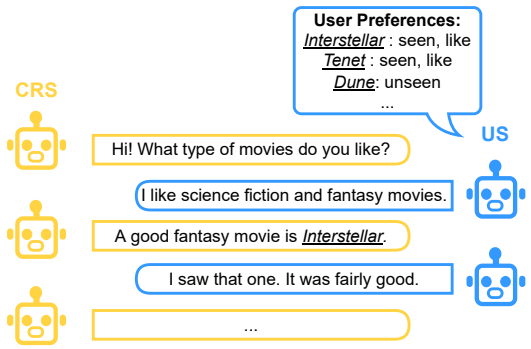


Figure 1: Overview of our proposed framework. The User Simulator (US) can interact with the Conversational Recommendation System (CRS) based on certain user preferences.

Although open-ended CRS can engage in natural and fluent conversations with users, their recommendation quality are often suboptimal. This is partly because these systems are typically trained using maximum likelihood estimation (MLE) to predict one utterance at a time, which hinders their ability to learn a long-term recommendation policy (Li et al., 2016b). Moreover, such MLE training fails to directly address the primary goal of CRS, which is to gradually explore user preferences and provide accurate, informative recommendations.

[1]Our code is released at https://github.com/ZQS1943/CRS_US.

Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023), pages 89–101

For instance, systems trained with MLE may generate generic and unhelpful responses, such as "*You're welcome. Bye.*"

Traditional attribute-centric CRS can learn effective recommendation policies by using reinforcement learning to enable a global view of the conversation. However, adapting this strategy to open-ended CRS is challenging due to the lack of a suitable User Simulator (US) for them. Developing a US for open-ended CRS is much harder than for attribute-centric CRS because it needs to generate natural-sounding utterances that are consistent with specific user preferences, rather than simply providing signal-level feedback as in the US for attribute-centric CRS. The US can serve not only as an environment for reinforcement learning but also provide more diverse and realistic human-like conversation scenarios and patterns than fixed training datasets. A suitable US for open-ended CRS would be a significant step toward improving their recommendation quality and making them more effective in real-world applications.

This paper proposes a framework that includes a CRS and a US to facilitate RL of the CRS. Specifically, we first develop a US for open-ended CRS, comprising three preference-aware modules that generate user utterances based on any given user preferences. Building on recent work in applying RL for dialogue generation (Tseng et al., 2021; Papangelis et al., 2019; Das et al., 2017; Li et al., 2016b), we propose optimizing the long-term performance of pre-trained CRS using RL during interaction with the US. We also introduce two rewards: the recommendation reward and the conversation reward, to better reflect the true objective of CRS. To the best of our knowledge, this is the first framework for training open-ended CRS in reinforcement learning strategies.

The contributions of this work are summarized as follows:

- We present the first US that can interact with the CRS using natural language based on specific user preferences. With three preference-aware modules, the proposed US not only gives the correct feedback to the CRS recommended items, but also expresses its preference actively to let the CRS know more about the user in a short dialog.
- We present the first framework for fine-tuning a pre-trained open-ended CRS with RL and introduce two rewards to improve both conversation and recommendation performance.
- Comprehensive experiments are conducted, which demonstrate that the proposed framework is powerful in improving both the accuracy of the recommendation and the informativeness of the generated utterances.

2 Methods

2.1 Overall Architecture

Formally, in the CRS scenario, we use u to denote a user from the user set $\mathcal{U}$ and i to denote an item from the item set $\mathcal{I}$. A dialog context can be denoted as a sequence of alternating utterances between the CRS and the user: $\{x_1^{crs}, x_1^{us}, x_2^{crs}, x_2^{us}, \cdots, x_t^{crs}, x_t^{us}\}$. In the t-th turn, the CRS generates an utterance x_t^{crs} that recommends the item $i_t \in \mathcal{I}$. Note that i_t can be None if x_t^{crs} is a chit-chat response or is a query to clarify the user preference and does not need to recommend. The user then provides a response x_t^{us}.

Our goal is to train the CRS with reinforcement learning to improve its long-term performance. Since online human interactive learning costs too much effort in training, a US is utilized to assist the RL process of the CRS, by simulating natural and personalized dialogue contexts. To train the overall framework, we first train a US that can simulate user utterances based on specific user preferences in each dialog, using supervised learning. We then fine-tune a pre-trained CRS by encouraging two novel rewards during the interaction with the US through reinforcement learning.

