

# Self-Supervised Bot Play for Transcript-Free Conversational Critiquing with Rationales

SHUYANG LI, Computer Science and Engineering, University of California, San Diego, La Jolla, United States

BODHISATTWA PRASAD MAJUMDER, Computer Science and Engineering, Allen Institute for Artificial Intelligence, Seattle, United States

JULIAN MCAULEY, Dept of Computer Science and Engineering, University of California, San Diego, La Jolla, United States

---

Conversational critiquing in recommender systems offers a way for users to engage in multi-turn conversations to find items they enjoy. For users to trust an agent and give effective feedback, the recommender system must be able to *explain* its suggestions and rationales. We develop a two-part framework for training multi-turn conversational critiquing in recommender systems that provide recommendation rationales that users can effectively interact with to receive better recommendations. First, we train a recommender system to jointly suggest items and explain its reasoning via subjective rationales. We then fine-tune this model to incorporate iterative user feedback via self-supervised bot-play. Experiments on three real-world datasets demonstrate that our system can be applied to different recommendation models across diverse domains to achieve state-of-the-art performance in multi-turn recommendation. Human studies show that systems trained with our framework provide more useful, helpful, and knowledgeable suggestions in warm- and cold-start settings. Our framework allows us to use only product reviews during training, avoiding the need for expensive dialog transcript datasets that limit the applicability of previous conversational recommender agents.

CCS Concepts: • **Information systems** → **Recommender systems** • **Computing methodologies** → **Learning from critiques**;

Additional Key Words and Phrases: Conversational recommendation, critiquing

## ACM Reference Format:

Shuyang Li, Bodhisattwa Prasad Majumder, and Julian McAuley. 2024. Self-supervised Bot-play for Transcript-free Conversational Critiquing with Rationales. *ACM Trans. Recomm. Syst.* 3, 1, Article 7 (August 2024), 20 pages. <https://doi.org/10.1145/3665502>

---

## 1 INTRODUCTION

Traditional recommender systems often give static suggestions, affording users no way to meaningfully express their preferences and feedback. Conversational recommendation allows users to

---

Authors' Contact Information: Shuyang Li, Computer Science and Engineering, University of California, San Diego, La Jolla, Meta 50 Hudson Yards New York, NY 10001, California, United States; e-mail: shuyangli94@gmail.com; Bodhisattwa Prasad Majumder, Computer Science and Engineering, Allen Institute for AI, 2157 N Northlake Way #110, Seattle, WA 98103, United States; e-mail: bmajumde@ucsd.edu; Julian McAuley, Dept of Computer Science and Engineering, University of California, San Diego 9500 Gilman Drive La Jolla, CA 92093, United States; e-mail: jmcauley@eng.ucsd.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 2770-6699/2024/08-ART7  
<https://doi.org/10.1145/3665502>

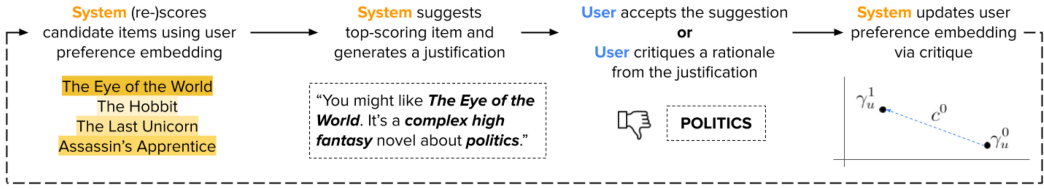


Fig. 1. In our conversational recommendation workflow, the system scores candidates and generates a justification for the top item. If the user critiques a rationale, then the system uses the critique to update the latent user representation.

interact with agents and suggestions, increasing their willingness to trust and accept recommendations [26]. Techniques for conversational recommendation are based on the *paradigm* of conversation: how an agent can explain their suggestions and how users can give feedback.

Recent work has explored conversational recommendation through dialog agents trained to ask the user questions in free-form dialog [36]. Such models require large training corpora comprising transcripts from crowd-sourced recommendation games [8]. To create high-quality training data, crowd-workers must be knowledgeable about many items in the target domain—this expertise requirement limits data collection to a few common domains like movies. It is thus difficult to scale dialog-based recommenders to domains where users have specific preferences about subjective rationales but no dialog transcripts exist (e.g., food and literature).

We address this challenge of data scarcity by proposing a framework for training conversational recommender systems based on conversational critiquing and self-supervised bot-play. Our approach reflects an *realistic interactive paradigm* where the agent suggests items and explains their rationale, while the user specifies their preferences via specific feedback to guide the next turn's suggestions [40]. Our framework does not rely on supervised dialog examples and can be applied to *any* setting where product reviews or opinionated text can be harvested.

We propose a framework comprising two parts: First, we learn to jointly recommend items and generate justifications based on subjective rationales, leveraging ideas from conversational critiquing systems [29, 37] trained via next-item recommendation. We then fine-tune our model for multi-turn recommendation via multiple turns of bot-play in a recommendation game based on natural-text product reviews and simulated critiques. We specifically make our framework adaptive to accept user critiques by simulating such feedback during the training time and evaluate our model at the test time for effective and efficient recommendations.

Our framework is model-agnostic—we apply our method to two different underlying recommendation architectures [27, 28] of differing sizes and evaluate our models on three large real-world recommendation datasets with user reviews but no dialog transcripts. Our method can provide more useful explanations and better adapts to user feedback compared to **state-of-the-art (SOTA)** conversational critiquing systems—users interacting with our rationales reach their goal items faster and with greater success. We conduct a study with real users, showing that our models can effectively help users find desired items in real time, even in a cold-start setting.

We summarize our main contributions as follows: (1) We present a framework for training conversational critiquing systems using bot-play on historical user reviews, without the need for large collections of human dialogs; (2) we apply our framework to two popular recommendation models (**BPR-Bot** and **PLRec-Bot**), with each showing superior or competitive performance in comparison to SOTA recommendation and critiquing methods; (3) we demonstrate through human evaluation and user studies that models trained with our bot-play framework are more useful, informative, knowledgeable, and adaptive compared to SOTA baselines.

## 2 RELATED WORK

*Justifying Recommendations.* Users prefer recommendations that they perceive to be transparent or justified [30, 32]. Some early recommender systems presented the same attributes of suggested items to all users [31, 34] but did not attempt to personalize the justifications. Another line of work attempts to generate natural language explanations of recommendations. McAuley et al. [22] mine key attributes from textual reviews via topic extraction. These attributes can be expanded into explanatory sentences via template-filling [39] or recurrent language models [24]. Due to the unstructured nature of these justifications, however, sentence-level justifications have not been used for iteratively refining recommendations. In this work, we allow the user to provide feedback about specific rationales mentioned across natural language product reviews in large recommendation datasets.

*Conversational Critiquing.* Critiquing systems allow users to incrementally construct preferences, mimicking how humans refine their preferences based on conversation context [25, 33]. Early critiquing methods treated user feedback as hard constraints to shrink the search space [3]. Wu et al. [37] introduced a critiquing model with justifications comprising natural language attributes mined from user reviews—with which users can then interact. Antognini et al. [2] provide a single-sentence explanation alongside a set of rationales, requiring users to interact only with the rationale set.

Luo et al. [20] use a **variational auto-encoder (VAE)** [10] for joint recommendation and justification, learning a bi-directional mapping function between latent user and rationale representations. Current critiquing techniques are either trained only for next-item recommendation, or to handle a single turn of critiquing [1], and struggle to incorporate feedback in multi-turn settings. We adopt techniques for encoding user feedback from critiquing systems [19], but we introduce a multi-step, model-agnostic bot-play method to explicitly train our models for multi-turn conversational recommendation.

*Dialog Agents for Recommendation.* We view recommenders as domain experts who can elicit preferences from human customers and suggest appropriate items over the course of a session [4]. A recent line of work formulates conversational recommendation as goal-oriented dialog: At each turn, the user is either (a) asked if they prefer a specified attribute or (b) recommended an item [5, 38]. Other question-answering models use reinforcement learning to dialog policies for when to ask users about attributes, updating a cumulative belief state of item attributes [13, 14]. These models ask templated questions and surface recommendations from an open candidate pool without explaining their reasoning to the user.

Another line of research treats conversational recommenders as free-text dialog agents that interact with users via natural language utterances. Bot-play has been explored as a way to train such dialog agents [8, 17], which requires models to be trained and fine-tuned using existing dialog transcripts. Such agents are thus limited to domains where crowd-sourced workers can accurately play the roles of expert and seeker to collect data via Wizard-of-Oz setups [6]. By allowing users to critique natural text rationales of a suggested item, our framework for conversational recommendation allows for multi-turn recommenders that can be trained using only product review texts—which are available in a wide range of domains. In Table 1, we compare our approach to recent frameworks for critiquing and dialog agents for conversational recommendation.

