Self-Supervised Bot Play for Transcript-Free Conversational Recommendation with Rationales

Shuyang Li
shl008@ucsd.edu
UC San Diego
San Diego, California, USA

Bodhisattwa Prasad Majumder
bmajumde@ucsd.edu
UC San Diego
San Diego, California, USA

Julian McAuley
jmcauley@ucsd.edu
UC San Diego
San Diego, California, USA

ABSTRACT

Conversational recommender systems offer 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 recommenders 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**.

KEYWORDS

Conversational Recommendation, Critiquing

ACM Reference Format:
Shuyang Li, Bodhisattwa Prasad Majumder, and Julian McAuley. 2022. Self-Supervised Bot Play for Transcript-Free Conversational Recommendation with Rationales. In *Sixteenth ACM Conference on Recommender Systems (RecSys '22), September 18–23, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3523227.3546783

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 interact with agents and suggestions, increasing their willingness to trust and accept recommendations [24]. 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 [32]. 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 [36]. 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 [33] 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.

Our framework is model-agnostic—we apply our method to two different underlying recommendation architectures [25, 26] 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 recommender 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 recommender 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

RecSys '22, September 18–23, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9278-5/22/09.
https://doi.org/10.1145/3523227.3546783

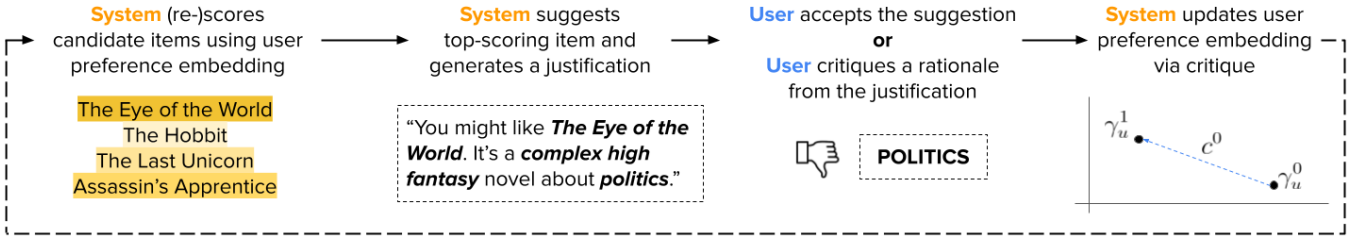


Figure 1: In our conversational recommendation workflow, the system scores candidates and generates a justification for the top item. If the user critiques a rationale, the system uses the critique to update the latent user representation.

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 [27]. Some early recommender systems presented the same attributes of suggested items to all users [30]. Another line of work attempts to generate natural language explanations of recommendations. McAuley et al. [21] mine key attributes from textual reviews via topic extraction. These attributes can be expanded into explanatory sentences via template-filling [35] or recurrent language models [23]. 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 [29]. Early critiquing methods treated user feedback as hard constraints to shrink the search space [3]. Wu et al. [33] 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, 34]. 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_{rec} that ranks items based on their suitability for a user;
(2) a justification module M_{just} that predicts rationales for a given recommendation; and
(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 γ_u via the critique to bring it closer to the target item.

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) [25] is a matrix factorization recommender system that aim to decompose the interaction matrix $\mathbf{R} \in \mathbb{R}^{|U|\times|I|}$ 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 embeddings: $\hat{x}_{u,i} = \langle \gamma_u^{\text{MF}}, \gamma_i^{\text{MF}} \rangle$. 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

Table 1: Critiquing systems (top) are not equipped for multi-turn interactions (*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.

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 [34]	2018	✗	✔	✔
	EAR [13]	2020	✗	✔	✔
	SCPR [14]	2020	✗	✔	✔
Dialog Agents	Li et al. [17]	2018	✗	✔	✗
	Kang et al. [8]	2019	✔	✔	✗
	Zhou et al. [36]	2020	✔	✔	✗
	Ours		✔	✔	✔

Table 2: Notation used in this paper.