2.2 User Simulator

In this section, we present our US, which aims to interact with CRS using natural language based on any given user preferences. However, developing such a US comes with two main challenges: (1) the US must be able to express its preferences both actively and passively. It should provide accurate feedback on recommended items and actively express its preferences to quickly provide the CRS with more information in a short dialogue. (2) preserving the long-term preferences of the user creates a large search space for item selection, which can burden the US. Additionally, users are only interested in a small set of items in each dialogue, requiring the US to model dynamic user preferences

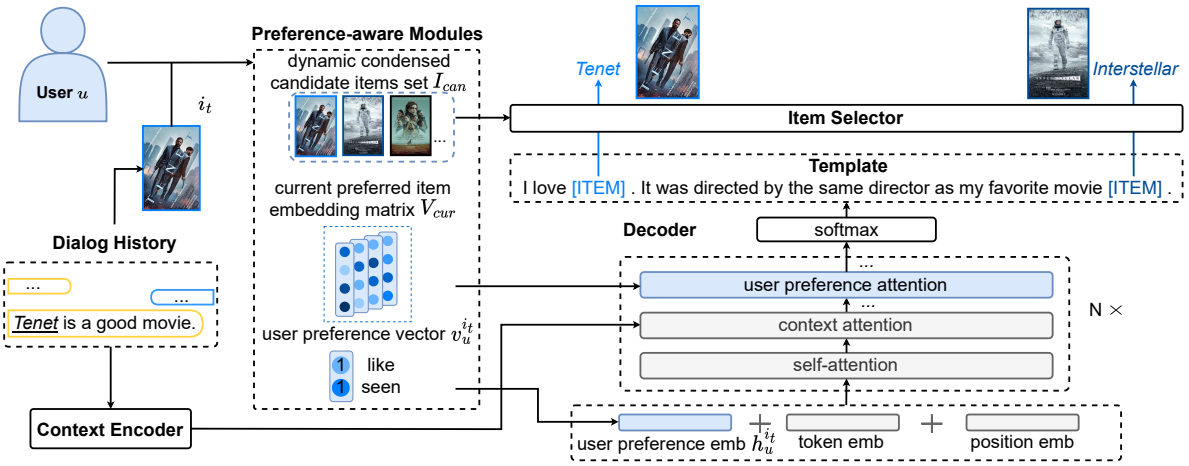


Figure 2: Our proposed User Simulator. Given the dialog history, a transformer-based Encoder-Decoder module enhanced with *user preference embedding* and *user preference attention* is used to generate a personalized response template with item slots. An Item Selector is used to select the appropriate items from the *dynamic condensed candidate items set* I_{can} based on the context and the user preference.

in the current dialogue. To address the first challenge, we propose two components: *User Preference Embedding* to capture the user's personalized characteristics for a recommended item, enabling the US to generate appropriate feedback, and *User Preference Attention* to prompt the US to express its preferred items. To tackle the second challenge, we employ the use of *Dynamic Condensed Candidate Item Set*, which captures the user's short-term preferences, thereby reducing the search space for item selection.

Figure 2 shows an overview of our proposed US, which is based on a dialog generation model NRTD (Liang et al., 2021). Given the dialogue context, we first utilize a knowledge-enhanced encoder-decoder-based template generator, depicted as the "context encoder" and "decoder" in Figure 2, to generate an utterance template with item slots. In the decoder module of the template generator, we incorporate *user preference embedding* to enhance token embedding with information about the last recommended item and add a *user preference attention* layer to incorporate user preferred items into the generated templates. Next, we use a template-aware item selector to select the appropriate items from a preference-based *dynamic condensed candidate items set*. We introduce these three preference-aware modules (*User Preference Embedding*, *User Preference Attention*, and *Dynamic Condensed Candidate Items Set*) in the following sections. We refer the reader to (Zhou et al., 2020) and (Liang et al., 2021) for more details of the whole model.

User Preference Embedding

When the CRS recommends an item, US is expected to provide the correct feedback for it. To achieve this, for each user u, we represent their *user preference vector* for item i as $v_u^i \in \mathbb{R}^{n_f}$, where n_f is the number of features to consider, such as a score indicating the user's liking for the item or a binary value indicating whether the user has purchased the item or not. We then map v_u^i to a continuous space using the following equation:

$$h_u^i = W v_u^i \tag{1}$$

where $h_u^i \in \mathbb{R}^d$ represents the *user preference embedding*, and $W \in \mathbb{R}^{d \times n_f}$ is a learnable matrix.

When generating user utterances, we incorporate the user u's preference embedding of the last recommended item i_t, *i.e.*,$h_u^{i_t}$, into each word embedding to assist the US in generating accurate feedback for the recommended item i_t.

User Preference Attention

In addition to providing accurate feedback for recommended items, a good US should also actively express its preferences to provide the CRS with more information about the user. A user may have a large set of preferred items in the long-term, but in a short-term, during a current dialogue, they may be looking for specific types of items such as comedies, scary movies, etc. To this end, we define the user's *current preferred item set* $\mathcal{I}_{cur}$ as the set of user's short-term preferred items mentioned in a single dialogue in the dataset. We then use $V_{cur} \in \mathbb{R}^{d \times |\mathcal{I}_{cur}|}$ to denote *current preferred item*

embedding matrix, where each column is a learnable representation of a preferred item enhanced by an external knowledge graph.