## 3 MODEL

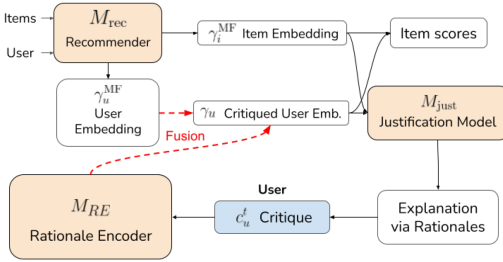
Our model comprises (Figure 2):

- (1) a recommender model  $M_{\text{rec}}$  that ranks items based on their suitability for a user;
- (2) a justification module  $M_{\text{just}}$  that predicts rationales for a given recommendation; and

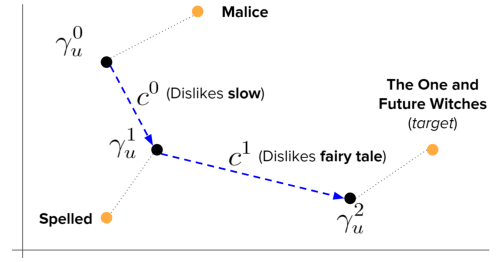
Table 1. Critiquing Systems (top) Are Not Equipped for Multi-turn Interactions

Paradigm	Model	Year	Justifies Suggestions	Multi-turn Conversations	Transcript-free
Conversational Critiquing	LLC [19]	2020	✗	✗	✓
	CE-VAE [20]	2020	✓	✗	✓
	M& M VAE [1]	2021	✓	✗*	✓
Question & Answer	SAUR [38]	2018	✗	✓	✓
	EAR [13]	2020	✗	✓	✓
	SCPR [14]	2020	✗	✓	✓
Dialog Agents	Li et al. [17]	2018	✗	✓	✗
	Kang et al. [8]	2019	✓	✓	✗
	Zhou et al. [40]	2020	✓	✓	✗
<b>Ours</b>			✓	✓	✓

(\*M& M VAE Is Trained for a Single Turn of Critiquing). Q & A systems (middle) ask the user to build a list of search criteria but do not provide rationales for recommended items. Dialog agents (bottom) learn multi-turn behavior via large corpora of domain-specific transcripts. Our framework allows us to train conversational recommenders without costly transcript data.



(a) Model Architecture



(b) Latent Critiquing Process

Fig. 2. (a) Given a user, items, and rationale critique vector, our model encodes the critique  $M_{RE}(c_u^t)$  and fuses it with the user embedding  $\vec{\gamma}_u^{MF}$  via critiquing function  $f_{crit}$ . The fused user ( $\vec{\gamma}_u$ ) and item ( $\vec{\gamma}_i$ ) representations are then used to predict the justification and score items. An example is shown in (b), where user feedback about the rationales *slow* ( $c^0$ ) and *fairy tale* ( $c^1$ ) modify our prior latent user preference vector to bring it closer to the target item (“The One and Future Witches”).

- (3) an interactive critiquing function  $f_{crit}$  that allows users to edit a rationale and modifies the user representation to recommend a different item on the next turn.

We support multi-step critiquing (Figure 2): At each turn, a user may indicate which rationales they dislike about the current suggestions via a critique  $c^t$ . The critiquing function then modifies the latent user representation  $\vec{\gamma}_u$  via the critique to bring it closer to the target item.<sup>1</sup>

### 3.1 Recommender System

Our method can be applied to any recommender that learns user and item representations. We show its effectiveness with two popular methods:

**Bayesian Personalized Ranking (BPR)** [27] is a matrix factorization recommender system that aims to decompose the interaction matrix  $R \in \{0, 1\}^{|U| \times |I|}$  into user and item representations [11]. BPR optimizes a ranked list of items given implicit feedback (binary interactions between users and items). Scores are computed via inner product of  $h$ -dimensional user and item em-

<sup>1</sup>Code for this system can be found at [https://github.com/shuyangli94/convrec\\_botplay](https://github.com/shuyangli94/convrec_botplay)

Table 2. Notation Used in This Article

Notation	Description
$U, I, A$	User, item, and rationale sets .
$\mathbf{R} \in \{0, 1\}^{ U  \times  I }$	Matrix of binary user-item interactions.
$\mathbf{K}^U \in \mathbb{R}^{ U  \times  A }$	User aspect frequency matrix; $\mathbf{K}_{u,a}^U$ is the number of times user $u$ mentioned aspect $a$ in their reviews.
$\mathbf{K}^I \in \{0, 1\}^{ I  \times  A }$	Binary matrix, where $\mathbf{K}_{i,a}^I$ is 1 if and only if aspect $a$ was used to describe item $i$ in any of reviews.
$\vec{y}_u, \vec{y}_i \in \mathbb{R}^h$	Learned $h$ -dimensional user and item embeddings.
$\hat{x}_{u,i} \in \mathbb{R}$	The predicted score of item $i$ for user $u$ .
$\hat{k}_{u,i} \in \{0, 1\}^{ A }$	Predicted justification (binary across all aspects).
$\vec{c}_u^t \in \mathbb{R}^{ A }$	The cumulative critique vector representing the user's evolving opinion about each aspect.
$\vec{m}_u^t \in \{0, 1\}^{ A }$	The user critique vector at turn $t$ . $m_{u,a}^t$ is 1 if and only if the user critiqued aspect $a$ at turn $t$ .

beddings:  $\hat{x}_{u,i} = \langle \vec{y}_u^{\text{MF}}, \vec{y}_i^{\text{MF}} \rangle$ . At training time, the model is given a user  $u$ , observed item  $i$ , and unobserved item  $j$ . We maximize the likelihood that the user prefers the observed item:

$$\mathcal{L}_R = P(i >_u j | \Theta) = \sigma(\hat{x}_{u,i} - \hat{x}_{u,j}),$$

where  $\sigma$  represents the sigmoid function  $\frac{1}{1+e^{-x}}$ .

**Projected Linear Recommendation (PLRec)** instead learns user- and item-embeddings separately [28]. First, PLRec takes a low-rank SVD approximation of  $\mathbf{R}$  such that  $\mathbf{R} = \mathbf{U}\Sigma\mathbf{V}^T$ . The  $h$ -dimensional user embeddings are computed by multiplying the user rating vector  $\vec{r}_u$  with  $\mathbf{V}$ :  $\vec{y}_u^{\text{MF}} = \vec{r}_u\mathbf{V}$ .

To then efficiently learn the item embeddings, PLRec fixes  $\mathbf{V}$  and learns the *item embedding*  $\mathbf{W}$  by optimizing the linear equation:

$$\arg \min_{\mathbf{W}} \sum_u \|\vec{r}_u - \vec{r}_u\mathbf{V}\mathbf{W}^T\|_2^2 + \Omega(\mathbf{W}).$$

We thus learn an item embedding matrix that most faithfully recovers the user's observed ratings  $\vec{r}_u$ , giving us an  $h$ -dimensional item embedding ( $\vec{y}_i^{\text{MF}} = \mathbf{W}i$ ).

### 3.2 Justification Module

Our justification model (rationale prediction head) consists of a fully connected network with two  $h$ -dimensional hidden layers and a final  $h \times |A|$  projection layer to predict a scalar score  $s_{u,i,a}$  for each natural language rationale  $a$ . We find empirically that simply taking the sum of user and item embeddings as an  $h$ -dimensional input vector to  $M_{\text{just}}$  works equally well or better compared to taking the element-wise mean or concatenating the two embeddings.

At training time, we incorporate a rationale prediction loss  $\mathcal{L}_A$  by computing the **binary cross-entropy (BCE)** for each rationale given the likelihood the user cares about the rationale ( $p_{u,i,a}$ ):

$$\mathcal{L}_A = -\frac{1}{|A|} \sum_{a=0}^{|A|} \mathbf{K}_{i,a}^I \cdot \log p_{u,i,a} + (1 - \mathbf{K}_{i,a}^I) \cdot \log(1 - p_{u,i,a}).$$

At inference time, we again compute the likelihood for each rationale  $p_{u,i,a} = \sigma(s_{u,i,a})$  and sample from the Bernoulli distribution with the likelihood  $p_{u,i,a}$  to determine which rationales  $a$  appear in the justification.

### 3.3 Critiquing Function

We posit that the user’s latent representation is partially explained by their written reviews. We thus learn a lightweight rationale encoder  $M_{\text{RE}}$ —a linear projection from the representation space for rationales to the representation space for user preferences:  $M_{\text{RE}}(\vec{c}_u^t) = \mathbf{W}_{\text{RE}}^T \vec{c}_u^t + b$ , where  $\vec{c}_u^t \in \mathbb{Z}^{|A|}$  is the critique vector representing the strength of a user’s preference for each rationale, and  $\mathbf{W}_{\text{RE}}$  and  $b$  are the learned linear weights and bias, respectively. We learn a lightweight linear encoder for rationales to fairly compare against prior work [19] and isolate the impact of our bot-play framework (Section 3.4) on critiquing performance.