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

the observed item:

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

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

Projected Linear Recommendation (PLRec) is an SVD-based method to learn low-rank user/item representations via linear regression [26]. The PLRec objective minimizes:

$$\arg\min_W \sum_u \| r_u - r_u V W^T \|_2^2 + \Omega(W)$$

where V is a fixed matrix obtained by taking a low-rank SVD approximation of $\mathbf{R}$ such that $\mathbf{R} = U\Sigma V^T$, and W is a learned embedding. We obtain an h-dimensional embeddings for users ($\gamma^{\text{MF}}_u = r_u V$) and items ($\gamma^{\text{MF}}_i = W_i$).

3.2 Justification Module

Our justification model (rationale prediction head) consists of a fully connected network with two h-dimensional hidden layers predicting a score $s_{u,i,a}$ for each natural language rationale a. This model takes the sum of user and item embeddings as input. 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:

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

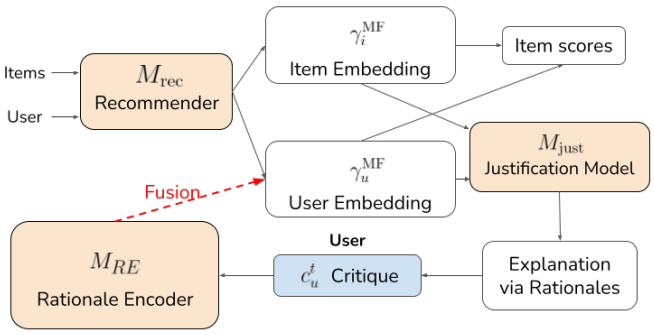
At inference time, we again compute the likelihood for each rationale $p_{u,i,a} = \sigma(s_{u,i,a})$ and sample from the Bernoulli distribution with $p_{u,i,a}$ to determine which rationales a appear in the justification.

3.3 Critiquing Function

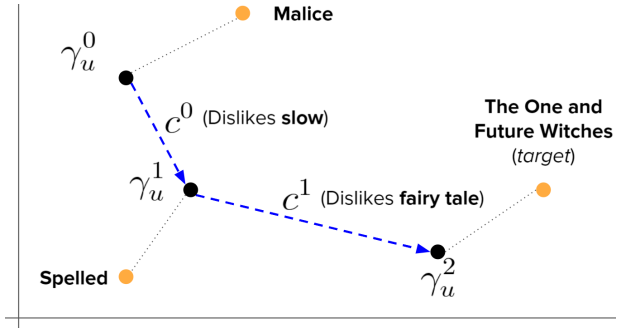
We posit that the user's latent representation is partially explained by their written reviews. We thus learn a rationale encoder M_{RE}—a linear projection from the rationale space to the user preference space: $M_{\text{RE}}(c^t_u) = W^T c^t_u + b$, where $c^t_u \in \mathbb{Z}^{|K|}$ is the critique vector representing the strength of a user's preference for each rationale. We fuse this rationale encoding with the latent user embedding from M_{rec} to form the final user preference vector:

$$\gamma_u = f_{\text{crit}}(\gamma^{\text{MF}}_u, M_{\text{RE}}(c^t_u))$$

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: $c^t_u = \mathbf{k}^U_u$.



(a) Model Architecture



(b) Latent Critiquing Process

Figure 2: (a) Given a user, items, and rationale critique vector, our model encodes the critique $M_{\text{RE}}(c_u^t)$ and fuses it with the user embedding γ_u^{MF} via critiquing function f_{crit}. The fused user (γ_u) and item (γ_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").

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:

$$\hat{x}_{u,i}^{t+1} = M_{\text{rec}}(\hat{\gamma}_u^{t+1}, i)$$
$$\hat{k}_{u,i}^{t+1} = M_{\text{just}}(\hat{\gamma}_u^{t+1}, i)$$
$$\hat{\gamma}_u^{t+1} \leftarrow f_{\text{crit}}(\hat{\gamma}_u^t, c_u^t)$$

Effectively, a user critique modifies our prior for the user's preferences; we then re-rank the items presented to the user.