Then we add a multi-head attention layer, *i.e.*, $\mathrm{MHA}(Q, K, V)$, to each layer of the decoder to incorporate this user preference information:

$$R' = \mathrm{MHA}(R, V_{cur}, V_{cur}) \tag{2}$$

where $R \in \mathbb{R}^{d \times l}$ and $R' \in \mathbb{R}^{d \times l}$ are the embedding matrix before and after user preference attention in each layer of the decoder.

Dynamic Condensed Candidate Items Set

Searching through a large space of candidate items can impose a significant burden on the US in generating accurate and controllable utterances, especially when dealing with a large number of candidate items as seen in real-world scenarios. Furthermore, users' short-term preferences can change dynamically throughout a dialogue, which can affect the distribution of preferred items in the search space. To address this, we propose the use of a *dynamic condensed candidate item set* $\mathcal{I}_{can}$ which limits the number and quality of items, and the item selector can only select items from $\mathcal{I}_{can}$ for recommendation.

There are two key considerations in constructing the dynamic condensed candidate item set. First, as previously discussed, the US is expected to provide accurate feedback on the last recommended item i_t, therefore the last recommended item must be included in the set. Second, to accommodate the dynamic short-term preference of the users, the current preferred item set is also included, as $\mathcal{I}_{can} = \mathcal{I}_{cur} \cup \{i_t\}$.

Optimization of US

The entire User Simulator (US) is trained end-to-end, using human-written dialogues as supervision. For template generation, we use a standard cross-entropy loss L_{gen}. For item selection, we calculate the loss as the negative log-likelihood of the ground-truth item for an item slot, denoted as L_{sle}. We then combine the two losses with a weighting hyperparameter as follows:

$$L = \lambda L_{gen} + L_{sle} \tag{3}$$

We refer the reader to (Liang et al., 2021) for more details.

2.3 Reinforcement Learning of CRS

With the proposed US, we can fine-tune any pre-trained CRS using RL, based on its interactions with the US. Our US is able to create diverse training scenarios for the CRS by altering user preferences, which it uses as a basis for generating user utterances. In each dialog session, the CRS is fine-tuned based on a fixed user's *current preferred item set* $\mathcal{I}_{cur}$ from a dialog in the training set, with the aim of recommending items in $\mathcal{I}_{cur}$. This approach enables the CRS to model the long-term effects of a generated utterance and more closely imitate the true goal of a CRS, which is to recommend items that users will like, by utilizing designed rewards (Li et al., 2016b).

RL Components

An **action** a refers to a dialogue utterance generated by the CRS; the **state** is represented by the previous dialogue history c; the **policy** of the CRS model is represented by $p(a|c)$, defined by its parameters; r represents the **reward** obtained for each action.

Reward Design

Compared to RL in the task-oriented dialog (Tseng et al., 2021; Papangelis et al., 2019), the main challenge of RL in CRS is that there are no predefined dialog acts to use, and the model must take into account both the recommendation and the conversation performance, rather than simply selecting the best dialog act. To address this, we design two novel rewards for reinforcement learning in CRS training.

For the *recommendation reward*, inspired by the studies of attribute-centric CRS (Lei et al., 2020a,b; Deng et al., 2021), which use RL to enhance the efficiency of recommendations, our environment contains two types of rewards: (1) r_{rec_suc}, a positive reward when the user likes the recommended item, *i.e.*, the recommended item is in the user's *current preferred item set* I_{cur}, and (2) r_{rec_fail}, a negative reward when the user dislikes the recommended item.

For the *conversation reward*, we first provide a slightly positive reward r_{con_rec} when the generated utterance recommends an item, to encourage the CRS to make recommendations. Additionally, when recommending an item, the CRS should also explain why it chose the item, making it more persuasive. For instance, in the movie domain, the CRS may recommend a movie that shares the same actor as the user's favorite movie

mentioned earlier. To encourage this, we construct a list of non-informative words, based on word frequency, excluding informative words about attributes of movies, such as movie genres and actor names. If the generated utterance contains a word that is not on this list of non-informative words, we consider it to be an informative utterance and provide a positive reward r_{con_info}. During our experiments, we also found that the CRS tends to use repeated templates to recommend different items in a single dialogue, which can make the conversation monotonous. To address this, we give slightly negative rewards r_{con_rep} to repeated templates.

Finally, the total reward is calculated as follows:

$$\begin{aligned} r =& \alpha(r_{rec\_suc} + r_{rec\_fail}) + \\ & \beta(r_{con\_rec} + r_{con\_info} + r_{con\_rep}) \end{aligned} \tag{4}$$

where α, β are weight hyperparameters.