We fuse this rationale encoding with the latent user embedding from  $M_{\text{rec}}$  to form the final user preference vector:

$$\vec{\gamma}_u = f_{\text{crit}}(\vec{\gamma}_u^{\text{MF}}, M_{\text{RE}}(\vec{c}_u^t)).$$

For both models, we fuse via the element-wise mean of the two vectors:  $f_{\text{crit}}(a, b) = \frac{a+b}{2}$ . In training, the rationale encoder takes in the user’s rationale history:  $\vec{c}_u^t = \mathbf{K}_u^U$ .

*Critiquing with Our Models.* To perform conversational critiquing with a model trained using our framework, we adapt the latent critiquing formulation from Luo et al. [19], as shown in Figure 1. At each turn  $t$  of a session for user  $u$ , the system assigns scores  $\hat{x}_{u,i}^t$  for all candidate items  $i$  and presents the user with the highest scoring item  $\hat{i}$ . The system also justifies its prediction with a set of predicted rationales  $\hat{k}_{u,i}^t$ . The user may either accept the recommended item (ending the session) or critique a rationale from the justification:  $a \in \{a | \hat{k}_{u,i,a} = 1\}$ .

Given a user critique, the system modifies the predicted scores for each item and presents the user with a new item and justification:

$$\begin{aligned} \hat{x}_{u,i}^{t+1} &= M_{\text{rec}}(\hat{\gamma}_u^{t+1}, i) \\ \hat{k}_{u,i}^{t+1} &= M_{\text{just}}(\hat{\gamma}_u^{t+1}, i) \\ \hat{\gamma}_u^{t+1} &\leftarrow f_{\text{crit}}(\hat{\gamma}_u^t, M_{\text{RE}}(\vec{c}_u^t)). \end{aligned}$$

Effectively, we encode a user critique via  $M_{\text{RE}}$  to modify our prior for the user’s preferences; we then use this modified user preference representation  $\hat{\gamma}_u^{t+1}$  to re-rank the items presented to the user and propose new rationales.

At inference time, we initialize the cumulative critique vector  $\vec{c}_u^t$  with the user’s rationale history ( $\vec{c}_u^0 = \mathbf{K}_u^U$ ). It is then updated via:

$$\vec{c}_u^t = \vec{c}_u^{t-1} - \max(\mathbf{K}_u^U, \mathbf{1}_{|A|}) \odot \vec{m}_u^t; \quad \vec{c}_u^0 = \mathbf{K}_u^U,$$

where  $\odot$  is element-wise multiplication and  $\mathbf{1}_{|A|}$  is a vector of ones. Here, the critique should match the strength of a user’s previous opinion of the rationale  $\mathbf{K}_u^U$ . We smooth the user’s prior opinions via  $\max(\cdot, \mathbf{1}_{|A|})$  to ensure a non-zero effect from each critique even if the user has not mentioned the rationale in their previous reviews. In future work, we hope to explore other ways to smooth user aspect opinions, including methods to inject noise and model uncertainty in user’s opinions about aspects they have not mentioned in reviews.

### 3.4 Training

To train our BPR-based model, we jointly optimize each component. Each training example comprises a user and observed/unobserved items. We predict scores for each item:

$$\hat{x}_{u,i} = \langle \vec{\gamma}_u^{\text{MF}} + M_{\text{RE}}(\mathbf{K}_u^U), \vec{\gamma}_i \rangle.$$

**ALGORITHM 1:** Bot play framework for fine-tuning conversational recommenders.

---

Recommender and Justifier  $M_{\text{rec}}, M_{\text{just}}$ ;  
 Critique fusion function  $f_{\text{crit}}$ ;  
 Seeker model  $M_{\text{seeker}}$ ;  
**for** each user  $u$  **do** > Fine-tune across users in training set  
     **for** goal item  $g \in I_u^+$  **do** > Sample goal item from reviewed items  
         initialize  $\vec{y}_u^1$  from  $M_{\text{rec}}, \mathcal{L} = 0$ ;  
         **for** turn  $t \in \text{range}(1, T)$  **do** > Simulate up to  $T$  turns of user feedback  
              $\hat{x}_{u,i}^t = M_{\text{rec}}(\vec{y}_u^t, i) \forall i \in I$ ;  
              $\mathcal{L} \leftarrow \mathcal{L} + \delta^t \cdot (\mathcal{L}_{\text{CE}}(g, \hat{x}_{u,i}^t) + \frac{1}{2} \mathcal{L}_A)$ ;  
              $\hat{i}^t = \arg \max_i \hat{x}_{u,i}^t$ ;  
             **if**  $\hat{i}^t = g$  **then** > Terminate session if goal item recommended.  
                 | break with success  
              $\hat{k}_{u,\hat{i}^t} = M_{\text{just}}(\vec{y}_u^t, \vec{y}_{\hat{i}^t})$  > Generate justification for suggested item  
             simulate user critique using  $M_{\text{seeker}}$ :  $\vec{c}_u^t$ ;  
              $\vec{y}_u^{t+1} \leftarrow f_{\text{crit}}(\vec{y}_u^t, \vec{c}_u^t)$  > Update user latent representation  
**return** fine-tuned agent

---

We first compute the BPR loss (see Section 3.1) with the predicted observed/unobserved scores. We add the rationale prediction loss, scaled by a constant  $\lambda_{\text{KP}}$  to the ranking loss for our training objective:  $\mathcal{L} = \lambda_{\text{KP}} \mathcal{L}_A - \mathcal{L}_R$ . We find empirically that  $\lambda_{\text{KP}} \in \{0.5, 1.0\}$  works well.

To train our PLRec-based model, we follow Luo et al. [19] and separately optimize the recommendation ( $M_{\text{rec}}$ ), justification ( $M_{\text{just}}$ ), and rationale encoding ( $M_{\text{RE}}$ ) modules. In future work, we hope to explore methods to jointly train such modules to take advantage of potential coupling between the recommendation and justification tasks.

We tightly couple  $M_{\text{RE}}$  and  $M_{\text{rec}}$  by co-embedding the item critique information with the user preference embeddings learned from  $M_{\text{rec}}$  via the linear regression:

$$\arg \min_{W,b} \sum_u \|\vec{y}_u^{\text{MF}} - M_{\text{RE}}(\mathbf{K}_u^U)\|_2^2 + \Omega(\mathbf{W}_{\text{RE}}).$$

Finally, we optimize the rationale prediction (justification) loss  $\mathcal{L}_A$  to train the justification head.

*Learning to Critique via Bot Play.* We propose to instill recommender systems with the ability to understand and respond to user critiques *in the absence of conversational trace data*. As such, we have no ground truth labels for how users will react to recommended justifications (i.e., which aspect they will critique). To address this, we propose a framework for critiquing via bot-play that simulates user sessions when provided just a set of user reviews. To mimic real-world conversational patterns, we train an *expert* model to engage with a *seeker* agent who plays the user role—accepting/rejecting recommendations and selecting aspects to critique. For each training example (user and a goal item they have reviewed), we allow the expert and seeker models to converse with the goal of recommending the goal item. We first pre-train our expert model (recommender, justifier, and rationale encoder).

In this article, we demonstrate that a simple rule-based seeker model can help efficiently train our conversational critiquing system to more effectively adapt user preferences from critiques. This seeker model proceeds from a simple prior: Provided a target item and justification, it selects the most popular rationale present in the justification but not the target’s historical rationales  $\mathbf{K}_i^I$  to critique. We use this critique selection method to approximate human behavior, assuming that most users with somewhat specific preferences but some domain knowledge tend to critique

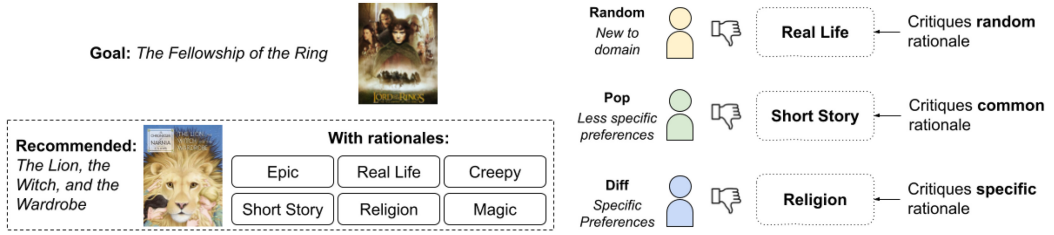


Fig. 3. Example of user behaviors after receiving a book recommendation with rationales. A new reader may *randomly* select a rationale to critique. Readers with less specific preferences may critique common/*popular* rationales. A knowledgeable reader with specific preferences will critique a specific *weakest* (most different from their target) rationale.

common rationales (Figure 3). In future work, we hope to explore more complex seeker models that can be jointly or iteratively optimized with the expert during the bot-play process.