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

$$c_u^t = c_u^{t-1} - \max(\mathbf{k}_u^U, 1) \odot m_u^t; \quad c_u^0 = \mathbf{k}_u^U$$

where $\odot$ is element-wise multiplication. Here the critique should match the strength of a user's previous opinion of the rationale $\mathbf{k}_u^U$. Even if a user has not mentioned a rationale in their previous reviews, the max ensures a non-zero effect from each critique.

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:

$$\hat{x}_{u,i} = \langle \gamma_u^{\text{MF}} + M_{\text{RE}}(\mathbf{k}_u^U), \gamma_i \rangle$$

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 λ_{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 M_{rec}, M_{just}, and M_{RE}. We optimize M_{RE} via the linear regression:

$$\arg\min_{W,b} \sum_u \| \gamma_u^{\text{MF}} - M_{\text{RE}}(\mathbf{k}_u^U) \|_2^2 + \Omega(W)$$

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

Learning to Critique via Bot Play. We propose a framework for critiquing via bot play that simulates user sessions when provided just a set of user reviews. We first pre-train our expert model (recommender, justifier, and rationale encoder). We use a rule-based seeker with 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. 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 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 δ 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 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

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

Recommender and Justifier $M_{\text{rec}}, M_{\text{just}}$;
Critique fusion function f_{crit};
Seeker model M_{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 γ_u^1 from M_{rec}, $\mathcal{L} = 0$;
    for turn $t \in range(1, T)$ **do** > Simulate up to T turns of user feedback
      $\hat{x}_{u,i}^t = M_{\text{rec}}(\gamma_u^t, i) \; \forall \; i \in I$;
      $\mathcal{L} \leftarrow \mathcal{L} + \delta^t \cdot (\mathcal{L}_{\text{CE}}(g, \hat{x}_{u,i}^t) + \frac{1}{2}\mathcal{L}_{\text{A}})$;
      $\hat{i}^t = \arg\max_i \hat{x}_{u,i}^t$;
      if $\hat{i}^t = g$ **then** > Terminate session if goal item recommended.
        break with success
      $\hat{k}_{u,\hat{i}^t} = M_{\text{just}}(\gamma_u^t, \gamma_{\hat{i}^t})$; > Generate justification for suggested item
      simulate user critique using M_{seeker}: c_u^t;
      $\gamma_u^{t+1} \leftarrow f_{\text{crit}}(\gamma_u^t, c_u^t)$; > Update user latent representation
return fine-tuned agent

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 2.2GHz 40-core CPU, 132GB 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 included in supplementary material. 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) [31], BeerAdvocate (Beer) [21], and Amazon CDs & Vinyl (Music) [22]. 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 dataset 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. [33] to extract subjective rationales (Table 3) from user reviews:

(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 qualities ranging from taste for beers (e.g. citrus) and emotions for music (e.g. soulful) to perceived character qualities in books (e.g. strong female). 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.

4.2 Multi-Step Critiquing

Following prior work on critiquing [15, 19], we simulate multi-step recommendation sessions to assess model performance. 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. 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.

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.

We assess linear critiquing baselines that co-embed critique and user representations [19], where f_{crit} is a weighted sum of the user preference vector γ_u and embeddings for each critiqued rationale. **UAC** uniformly averages γ_u and all critiqued rationale embeddings.