Optimization of CRS

The model parameters are initialized using the pre-trained CRS model. We then use *Policy Gradient Theorem* (Sutton et al., 1999) to find parameters that maximize the expected reward, which can be written as

$$J(\theta) = \mathbb{E}[\sum_{i=1}^{T} R(a_i, c_i)] \tag{5}$$

where $R(a_i, c_i)$ denotes the reward resulting from action a_i given context c_i. We use the likelihood ratio trick (Williams, 1992; Li et al., 2016b) for gradient updates:

$$\nabla J(\theta) \approx \sum_{i} \nabla \log p(a_i|c_i) \sum_{i=1}^{i=T} R(a_i, c_i) \tag{6}$$

.

3 Experimental

3.1 Dataset

We conduct all the experiments on the *REcommendations through DIALog* (REDIAL) dataset(Li et al., 2018). It is collected on Amazon Mechanical Turk (AMT) platform where paired workers, recommender and seeker, make conversations about movie seeking and recommendation. It consists of 10006 dialogues with an average of 18.2 turns. 738 workers play the seeker roles at least in one dialog. There are 51699 movie mentions, of which 16278 are mentioned by the seeker and 35421 are recommended by the recommender. After the two workers complete the conversation, the system would ask the seeker to complete a table about whether he/she likes each mentioned movie or not and has seen it or not, which are the two features we use to model the user preferences. The seekers like most movies with more than 95% of all movie mentions are liked by the seekers. We first use the dialogues in the dataset to train the US in a supervision style. For the reinforcement learning of the CRS, at each round, we start the conversation based on the above-mentioned dataset, and continue the training of CRS during its interaction with the US, which is based on the user preference from the training data.

3.2 Evaluation Metrics

Following the previous open-ended work, we evaluate the CRS in terms of recommendation and conversation performance. However, existing works only evaluate the conversation quality locally, namely, one-round conversation, and the input dialogue history of the CRS is always the human-written utterances without any self-generated context. Thus, to evaluate the CRS in terms of its global performance in one dialog, we propose two novel global metrics in addition to the local evaluation. The details of the local metrics and the global metrics are provided as follows.

Local Metrics For recommendation evaluation, previous work often use *recall in response* (ReR), which shows whether the ground-truth item suggested by human is included in the final generated response. However, this deviates from the true goal of the CRS, which is to recommend user-liked items. Thus, we suggest expending the target item set to the user *current preferred item set* $\mathcal{I}_{cur}$, and using *recall of preferred items* (ReP) to measure whether the recommended item is included in $\mathcal{I}_{cur}$. For the evaluation of conversation, following previous work, we use *perplexity* (PPL) and *distinct n-gram* (Dist-n) (Li et al., 2016a) to measure the fluency and distinctiveness of generated utterances. We also use human evaluation to measure fluency and information quality.

Global Evaluation We propose two global metrics to evaluate the recommendation performance of the CRS during its interaction with the US. *Global recall* (GlobalRe) is calculated as the percentage of items recommended in the entire dialog that are in

the user *current preferred item set* $\mathcal{I}_{cur}$. We also use *success rate* (Succ) where success means that the CRS has recommended at least one item that is in $\mathcal{I}_{cur}$ within a certain number of maximum turns. During the evaluation, the US employs user preferences, i.e., the *current preferred item set* $\mathcal{I}cur$ from the test set. This means that each user in a dialogue is treated as a distinct entity, and their $Icur$ represents the set of items mentioned in the dialogue that are liked by that particular user.

3.3 Implementation Details

Our framework can theoretically be paired with any CRS models.[2] In this experiment, we implement our model based on the CRS model NTRD (Liang et al., 2021), which consists of a recommendation component and a conversation component. We freeze the parameters of one component and train another one at a time using the corresponding reward to make the training process more stable. Both components are optimized with Adam optimizer with a batch size of 16. The maximum number of turns is set to 5. We train the recommendation component with a learning rate of 1e-4 for 20 epochs and the conversation component with a learning rate of 1e-7 for 40 epochs. On average, it takes approximately one hour to train an epoch with a Tesla P100GPU with 16GB of DRAM. For more implementation details, including the training of the US and the exact number of each reward, please refer to the Appendix.

3.4 Baselines

- **REDIAL** (Li et al., 2018): original model proposed with the dataset.
- **KBRD** (Chen et al., 2019): based on transformer, utilizing an external knowledge graph to enhance the item representations.
- **KGSF** (Zhou et al., 2020): utilized two external knowledge graphs to further enhance the user preference modelling.
- **NTRD** (Liang et al., 2021): proposes the two-step framework with a template generator and an item selector to better incorporate the recommended items into the generated responses.
- **RID** (Wang et al., 2021): utilizes the pre-trained language model to improve the CRS.
- **MESE** (Yang et al., 2022): also utilizes the pre-trained language model but use items meta information instead of the KG as the external knowledge.