We fine-tune the expert by maximizing its reward (minimizing loss) in the bot-play game (Algorithm 1). We end the session after the goal item is recommended or a maximum session length of  $T = 10$  turns is reached. We define the expert’s loss to target both surfacing the correct recommendation and inferring the user’s ground truth preferences per turn:

$$\mathcal{L}^{\text{expert}} = \sum_t^T \delta^{t-1} \cdot (\mathcal{L}_{\text{CE}}(g, \hat{x}_{u,i}^t) + \frac{1}{2} \mathcal{L}_A),$$

where  $\delta$  is a discount factor to encourage successfully recommending the goal item at earlier turns,  $\mathcal{L}_{\text{CE}}(g, \hat{x}_{u,i}^t)$  is the cross-entropy loss between predicted scores (transformed into a probability distribution via softmax) and the goal item, and  $\mathcal{L}_A$  is the binary cross-entropy rationale loss defined in Section 3.2. We find that a discount factor of  $\delta = 0.9$  is effective for both BPR- and PLRec-based conversational recommenders.

## 4 EXPERIMENTAL SETTING

We select hyperparameters for our initial models via AUC and for bot-play fine-tuning via the success rate at 1 (SR@1) on the validation set. We train each model once, taking the median of three evaluation runs per experimental setting. For baseline models, we re-used the authors’ code.

All experiments were conducted on a machine with a 2.2 GHz 40-core CPU, 132 GB memory, and one RTX 2080Ti GPU. We use PyTorch version 1.4.0 and optimize our models using the Rectified Adam [18] optimizer. Best hyperparameters for each base recommender system model are shown in Table 3. We perform hyperparameter search over a coarse sweep of:  $h \in [2, 512]$ ,  $LR \in [1e - 5, 1e - 2]$ ,  $\lambda \in [1e - 5, 1e - 2]$ . Model parameter sizes are a function of the hidden dimensionality  $h$  and number of items  $|I|$  and users  $|U|$  and is dominated by  $h \cdot (|I| + |U|)$ .

### 4.1 Datasets

We evaluate our models on three public real-world recommendation datasets with 100K+ reviews each: Goodreads Fantasy (Books) [35], BeerAdvocate (Beer) [22], and Amazon CDs & Vinyl (Music) [23]. We keep only reviews with positive ratings, setting thresholds of  $t > 4.0$  for Beer and Music and  $t > 3.5$  for Books. All reviews in these datasets are in English; we hope to extend our work to identify related rationales in multi-lingual reviews in the future. We partition each dataset into 50% training, 20% validation, and 30% test splits.

We follow the pipeline of Wu et al. [37] to extract subjective rationales (Table 4) from user reviews:

Table 3. Best Hyperparameter Settings for Each Base Recommendation Model

Dataset	Model	$h$	LR	$\lambda_{L2}$	$\lambda_{KP}$	$\lambda_c$	$\beta$	Iterations	Epoch
Books	BPR [27]	20	0.001	0.01	0.5	–	–	–	200
	PLRec [28]	50	–	80	–	–	–	10	–
	CE-VAE [20]	100	0.0001	0.0001	0.01	0.01	0.001	–	300
Beer	BPR	20	0.001	0.01	0.5	–	–	–	200
	PLRec	50	–	80	–	–	–	10	–
	CE-VAE	100	0.0001	0.0001	0.01	0.01	0.001	–	300
Music	BPR	20	0.01	0.1	1.0	–	–	–	100
	PLRec	400	–	1000	–	–	–	10	–
	CE-VAE	200	0.0001	0.0001	0.001	0.001	0.001	–	600

Linear critiquing methods (UAC, BAC, LLC-Score, LLC-Rank) use PLRec as a base model. BPR-Bot uses BPR as a base model, and PLRec-Bot uses PLRec as a base model.

Table 4. Dataset Statistics, Including Number of Unique Rationales (R), Sample Subjective Rationales from User Reviews, and Average Unique Rationales per User, Item, and Review

	Users	Items	Reviews	Uniq. R	Sample Subjective Rationale	R/User	R/Item	R/Review
Books	13,889	7,649	654,975	75	Realistic, Strong Female	25.0	27.0	1.77
Beer	6,369	4,000	935,524	75	Smoky, Citrus, Nutty	54.6	60.2	7.39
Music	5,635	4,352	119,081	80	Techno, Reggae, Catchy	16.5	20.0	2.54

- (1) Extract high-frequency unigram and bigram noun- and adjective phrases;
- (2) Prune bigram keyphrases using a **Pointwise Mutual Information (PMI)** threshold, ensuring rationales are statistically unlikely to have randomly co-occurred; and
- (3) Represent reviews as sparse binary vectors indicating whether each rationale was expressed in the review.

These noun/adjective phrase rationales describe various subjective qualities of the items. For beers, users commonly describe the malt (e.g., roasted) and taste (e.g., citrus). For music, rationales range from perceived genres (e.g., techno) to emotions (e.g., soulful). Users describe books by reacting to character descriptions (e.g., strong female) and settings (e.g., realistic). Our framework is agnostic to the rationale format, and in future work, we aim to extend our models to encode full sentences and utterances as critiques. All reviews in this dataset are in English; we hope to extend our work to identify related rationales in multi-lingual reviews in the future.

## 4.2 Multi-step Critiquing

As our data domains consist of offline recommendation data with no conversational traces, we evaluate multi-step critiquing performance by simulating recommendation sessions [15, 19]. We simulate user sessions following Algorithm 1, with two main differences: (1) We randomly sample user  $u$  and their goal item  $g$  from the *test* set, and (2) we do not compute loss or update our model during a session. We set a maximum session limit of  $T = 10$  turns.

Evaluating multi-step recommendation performance via user simulation does have limitations compared to real user or oracle conversational data. In the real world, users may have different degrees of knowledge about certain aspects of each domain (e.g., a beer drinker may have deep knowledge of wheat beers but little knowledge of stouts and their flavor profiles), and the way they critique recommendations may differ between user groups. To evaluate how our models can

help different types of users, we simulate each observation with three different critique selection strategies [15] as seen in Figure 3:

- (1) **Random:** Users who are new to the domain (e.g., new readers) tend to critique rationales at random;
- (2) **Pop:** Users with some domain knowledge and general preferences can correct more common rationales; and
- (3) **Diff:** Knowledgeable users with *specific* preferences will try to correct the *weakest* rationale.

In all settings, a user may only see any single item once and critique each rationale once per session.

In the real world, users may also discover their preferences throughout the course of a conversation, rather than starting with full knowledge of their target item (i.e., all aspects that apply to the target item). When simulating users, we assume that they will give feedback consistent with their knowledge level (Random/Pop/Diff) and fully consistent with their target item. In future work, we aim to model user behavior in this setting with incomplete information, where critiques throughout the course of a session may be inconsistent with one another or the final accepted item.

In light of these limitations of simulated multi-step recommendation sessions, we additionally conduct a user study in Section 6.2 to evaluate how effective our bot-play framework is for cold-start recommendation with real humans.

### 4.3 Candidate Algorithms

Our method can apply to any base recommender system; here, we train bot-play models based on BPR and PLRec—**BPR-Bot** and **PLRec-Bot**, respectively. BPR-Bot is lightweight and much faster, while PLRec-Bot is similar in size to SOTA baseline models for conversational critiquing. We demonstrate in Section 5.1 that our framework is indeed model-agnostic, and that BPR-Bot and PLRec-Bot both out-perform baselines.

*Baseline methods.* We assess linear critiquing baselines that co-embed critique and user representations [19], where  $f_{\text{crit}}$  is a weighted sum of the user preference vector  $\vec{y}_u$  and embeddings for each critiqued rationale. **UAC** uniformly averages  $\vec{y}_u$  and all critiqued rationale embeddings. **BAC** averages  $\vec{y}_u$  with the *average* of critiqued rationale embeddings. **LLC-Score** learns weights by maximizing the rating margin between items containing critiqued rationales and those without. Instead of directly optimizing the scoring margin, **LLC-Rank** [15] minimizes the number of ranking violations. These models cannot generate justifications; we binarize the historical rationale frequency vector for the item ( $\mathbf{K}_{u,i}^I$ ) as a justification at each turn. We also compare against a SOTA interactive recommender, **CE-VAE** [20], which learns a VAE with a bidirectional mapping between critique vectors and the user latent preference space.

## 5 EXPERIMENTS

In this section, we evaluate our bot-play models to answer the following questions:

- **RQ 1:** Can our framework enable multi-step critiquing?
- **RQ 2:** Does *bot-play* specifically improve multi-step critiquing ability?
- **RQ 3:** Can our models generate useful and accurate rationales?