Table 3: 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	Unique R	Sample Subjective R	R/User	R/Item	R/Review
Books	13,889	7,649	654,975	75	Realistic, Strong Female, Gripping	25.0	27.0	1.77
Beer	6,369	4,000	935,524	75	Smoky, Citrus, Nutty, Bitter, Metallic	54.6	60.2	7.39
Music	5,635	4,352	119,081	80	Techno, Reggae, Catchy, Soft, Soulful	16.5	20.0	2.54

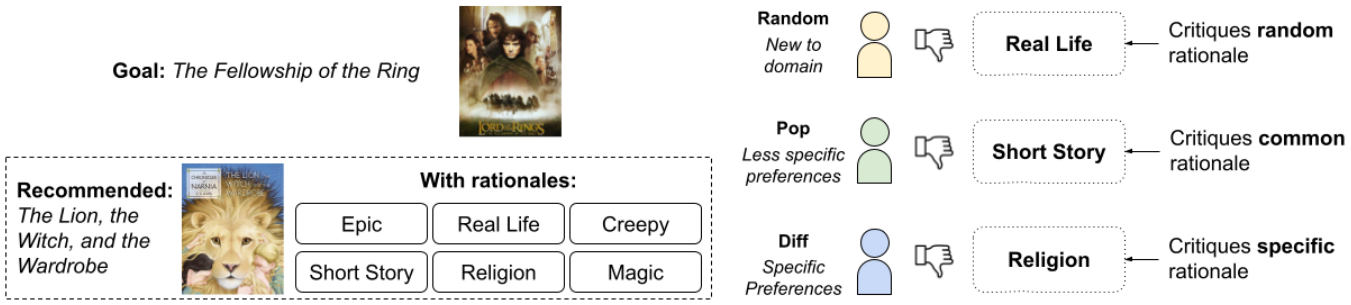


Figure 3: Example of user behaviors after receiving a book recommendation with rationales. A new reader may *randomly* select a rationale to critique. Readers with less specific preferences may critique common / *popular* rationales. A knowledgeable reader with specific preferences will critique a specific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

BAC averages γ_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 ($\mathbf{k}^I_{u,\hat{i}_t}$) 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?

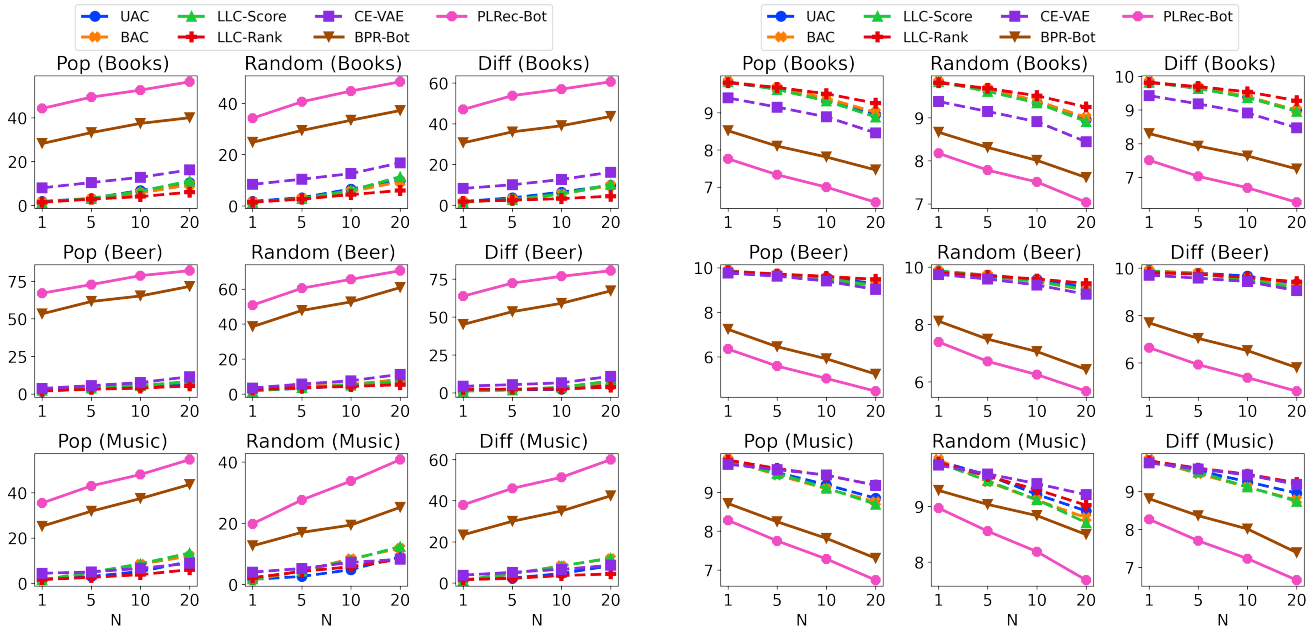
5.1 RQ1: Can our framework enable multi-step critiquing?