[2]We do not incorporate the proposed framework into CRSs (Yang et al., 2022; Wang et al., 2021) with pre-trained language models since it costs too much memory to perform reinforcement learning.

3.5 Experimental Results

Machine-based Evaluation Table 1 shows the machine-based evaluation results of the models. Compared to the NTRD base model, our framework consistently improves the performance of the model in all metrics. In particular, our framework improves all recommendation metrics by almost 100%. This indicates that the CRS learns a good policy of recommending through the interaction with the US with the designed rewards. Note that after fine-tuning with our framework, the NTRD even outperforms the RID, which leverages a pre-trained language model (PLM) in terms of the recommendation.

The ablation study shows that both the recommendation reward and the conversation reward contribute to the final results. The conversation reward also improves the recommendation performance, which may be because a more informative response helps the model choose the correct items. The conversation reward improves the distinctiveness of generated utterances, since it encourages the model to generate more informative utterances.

Human-based Evaluation We asked three workers to read 100 randomly selected contexts and the generated response of each model and to give a score between 0 and 2 to evaluate both the fluency and the informativeness of the responses. Table 2 shows the average score of the human evaluation results. The intraclass correlation coefficient(ICC) between workers is 0.49 for fluency scores and 0.71 for informativeness scores. Our framework improves the performance of the base model NTRD, especially in terms of informativeness, which shows the effectiveness of the proposed design of the conversation reward.

Case Study of the US

In this section, we present an example to demonstrate the quality of our proposed US. Please refer to the Appendix for more cases. In Table 3, we compare the output of our proposed US with the

Model	Recommendation metrics				Conversation metrics			
	Local metrics		Global metrics					
	ReR ↑	ReP ↑	GlobalRe ↑	Succ ↑	PPL ↓	Dist2 ↑	Dist3 ↑	Dist4 ↑
ReDial	0.7	-	-	-	28.1	0.225	0.236	0.228
KBRD	0.8	-	-	-	17.9	0.263	0.368	0.423
KGSF	1.1	-	-	-	8.3	0.302	0.431	0.508
NTRD	1.7	11.7	5.7	26.7	6.41	0.569	0.804	0.940
Ours (NTRD)	**3.2**	**22.3**	**12.2**	**50.5**	**6.23**	0.528	0.807	1.010
- w/o con-R	2.8	18.3	11.5	49.8	6.41	0.449	0.670	0.807
- w/o rec-R	1.9	14.2	6.7	29.9	**6.23**	**0.671**	**0.965**	**1.169**
RID*	3.1	-	-	-	54.1	0.518	0.624	0.598
MESE*	6.4	-	-	-	12.9	0.822	1.152	1.313

Table 1: **Machine-based Evaluation**. * indicates leveraging of pre-trained transformer-based models. We **bold** the best result on the same base model NTRD; underline the best result on all models.

Model	Fluency	Informativeness
NTRD	1.44	0.46
Ours (NTRD)	**1.65**	**0.79**

Table 2: **Human-based Evaluation**.

Case Study of the User Simulator	
User Preferences	*Iron Man 2*: seen, liked *The Avengers*: seen, liked *It*: unseen, liked *Ant Man*: seen, liked
Human:	I would like to watch any movie. Tell me any movie Like *Ant Man*.
CRS:	Have you seen *The Avengers*?
US:	I have seen that one. I also liked *Iron Man 2*.
- w/o PAM:	I haven't see that one. Is it good?
Human:	I really enjoyed that one. Yes and I liked it. Which another one would you recommend me?
CRS:	Do you like scary movies? Have you seen *It*?
US:	I have not seen that one. I will check it out.
- w/o PAM:	I have not seen that one.
Human:	I watched *Iron Man 2* and I liked it.
	...

Table 3: A case study comparing the user utterances generated by our model, the baseline, and the ground truth. We mark the item mentions in blue color.

baseline which has no preference-aware modules (PAM) and with the human written response. The US without PAM generates utterances simply with higher probability; that is, the user has not seen the movie recommended by the CRS. This may be contradictory to user preferences: When the CRS recommends the movie *The Avengers*, the baseline says that it has not seen the movie, which is not true, since the user has seen and liked it. Instead, our US with preference-aware modules provides the correct feedback for two recommendations, *The Avengers* and *It*. Furthermore, our US can actively express its preference to help the CRS know more about it: it actively says that it likes the movie *Iron Man 2*.