### 5.1 RQ1: Can Our Framework Enable Multi-step Critiquing?

Following standard practice [1, 15, 19], we measure multi-step critiquing performance via average success rate (SR@N)—the percentage of sessions where the target item reaches rank threshold

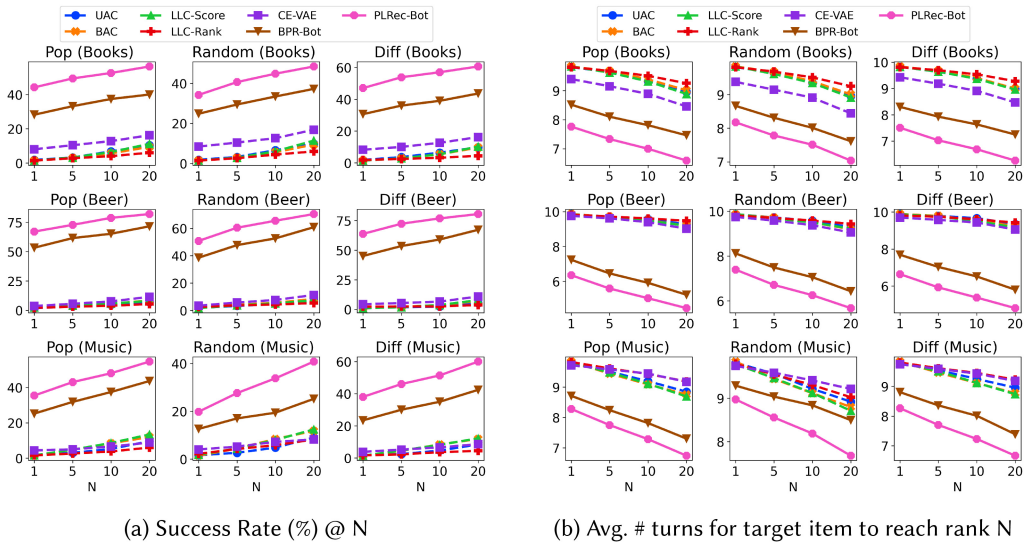


Fig. 4. User simulation evaluation of our models—BPR-Bot (brown triangle) and PLRec-Bot (pink circle)—compared to linear critiquing and variational baselines for conversational recommendation (dashed lines). Models trained with our bot-play framework succeed at significantly higher rates (a) and surface desired items significantly faster (b) than all baselines.

$N$ —and the average session length for the target to reach a rank threshold (Figure 4). We find that both of our candidate models (**BPR-Bot** and **PLRec-Bot**) out-perform all baselines. As our bot-play fine-tuning seeker model picks critiques by popularity, we expect our models to perform best in the Pop setting. However, BPR-Bot and PLRec-Bot succeed faster and at a higher rate than baselines in *all* user settings, including random critiquing with no prior on user behavior.

Linear critiquing models (UAC, BAC, LLC-Score/Rank) perform poorly on multi-step critiquing compared to models that can generate justifications, especially when trying to find the goal item outright ( $N=1$ ). This suggests that personalized justifications help users choose more effective rationales to critique. Despite out-performing linear critiquing models, CE-VAE performs worse across all settings compared to models trained in our bot-play framework. This suggests that our models generate personalized justifications that are more helpful for narrowing down a user’s preferences compared to CE-VAE. In Section 5.3, we further investigate the usefulness and accuracy of our rationales.

For **PLRec-Bot**, our base recommender system is initialized with the same base model used in linear critiquing models (UAC, BAC, LLC-Score/Rank). However, we observe an order of magnitude improvement in success rate across all rank thresholds  $N$  compared to linear models (and the similarly complex CE-VAE model). This demonstrates that we do not need to solve a linear programming problem for each critiquing step (like LLC-Score/Rank)—fine-tuning a model with our bot-play framework is more effective at teaching conversational agents to incorporate user feedback.

With **BPR-Bot**, we demonstrate that our bot-play framework can also be effectively applied to extremely lightweight and simple base recommender systems. Our base BPR models require an order of magnitude ( $5\times$ – $40\times$ ) fewer parameters than baseline models, representing users and items with only 20 latent dimensions. Nonetheless, by fine-tuning this model with our bot-play framework, we are able to again out-perform baselines by wide margins in all settings. Success

Table 5. Example Dialog where a User Is Looking for the Book “Shadow of Night”

	Model	Recommendation	Rationale	Target Rank
Turn 1	CE-VAE	Dark Lover (Black Dagger Brotherhood, #1)	comic relief, <b>greek</b> , <b>norse</b> , third person	76
	Ours (PLRec-Bot)	The Twilight Saga (Twilight, #1–4)	epic, battle, character development, cliffhanger, <b>fairytale</b>	94
Turn 2	CE-VAE	Lover Awakened (Black Dagger Brotherhood, #3)	<b>norse</b> , comic relief	76 (=0)
	Ours	Bloodlines (Bloodlines, #1)	epic, demon, action, slow, <b>creepy</b>	25 (↓69)
Turn 3	CE-VAE	Lover Eternal (Black Dagger Brotherhood, #2)	<b>comic relief</b>	75 (↓1)
	Ours	<b>Shadow of Night (All Souls Trilogy, #2)</b>	epic, fantasy world, urban, action, classic	0 (↓25)

Our model (PLRec-Bot) uses user critiques to rapidly improve its understanding of user preferences, recommending the target item in three turns. CE-VAE continues to recommend books from the same series, while the target item rank remains roughly the same over the same dialog length. Critiqued aspects displayed in **red**.

with both PLRec-Bot and BPR-Bot showcases the model-agnostic nature of our framework, and in future work, we hope to investigate its benefits with a wider range of base recommender systems.

While we were unable to access to code for and replicate the results for the recent MM-VAE model [1], we note that both of our bot-play models significantly out-perform MM-VAE’s reported success rates: For Beer, we achieve 38%–53% SR@1 with BPR-Bot and 51%–67% SR@1 with PLRec-Bot compared to 5%–6% reported SR@1 for MM-VAE; for Music, we achieve 13%–25% SR@1 with BPR-Bot and 20%–35% SR@1 with PLRec-Bot compared to 3%–8% for MM-VAE.

As shown in Table 5, baseline methods tend to have high confidence in their initial recommendation and continue to recommend similar items despite critiques. In this case, CE-VAE recommends other books in the same series and cannot provide the user with diversified rationales to critique. The model does not leverage the critique feedback to improve its understanding of the user’s preference during this session—the rank of the target item remains roughly similar throughout the first three turns of conversation. Meanwhile, our model (PLRec-Bot) uses critique feedback to significantly improve its user understanding each turn, raising the target rank by a large margin with each piece of feedback and recommending the target item on the third turn. This suggests that our bot-play loss (Section 3.4) successfully enables the model to better incorporate user feedback at each turn.

Overall, our models can better assist users with varying levels of domain knowledge and specific preferences compared to SOTA methods for conversational critiquing. We have thus shown that our bot-play framework enables the training of multi-turn conversational recommenders *without the need for costly supervised dialog transcripts*.

## 5.2 RQ2: Does *bot-play* Specifically Improve Multi-step Critiquing Ability?

We next demonstrate that our bot-play fine-tuning is responsible for gains in multi-step critiquing performance (Figure 5(a)) by comparing BPR-Bot (crosses) and PLRec-Bot (squares) against ablated versions that were trained using the first step of our framework but *not* fine-tuned via bot-play. For clarity, we display only results using the Pop user behavioral model, as we observe the same trends with all three user models.

Bot-play confers a noticeable benefit for both BPR-Bot (100%–300% improvement in success rate for various N) and PLRec-Bot (250%–400% improvement) across domains, with the largest

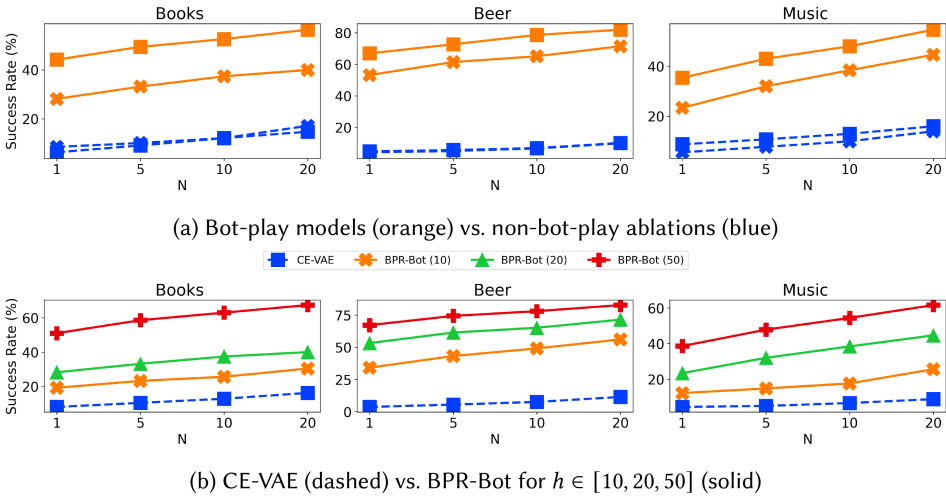


Fig. 5. Success rate @  $N$  (% sessions where target item rank  $\leq N$ ) for ablation settings: (a) Bot-play improves target item ranking across datasets compared to the ablation for PLRec-Bot (squares) and BPR-Bot (crosses). (b) As latent dimension grows ( $h \uparrow$ ), bot-play fine-tuning confers greater benefits. All models, including extremely lightweight  $h = 10$ , out-perform the best baseline model (CE-VAE).