Case Study of the CRS Table 4 shows some examples of the responses generated by NTRD and our model given the same context. In the first case, the NTRD generates a general response that is not fluent with the context, while our model, which is the RL fine-tuned NTRD, recommends a movie with a description of the movie. In the second case, our model recommends the movie *It* which is a scary movie consistent with the user's short-term preference. These indicate that our framework can improve the informativeness of the responses by providing more details of the recommended movie. In the third case, our model recommends a movie and introduces its actress. However, the actress does not play any role in the movie, which shows the limitation of current CRSs, that is, it cannot guarantee the correctness of the generated information in a fine-grained way.

Remaining Challenges

Though effective, improvements in the CRS are highly dependent on the good quality of the US. Currently, we only use reinforcement learning to optimize the CRS. However, previous work (Tseng et al., 2021) shows that joint-learning of the dialog system and the US can further enhance the performance of the dialog system. We leave the joint learning of the CRS and the US for future work.

Case Study of the Conversational Recommendation System			
Context		CRS responses	
...		NTRD:	I think you will like it.
CRS:	If you like action movies that are also sci-fi, there's *Star Wars*.	Ours:	*Jumanji* is a good action packed comedy.
User:	Yes, I did like all of the *Star Wars* movies. I also like *Paycheck*.		
...		NTRD:	I like *Freddy vs. Jason*.
CRS:	I think *Scream* was a fail as far as being scary, but it was a good movie overall.	Ours:	*It* is a good one if you like scary movie.
User:	I have seen all of the *Halloween* and *Jason X*.		
CRS:	Hello. How is your night going?	NTRD:	*Fargo* is a good one.
User:	Hi. I'm looking for a movie. One that is funny but not too stupid.	Ours:	*The Naked Gun* is a funny movie with Jennifer Lawrence.

Table 4: Case studies comparing the CRS responses generated by the original NTRD and our improved model given the same contexts. We only give the last turn of the dialog history to save space here. We mark the item mentions in blue color, and the user preferences in red color.

4 Related Work

Conversational Recommendation System Current CRS studies can be roughly categorized into two directions (Liang et al., 2021): (1)Attribute-centric CRS (Deng et al., 2021; Lei et al., 2020b,a; Zhang et al., 2022). These systems ask questions about the user preferences of certain attributes or make recommendations at each turn and gradually narrow down the hypothesis space of items to make optimal recommendations. These studies focus on the recommendation part and use question/answer templates with attribute or item slots. They often use reinforcement learning to achieve better recommending and asking policies. (2)Open-ended CRS (Li et al., 2018; Chen et al., 2019; Zhou et al., 2020; Liang et al., 2021; Yang et al., 2022). These studies focus on understanding user preferences according to user utterances and generating fluent responses to recommend items. Compared to attribute-centric CRSs, open-ended CRSs have more free-style recommendations and more flexible interactions, which provides a better user experience. In this paper, we focus on open-ended CRSs and borrow the idea of improving the recommendation by reinforcement learning from the studies of attribute-centric CRSs.

User Simulator Traditional USs are rule-based such as the agenda-based user simulator (ABUS) (Schatzmann and Young, 2009; Li et al., 2016c). For different tasks, ABUS needs to design different hand-crafted structures, which poses challenges in scenario shifting. Data-driven US(Asri et al., 2016; Gur et al., 2018) is another line of work. A seq2seq model is used to generate semantic-level dialog acts (Asri et al., 2016; Gur et al., 2018; Tseng et al., 2021) or natural languages (Kreyssig et al., 2018). However, most of the USs are designed for task-oriented dialog systems and cannot be directly used for CRS. To the best of our knowledge, our work is the first to explore US for open-ended CRS that can generate consistent responses based on certain user preferences.

5 Conclusion

In this paper, we propose a framework to be packed with any CRS to improve its recommendation accuracy and language informativeness. We first build a User Simulator for open-ended CRS with three preference-aware modules to give the appropriate feedback to the CRS based on certain user preferences. We then fine-tune a pre-trained CRS with reinforcement learning based on its interaction with the US with two types of designed rewards. Experiments demonstrate that our framework can significantly improve the recall of the recommendation, and human evaluation shows that the generated language is more informative with more descriptions of the recommended items. For future work, the first is to use joint optimization of CRS and US to further improve the interactive qualities, and the second is to explore the generalizability of the framework to other domains of recommendation.

6 Limitations

The proposed framework has a limitation in terms of the large GPU resources required, as it necessitates double the memory compared to training a CRS alone. Due to this limitation, we have to forego the use of pre-trained language models such as BERT, which could have been beneficial in enhancing language quality, but their extreme memory requirements make it infeasible.

References

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 1151–1155. ISCA.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1803–1813. Association for Computational Linguistics.

Abhishek Das, Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2970–2979. IEEE Computer Society.

Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1431–1441. ACM.

Izzeddin Gur, Dilek Hakkani-Tür, Gökhan Tür, and Pararth Shah. 2018. User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, pages 900–906. IEEE.

Florian Kreyssig, Iñigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, pages 60–69. Association for Computational Linguistics.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020a. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 304–312. ACM.

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020b. Interactive path reasoning on graph for conversational recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2073–2083. ACM.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1192–1202. The Association for Computational Linguistics.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9748–9758.

Xiujun Li, Zachary C. Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016c. A user simulator for task-completion dialogues. *CoRR*, abs/1612.05688.

Zujie Liang, Huang Hu, Can Xu, Jian Miao, Yingying He, Yining Chen, Xiubo Geng, Fan Liang, and Daxin Jiang. 2021. Learning neural templates for recommender dialogue system. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7821–7833. Association for Computational Linguistics.

Alexandros Papangelis, Yi-Chia Wang, Piero Molino, and Gökhan Tür. 2019. Collaborative multi-agent dialogue model training via reinforcement learning. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, pages 92–102. Association for Computational Linguistics.

Jost Schatzmann and Steve J. Young. 2009. The hidden agenda user simulation model. *IEEE Trans. Speech Audio Process.*, 17(4):733–747.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. 2021. Transferable dialogue systems and user simulators. In *Proceedings of the 59th Annual*

Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 152–166. Association for Computational Linguistics.

Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph. *CoRR*, abs/2110.07477.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2022. Improving conversational recommendation systems' quality with context-aware item meta-information. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 38–48. Association for Computational Linguistics.

Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Bo Long, and Jian Pei. 2022. Multiple choice questions based multi-interest policy learning for conversational recommendation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2153–2162. ACM.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1006–1014. ACM.

A Appendix

A.1 User Preferences Extension

For each user u, we know its preference vectors of a constrained set of movies $\mathcal{I}^u_{known}$ from the dataset. We need to extend the user preference to each movie $i \in \mathcal{I}$, since during the interaction between the US and the CRS, the CRS may recommend a movie that is not in $\mathcal{I}^u_{known}$. Therefore, for each movie i_{unk} that is not in $\mathcal{I}^u_{known}$, we consider that the user u has not seen it. We then calculate the cosine similarities between i_{unk} and each movie in $\mathcal{I}^u_{known}$ and set the like/dislike label of i_{unk} the same as the closest movie to it, *i.e.*,

$$i^* = _{i \in \mathcal{I}_{known}} \cos\left((i), (i_{unk})\right) \quad (7)$$

, where (i) returns the embedding of the movie i, and the user u has the same like/dislike label to i_{unk} and i^*.

A.2 Hyper-parameters for Reproducing

The Hyper-parameters of RL

In this section, we introduce the detailed setting of reinforcement learning of the Conversational Recommendation System (CRS). To train the recommendation component, we only use recommendation rewards *i.e.*, $\alpha = 1, \beta = 0$, and for the conversation component, we only use conversation rewards *i.e.*, $\alpha = 0, \beta = 1$. Detailed reward values are listed in Table 5.

Reward Type	Value
r_{rec_suc}	5
r_{rec_fial}	0
r_{con_rec}	1
r_{con_info}	5
r_{con_rep}	-5

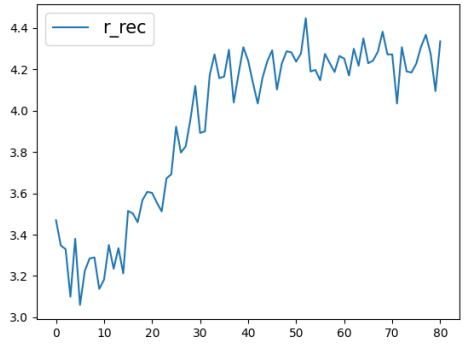
Table 5: The reward values of the RL of CRS.

The Hyper-parameters of the User Simulator

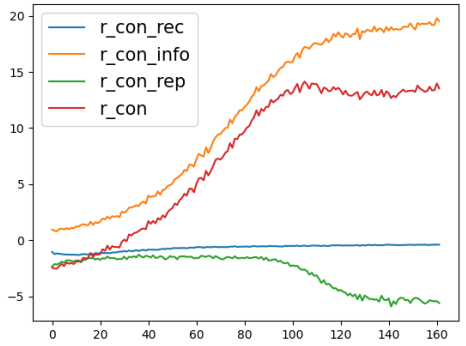
In this section, we introduce the hyper-parameters of the architecture of the User Simulator (US). The US consists of a template generator and an item selector, where the backbone of each component is a transformer with multi-head attentions. We use $\lambda = 5$ as the weight for generation loss L_{gen} and item selection loss L_{sle}. We train the US using Adam optimizer with a learning rate of 1e-3 and a batch size of 32 for 90 epochs. Detailed hyper-parameters for architecture are listed in Table 6.