Table 6. Given the Same Feedback about a Recommended Item, Models Trained with Bot Play Improve Their Ranking of the Target Item Significantly Faster than Baseline

Domain	Target Item	Recommendation	Critique	Model	$\Delta$ Target Rank
Beer	La Folie	Punkin Ale	Mocha	PLRec	$\downarrow 20$
				+Bot Play	$\downarrow 68$
Music	TLC - Fanmail	Madonna - Erotica	Rock	PLRec	$\downarrow 2$
				+Bot Play	$\downarrow 62$
Books	Golden Son	Cinder	Mystery	PLRec	$\downarrow 26$
				+Bot Play	$\downarrow 88$

This shows that bot-play helps models adapt their understanding of user preferences from feedback.

improvements observed with the Beer domain. This may be due to relatively dense occurrence of rationales in user reviews, with an average of 7.4 unique rationales expressed in each review (Table 4). In Table 6, we show representative sample turns from each domain where the user gives the same feedback about a recommended item. Models trained with bot-play improve the target rank significantly faster given the same piece of feedback. This points to the effectiveness of bot-play fine-tuning loss  $\mathcal{L}^{\text{expert}}$ : The aspect loss  $\mathcal{L}_A$  encourages the model to return accurate rationales that users care about, and the discount term  $\delta$  helps the model make larger, more accurate changes in its understanding of user preferences given a piece of feedback. This demonstrates that we can effectively train conversational recommender systems using our bot-play framework using domains with user reviews in lieu of crowd-sourced dialog transcripts.

In domains with more sparse coverage of subjective rationales (i.e., Books with 1.8 rationales/review and Music with 2.5 rationales/review), we observe lower improvement when using bot-play—our model may encounter insufficient cases of rare rationales being critiqued. This seems to affect lightweight models (BPR-Bot) much more than more complex base recommender systems (PLRec-Bot). In future work, we will explore adding noise to our user model to ensure that the bot-play process encounters more rare rationales.

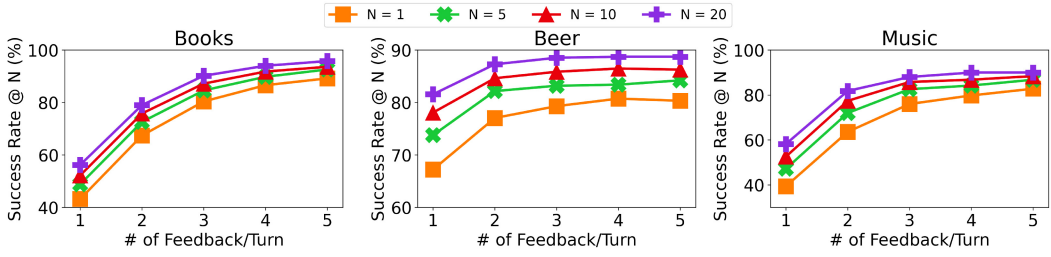


Fig. 6. Success rate @ N for PLRec-Bot with a maximum of 1 to 5 critiques at each turn. Despite our bot-play training using only a single critique per turn, each additional piece of feedback provides additional improvements in success rate, reaching 60%+ success rates at 1 for 2 pieces of feedback.

We next investigate whether our framework is model size-agnostic. We fine-tune BPR models of varying sizes (varying user/item representation dimensionality  $h$  between 10 and 50), with success rates shown in 5(b). We see that regardless of model size, simple recommender systems fine-tuned under our framework out-perform state-of-the-art conversational critiquing methods (CE-VAE). Models with higher latent dimensionality ( $h = 10 \rightarrow 20 \rightarrow 50$ ) benefit more from bot-play, suggesting that our method learns to effectively navigate complex preference spaces.

The marginal benefit of increasing latent dimensionality seems to slow for the Beer domain (with the highest density of rationales per review, item, and user), while we continue to observe large benefits from increasing model size in Books and Music. This suggests that our bot-play framework allows large models to more effectively learn to encode user feedback in domains with sparse user feedback.

*Multiple Simultaneous Critiques.* Finally, we consider conversational recommendation with multiple simultaneous critiques. As we observe in our user studies (Section 6), people tend to give multiple pieces of feedback at a given turn, with an average of around 2 critiques. As our bot-play training (Algorithm 1) only simulates a single critique per turn, we investigate whether such a model can handle more realistic behavior.

In Figure 6, we plot the success rate at N for  $N \in [1, 5, 10, 20]$  for different numbers of critiques per turn. Our bot-play successfully allows our model to appropriately react to user behavior with varying degrees of feedback—the marginal value of each additional piece of feedback per turn is fairly high for the second and third pieces of feedback. Indeed, while the success rate at 1 (rate at which our agent returns the goal item exactly within the turn limit) varies between 40%–68% across datasets, adding an additional piece of feedback improves this to 65%–77%. Our models can quickly narrow down the most appropriate candidate items, approaching 90%–100% success rate for  $N = 20$ .

We thus confirm that our method is model-agnostic, as it improves recommendation success rates for both the matrix factorization-based (BPR) and linear (PLRec) recommender systems. Similarly, we have shown that our bot-play method is size-agnostic and is generally applicable to base recommender systems with any latent dimensionality. Finally, we observe that our bot-play fine-tuning allows our model to accommodate multiple simultaneous critiques per turn—suggesting its usefulness in real-world scenarios.

### 5.3 RQ3: Can Our Models Generate Useful and Accurate Rationales?

We next explore whether our model is surfacing appropriate rationales to guide the user and elicit feedback. We evaluate two main criteria with regards to rationales: (1) *usefulness*, or whether the rationales can help the user give effective feedback to more easily find their desired item; and (2)

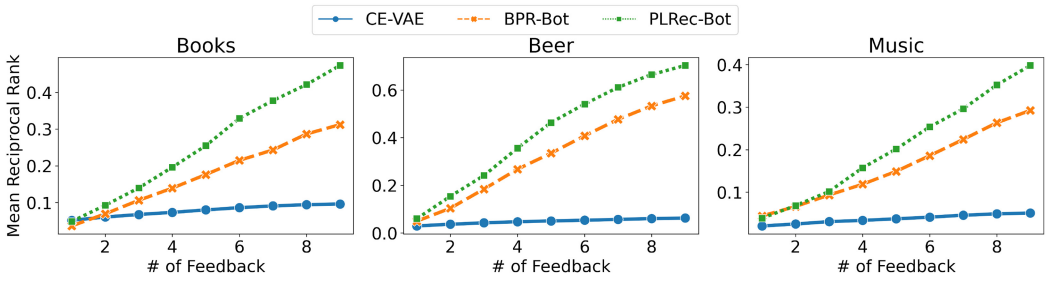


Fig. 7. Mean Reciprocal Rank (MRR) vs. pieces of user feedback received, comparing the best baseline (CE-VAE, blue circles) against BPR-Bot (orange crosses) and PLRec-Bot (green squares). Users are able to give much more useful feedback when presented with rationales for both of our models, improving MRR faster than CE-VAE.

*accuracy*, or whether our model surfaces rationales related to the user’s true preferences in that session.

We note that accuracy and usefulness of rationales must be balanced in a conversational critiquing system. This is because a user’s reviews are necessarily incomplete: the user is unlikely to take the time to express every single one of their opinions about a product—including subtle preferences that may help them decide between very similar items. As a result, the system must both predict the rationales a user would express in their review of the target item *and* the qualities specific to a recommended item that help users distinguish between similar items.

To measure the **usefulness** of our rationales, we measure the **mean reciprocal rank (MRR)** of the target item for each piece of feedback given by the user. This reflects the value of each piece of feedback: We desire a model that can properly incorporate user feedback to more quickly identify the user’s real preference (improve the goal item rank and MRR). In Figure 7, we plot the MRR against pieces of user feedback for PLRec-Bot (squares) and BPR-Bot (crosses) compared to the best baseline conversational critiquing system (CE-VAE). We see that as the conversation progresses, models trained with our bot-play framework can more accurately rank the user’s preferred items compared to CE-VAE.

More importantly, the “slope” of this graph represents the *marginal value* of each piece of feedback. For both PLRec-Bot and BPR-Bot, we observe a significantly higher marginal value of user feedback, suggesting that our rationales are more useful than those surfaced by CE-VAE. We also find that the marginal value of user feedback stays roughly constant for each piece of feedback, showing that our models can effectively refine user preferences even if a user has already provided several pieces of feedback.

We next measure the **accuracy** of rationales surfaced by conversational recommender systems. We assume that when writing a review, the user faithfully expresses their true preferences via the rationales contained in the review. As such, for each session where a user  $u$  tries to find item  $i_2$  we take as ground truth the rationales extracted from the user’s true review of the target item  $k_{u,i}$ . In Figure 8, we plot the average F1 score of the rationales presented to the user (compared to the ground truth session preferences) at each turn of conversation for BPR-Bot, PLRec-Bot, and the CE-VAE baseline.

Across all datasets, we find that bot-play models provide more accurate justifications compared to CE-VAE. Furthermore, unlike CE-VAE, the accuracy of our justifications tends to increase as the session progresses. This suggests that when receiving feedback from the user, our models can improve their understanding of the user’s preference in that particular session. This may help reinforce the user’s trust of our system, as it provides the sense of an agent who “learns” the user’s preferences during a conversation.