Hyper-parameters	Value
num_attention_heads	2
num_hidden_layers	2
embedding_size	300
hidden_size	128
intermediate_size	300
gradient_clipping	0.1
dropout_prob	0.1
attention_dropout_prob	0
hidden_act	“relu"
relu_dropout_prob	0.1
max_context_length	256
max_response_length	30
vocab_size	17641

Table 6: The detailed hyper-parameters of the US.



(a)



(b)

Figure 3: Averaged recommendation rewards (a) and conversation rewards (b) curve of the policies.

A.3 Reward Curves of RL

Figure 3 shows the average reward the policies receive at each 2000 steps during the training process. For the recommendation component, the rewards quickly converge, as the action space is limited by the number of the entire set of items $\mathcal{I}$. However, the reinforcement learning of the conversation component is much more unstable, since it needs to generate fluent utterances, and the action space

	NTRD with the US	Ours (NTRD) with the US
User Preferences:	*Taken*: unseen, like *Avengers: Infinity War (2018)*: seen, like *Die Hard*: unseen, like *Spider-Man*: seen, like	*Black Panther*: seen, like *The Sixth Battalion*: unseen, like *Thor: Ragnarok*: unseen, like *The Avengers*: unseen, like
(given) CRS:	What kind of movies do you like? Have you seen *Avengers: Infinity War*? or *Spider-Man*?	
(given) US:	Yes and i liked them. Those are awesome. I like that kind of movies.	
CRS:	*Thor: Ragnarok* is a good one.	*Black Panther* is a great comedy movie.
US:	I haven't seen that one yet.	Yes, I have seen that one. I liked it.
CRS:	*The Avengers* is a good one.	*The Avengers* is a great movie!
US:	I haven't seen that one yet.	I have not seen that one yet.
CRS:	I haven't seen that one. I'll check it out.	*Spider-Man* is a great movie.
US:	I will have to check it out.	Yes, I have seen that one. It was great.
CRS:	I hope you enjoy them!	*Die Hard* is a great movie.
US:	I will have to check it out.	I have not seen that one yet.
CRS:	I hope you enjoy them!	*Taken* is a classic.
US:	Thank you for your help.	I have not seen that one yet.
User Preferences:	*Porky's*: unseen, like *The Avengers*: unseen, like *Fist Fight*: unseen, like *Homecoming*: seen, like *Taken*: unseen, like	*Pineapple Express*: seen, like *The Hangover*: unseen, like *Animal House*: unseen, like *House*: seen, like
(given) CRS:	Hi there. Do you like movies?	
(given) US:	Hi. I like fun movies. I like *Homecoming*.	
CRS:	I have not seen that one. Is it a good one.	Have you seen *Porky's*?
US:	I have not seen that one.	I haven't seen either of those.
CRS:	*Pineapple Express* is a good one.	*Animal House* is a great comedy with Kevin Hart.
US:	I have not seen that one.	I haven't seen that one yet.
CRS:	*The Avengers* is a good one.	*Fist Fight* is a good comedy also.
US:	I have not seen that one.	I 'm not sure if I have seen that one.
CRS:	I have not seen that one. Is it a good one.	*The Hangover* is a good comedy with Bradley Cooper.
US:	I have not seen that one.	I haven't seen that one either.
CRS:	I have not seen that one. Is it a good one.	*Taken* is a classic.
US:	I have not seen that one.	I have not seen that one yet.

Table 7: Interactive Cases. Comparison of CRSs before (NTRD) and after (Ours) fine-tuning with reinforcement learning.

is infinite. Thus, we use a small learning rate and more steps to train the component. During training, the total reward (red curve) increases and converges. However, the convergence status consists of a high informative reward and a low repetition reward, which is caused by the model keeping generate simple but informative utterances like "xxx is a good comedy". This shows a limitation of our design of informative rewards: Though simple and effective, it is only a binary reward with informative or noninformative, which lacks the ability to judge the level informativeness. Therefore, the utterance "xxx is a good sci-fi" and "xxx is a sci-fi about a human trying to find another habitable planet." would get the same informative score, but obviously the latter one contains more information about the movie and deserves a higher score. In future work, we will design a better informative reward to encourage the model to generate more informative utterances and make the recommendations more persuasive.

A.4 Interactive Cases

Table 7 shows two cases of interactive conversations between the US and CRSs before and after fine-tuning with reinforcement learning. Given the first turn of the conversation, the US and CRS continue to interact for 5 turns. In each dialog, the US is based on different user preferences. Generally speaking, our CRS has a more fluent conversation with the US. The NTRD tends to generate generic utterances, and the conversation becomes stuck in an infinite loop of repetitive responses. Another improvement of our CRS is that it generates more informative utterances when recommending items, which are highlighted with red. However, as we discussed in the paper, there may be some mistakes when talking about actors / actresses: while Bradley Cooper plays an important role in *The Hangover*, Kevin Hart does not play any role in *Animal House*.