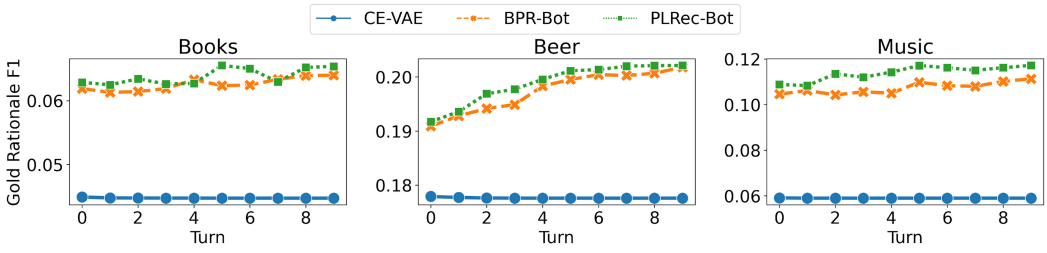


Fig. 8. F1 score of rationales surfaced by conversational recommender systems compared to the user’s ground truth rationales of the target item. When comparing CE-VAE (blue circles) to models trained with our bot-play framework—BPR-Bot (orange crosses) and PLRec-Bot (green squares)—our models more accurately infer the user’s session preferences and can improve their accuracy with each piece of user feedback.

We note that models are able to better refine rationales in domains with more dense expression of subjective rationales per user review (Table 4). In particular, the book domain contains both the most users and the lowest density of rationales per review, and our models see the least improvement in F1 score over a conversation. However, this may reflect how our models suggest more rationales than users typically reveal to help users better evaluate suggested novels.

## 6 HUMAN STUDY

### 6.1 Human Evaluation

Following Li et al. [16], we conduct a comparative evaluation of 100 simulated user sessions on four criteria: which agent seems more useful, informative, knowledgeable, and adaptive. We compare each bot-play model (BPR-Bot and PLRec-Bot) against an ablative version (with no bot-play) and the best baseline (CE-VAE).

Each sample is evaluated by three annotators, with all annotators recruited via the **Amazon Mechanical Turk (MTurk)** platform. We used crowd-workers with a historical 99% acceptance rate on their work to ensure quality, and crowd-workers were paid in excess of federal minimum wage in the United States given the average time taken to complete an evaluation. The datasets we used have been processed to remove offensive words and phrases before presenting them to human evaluators and users. We do not collect biometrics or **personally identifiable information (PII)** from human evaluators, and they were informed that this study was part of an academic research project and may be published. All crowd-workers were native English-speaking individuals.

For the human evaluation, we presented two user simulation traces from different models (e.g., PLRec-Bot and CE-VAE) in a random order, then asked users to decide which of the two models is more useful, which is more informative, which is more knowledgeable, and which is more adaptive. Each user simulation trace is for the same user and target item to be able to fairly compare models. Annotators were not told which item was the target. However, this scenario mimics a scenario where the annotators, as a third party, observe a conversation with a simulated user and a recommender system, starting from an input query to the final recommendation. While this is not exactly the same as a user directly interacting with the system, similar (passive) evaluation is standard for conversational systems, as it provides enough comparative evidence on the quality of the competing systems [16].

We observe substantial [12] inter-annotator agreement, with Fleiss  $\kappa$  [7] of 0.67, 0.79, 0.73, and 0.60 for the usefulness, informativeness, knowledgeable, and adaptiveness criteria, respectively. Scores are shown in Table 7.

BPR-Bot and PLRec-Bot are judged to be significantly more informative and knowledgeable than ablative models and CE-VAE, showing that our models can accurately and convincingly *explain*

Table 7. Session-level Human Evaluation via ACUTE-EVAL

<b>BPR-Bot vs.</b>	Useful		Inform.		Know.		Adaptive	
	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Ablation (BPR)	<b>78*</b>	10	<b>73*</b>	11	<b>68*</b>	15	<b>85*</b>	5
CE-VAE	<b>83*</b>	9	<b>74*</b>	10	<b>63*</b>	16	<b>81*</b>	8

<b>PLRec-Bot vs.</b>	Useful		Inform.		Know.		Adaptive	
	Win	Lose	Win	Lose	Win	Lose	Win	Lose
Ablation (PLRec)	<b>86*</b>	5	<b>78*</b>	7	<b>74*</b>	8	<b>81*</b>	9
CE-VAE	<b>87*</b>	7	<b>79*</b>	11	<b>77*</b>	12	<b>83*</b>	10

Users were asked which model was more Useful, (Inform)ative, (Know)ledgeable, and Adaptive when comparing bot-play models against CE-VAE and an ablative baseline with no bot-play fine-tuning. Results are shown for BPR-Bot (left) and PLRec-Bot (right). W/L percentages are reported, while ties are not. All results are statistically significant with  $p < 0.05$  via binomial test.

each suggestion. This supports our findings from user simulations in Section 5.3. In particular, wins in informativeness and knowledgeability reflect how rationales surfaced by our models accurately describe the subjective opinions of users regarding the suggested item. If users believe a conversational agent can both accurately describe an item and reflect their personal opinions, then they are more likely to trust the system and continue to interact with the agent in a meaningful way [32].

The usefulness and adaptiveness criteria capture how models help the user achieve their end goal (i.e., finding the most relevant item in as few turns as possible). Bot-play models are judged to be more useful than alternatives and follow critiques more consistently when adapting recommendations. This again suggests that users (1) trust our models' rationales for recommendations and (2) can meaningfully interact with our model to achieve their end goal.

Our framework allows us to train conversational agents that are useful and engaging for human users: evaluators overwhelmingly judged the models trained via bot-play to be more useful, informative, knowledgeable, and adaptive compared to CE-VAE and ablated variants.

## 6.2 Cold-start User Study

We conduct a user study using the Books dataset to evaluate if our model is a useful real-time conversational recommender. In particular, we wish to see if models trained with bot-play using existing user reviews could effectively make use of feedback from new users (cold-start). We recruited 64 native English speakers from universities across the United States, randomly assigning half to interact with **BPR-Bot** and half to interact with the ablation (no bot-play). No target item was provided—users were instructed to interact with the model for 10 turns or until the model provided a satisfactory recommendation.

Like the human evaluation, we do not collect biometrics or PII from study participants, and they were informed that this study was part of an academic research project and may be published. Users in both our user evaluation and user study were permitted to exit the task at any time and have their interactions wiped from the project.

We initialize each session with the mean of all learned user embeddings to provide the same initial set of suggestions for each new user. At each turn, the user sees the three top-ranked items with justifications (rationales) and can critique multiple rationales. On average, users critiqued two rationales per turn—this suggests that when training interactive agents, we can assume multiple critiques at each turn. In future work, we aim to study whether users in warm-start and cold-start situations give differing amounts of feedback at each turn of conversation.

Table 8. Cold-start User Study Results

	Avg. Feedback	Useful	Informative	Adaptive	Would Use Again
Ablation (No Bot Play)	1.77 ± 0.08	0.67 ± 0.24	0.75 ± 0.21	0.64 ± 0.27	41%
Our Method	2.05 ± 0.13	<b>0.79 ± 0.24*</b>	<b>0.88 ± 0.18</b>	<b>0.78 ± 0.23*</b>	<b>69%*</b>

On a per-turn basis, users found our bot-play model to be significantly ( $p < 0.01$ ) more useful, informative, and adaptive compared to the baseline. On a session basis, significantly more users ( $p < 0.01$ ) would use the bot-play model “often” or “always” to receive book recommendations compared to the baseline.

Table 9. Mean and Standard Error of Wall-clock Time (ms) per Turn of Critiquing for Linear (LLC-Score) and Variational (CE-VAE) Baselines vs. Our Models (BPR-Bot, BPR-PLRec)

	LLC-Score	CE-VAE	BPR-Bot	PLRec-Bot
Books	40.64 ± 20.46	4.61 ± 1.16	2.70 ± 3.95	48.84 ± 14.08
Beer	15.94 ± 14.52	3.26 ± 1.18	2.54 ± 2.36	49.43 ± 14.81
Music	42.21 ± 21.04	3.36 ± 1.37	2.25 ± 0.62	6.80 ± 7.53

At each turn, we again follow Li et al. [16] to ask users if the generated explanations are *informative*, *useful* in helping to make a decision, and whether our system correctly *adapted* its suggestions in response to the user’s feedback. We provide four options for each question: no, weak-no, weak-yes, and yes. We then map these values to a score between 0 and 1 [9, 21], with normalized scores for each question shown in Table 8. **BPR-Bot** significantly out-scores the ablation in all three metrics ( $p < 0.01$ ), showing that fine-tuning via our bot-play framework instills a stronger ability to respond to critiques and provides meaningful explanations—even for new users.

At the end of a session, we additionally ask the user how frequently (if at all) they would choose to engage with our interactive agent in their daily life. Users preferred BPR-Bot by significant margins—69% indicated they would “often” or “always” use BPR-Bot to find books compared to 41% for the ablation. We are encouraged that over two-thirds of users would regularly use our system, which confirms that our critiquing approach to conversational recommendation reflects a realistic and appealing human interaction paradigm.

## 7 DISCUSSION

### 7.1 Computational Complexity

In Table 9, we report the mean and standard error of time taken per turn for LLC-Score, CE-VAE, BPR-Bot, and PLRec-Bot. As baseline code does not leverage the GPU, we also critique with PLRec-Bot and BPR-Bot on the CPU only. We observe LLC-Score and PLRec-Bot to be an order of magnitude slower per critiquing cycle compared to CE-VAE and BPR-Bot. BPR-Bot shows acceptable latency for real-world applications (sub-10 ms), and we observe empirically in our cold-start user study that we can host BPR-Bot as a real-time recommendation service. Time trials were conducted with batch size of 1; production throughput can be improved further with parallel processing. Each model executes using a different framework (numpy for LLC-Score, Tensorflow for CE-VAE, and Pytorch for PLRec-Bot/BPR-Bot), which may contribute to differences in inference speed.

### 7.2 Ethics and Broader Impact

As we aim to train conversational multi-turn recommendation agents, the primary risks of our approach lie in taking too long to present a user with good items or suggesting items they dislike. This risk is not unique to our approach and to some extent depends on the target domain (e.g., users may hold stronger opinions about food than they do computer hardware). One risk

surface is the natural language rationales (and product names) that we surface to users as part of our recommend-and-justify approach. These could theoretically contain offensive or uncomfortable phrasing, but this risk can be minimized by a human-in-the-loop review of the rationale extraction process (e.g., blacklisting certain extracted rationales) or by applying toxic text detection to filter user reviews as a pre-processing step.

## 8 CONCLUSION

In this work, we develop conversational recommenders that can engage with users over multiple turns, providing rationales for suggestions and incorporating user feedback. We present a model-agnostic framework to train conversational agents in this modality via self-supervised bot-play in any domain using only review data. We use two popular underlying recommender systems to train the **BPR-Bot** and **PLRec-Bot** agents using our framework, showing quantitatively on three datasets that our models (1) offer superior multi-turn recommendation performance compared to current SOTA methods; (2) provide more useful and informative rationales for each recommended item compared to current SOTA methods; and (3) can effectively refine suggestions in real-time, as shown in user studies. We further show that our bot-play framework confers its benefits for models with different underlying architectures and levels of complexity. In future work, we aim to adapt our framework to free-form natural language critiques, allowing users to more flexibly express feedback.

## REFERENCES

- [1] Diego Antognini and Boi Faltings. 2021. Fast multi-step critiquing for VAE-based recommender systems. In *RecSys*. ACM, 209–219. DOI: <https://doi.org/10.1145/3460231.3474249>
- [2] Diego Antognini, Claudiu Musat, and Boi Faltings. 2021. Interacting with explanations through critiquing. In *IJCAI*, Zhi-Hua Zhou (Ed.). ijcai.org, 515–521. DOI: <https://doi.org/10.24963/ijcai.2021/72>
- [3] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1996. Knowledge-based navigation of complex information spaces. In *AAAI*. AAAI Press/The MIT Press, 462–468.
- [4] Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1997. The FindMe approach to assisted browsing. *IEEE Expert* 12, 4 (1997), 32–40. DOI: <https://doi.org/10.1109/64.608186>
- [5] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *KDD*. ACM, 815–824. DOI: <https://doi.org/10.1145/2939672.2939746>
- [6] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies: Why and how. In *IUI*. ACM, 193–200. DOI: <https://doi.org/10.1145/169891.169968>
- [7] Joseph L. Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educat. Psychol. Measur.* 33, 3 (1973), 613–619.
- [8] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul A. Crook, Y-Lan Boureau, and Jason Weston. 2019. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *EMNLP-IJCNLP*. Association for Computational Linguistics, 1951–1961. DOI: <https://doi.org/10.18653/v1/D19-1203>
- [9] Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-ViL: A dataset and benchmark for natural language explanations in vision-language tasks. In *ICCV*. IEEE, 1224–1234. DOI: <https://doi.org/10.1109/ICCV48922.2021.00128>
- [10] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *ICLR*. ICLR, 1–9. Retrieved from: <http://arxiv.org/abs/1312.6114>
- [11] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37. DOI: <https://doi.org/10.1109/MC.2009.263>
- [12] J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33, 1 (1977), 159–174. Retrieved from: <http://www.jstor.org/stable/2529310>
- [13] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *WSDM*. ACM, 304–312. DOI: <https://doi.org/10.1145/3336191.3371769>
- [14] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *KDD*. ACM, 2073–2083. DOI: <https://doi.org/10.1145/3394486.3403258>

- [15] Hanze Li, Scott Sanner, Kai Luo, and Ga Wu. 2020. A ranking optimization approach to latent linear critiquing for conversational recommender systems. In *RecSys*. ACM, 13–22. DOI : <https://doi.org/10.1145/3383313.3412240>
- [16] Margaret Li, Jason Weston, and Stephen Roller. 2019. ACUTE-EVAL: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *CoRR* abs/1909.03087 (2019), 1–8.
- [17] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NeurIPS*. NeurIPS, 9748–9758.
- [18] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *ICLR*. ICLR, 1–9.
- [19] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent linear critiquing for conversational recommender systems. In *WWW*. ACM/IW3C2, 2535–2541. DOI : <https://doi.org/10.1145/3366423.3380003>
- [20] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep critiquing for VAE-based recommender systems. In *SIGIR*. ACM, 1269–1278. DOI : <https://doi.org/10.1145/3397271.3401091>
- [21] Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian J. McAuley. 2022. Knowledge-grounded self-rationalization via extractive and natural language explanations. In *ICML*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 14786–14801. Retrieved from: <https://proceedings.mlr.press/v162/majumder22a.html>
- [22] Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *ICDM*. IEEE Computer Society, 1020–1025. DOI : <https://doi.org/10.1109/ICDM.2012.110>
- [23] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. ACM, 43–52. DOI : <https://doi.org/10.1145/2766462.2767755>
- [24] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*. Association for Computational Linguistics, 188–197. DOI : <https://doi.org/10.18653/v1/D19-1018>
- [25] Denis Parra and Peter Brusilovsky. 2015. User-controllable personalization: A case study with SetFusion. *Int. J. Hum. Comput. Stud.* 78 (2015), 43–67. DOI : <https://doi.org/10.1016/j.ijhcs.2015.01.007>
- [26] Lingyun Qiu and Izak Benbasat. 2009. Evaluating anthropomorphic product recommendation agents: A social relationship perspective to designing information systems. *J. Manag. Inf. Syst.* 25, 4 (2009), 145–182.
- [27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI Press, 452–461.
- [28] Suvash Sedhain, Hung Hai Bui, Jaya Kawale, Nikos Vlassis, Branislav Kveton, Aditya Krishna Menon, Trung Bui, and Scott Sanner. 2016. Practical linear models for large-scale one-class collaborative filtering. In *IJCAI*. IJCAI/AAAI Press, 3854–3860. Retrieved from: <http://www.ijcai.org/Abstract/16/542>
- [29] Tianshu Shen, Zheda Mai, Ga Wu, and Scott Sanner. 2022. Distributional contrastive embedding for clarification-based conversational critiquing. In *WWW*. Retrieved from: <https://api.semanticscholar.org/CorpusID:248367589>
- [30] Rashmi R. Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI*. ACM, 830–831. DOI : <https://doi.org/10.1145/506443.506619>
- [31] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2009. MoviExplain: A recommender system with explanations. In *RecSys*. ACM, 317–320. DOI : <https://doi.org/10.1145/1639714.1639777>
- [32] Nava Tintarev and Judith Masthoff. 2011. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*. Springer, New York, 479–510.
- [33] Amos Tversky and Itamar Simonson. 1993. Context-dependent preferences. *Manag. Sci.* 39, 10 (1993), 1179–1189.
- [34] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: Explaining recommendations using tags. In *IUI*. ACM, 47–56. DOI : <https://doi.org/10.1145/1502650.1502661>
- [35] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *RecSys*. ACM, 86–94. DOI : <https://doi.org/10.1145/3240323.3240369>
- [36] Pontus Wärnestål. 2005. Modeling a dialogue strategy for personalized movie recommendations. In *Beyond Personalization Workshop*. ACM, 77–82.
- [37] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language-based critiquing for recommender systems. In *RecSys*. ACM, 137–145. DOI : <https://doi.org/10.1145/3298689.3347009>
- [38] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *CIKM*. ACM, 177–186. DOI : <https://doi.org/10.1145/3269206.3271776>
- [39] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*. ACM, 83–92. DOI : <https://doi.org/10.1145/2600428.2609579>
- [40] 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*. ACM, 1006–1014.

Received 15 January 2023; revised 26 February 2024; accepted 27 April 2024