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 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 (5x-40x) 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 with both PLRec-Bot and BPR-Bot showcases the model-agnostic nature of our framework, and in future



(a) Success Rate (%) @ N

(b) Avg. # turns for target item to reach rank N

Figure 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.

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.

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?

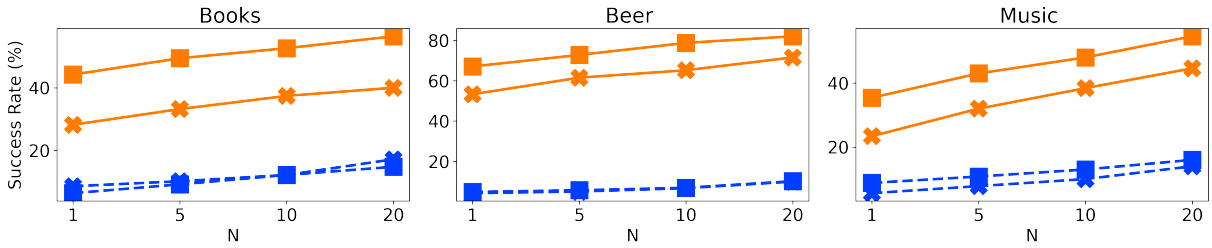
We next demonstrate that our bot-play fine-tuning is responsible for gains in multi-step critiquing performance (Figure 5a) by comparing BPR-Bot (crosses) and PLRec-Bot (squares) against ablated versions that were trained using the first step of our framework but *not* fine-tuned via bot-play. For clarity, 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 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 3). 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.

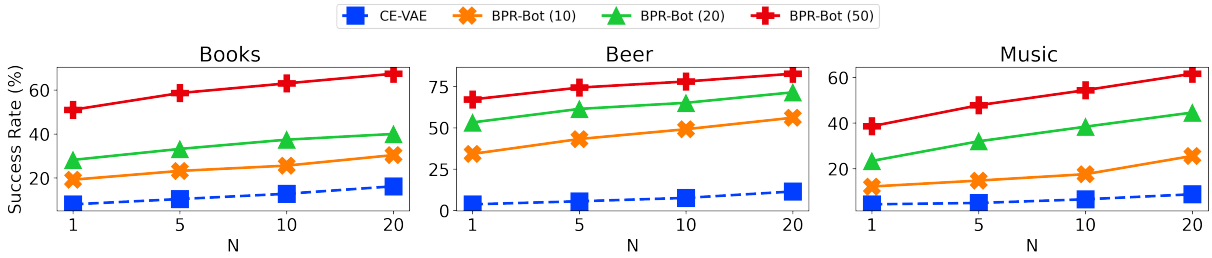
In domains with more sparse coverage of subjective rationales (i.e. Books with 1.8 rationales/review and Music with 2.5 rationales/review), we observe lower improvement when using bot-play—our model may encounter insufficient cases of rare rationales being critiqued. This seems to affect lightweight 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.

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 5b. We see that regardless of model size, simple recommender systems fine-tuned under our framework outperform 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



(a) Bot-play models (orange) vs. non-bot-play ablations (blue)



(b) CE-VAE (dashed) vs. BPR-Bot for $h \in [10, 20, 50]$ (solid)

Figure 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).

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.

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.

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) *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 6, 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 PLRect-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, we take as ground truth the rationales extracted from the user's true review of the target item $\mathbf{k}_{u,i}$. In Figure 7, 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.

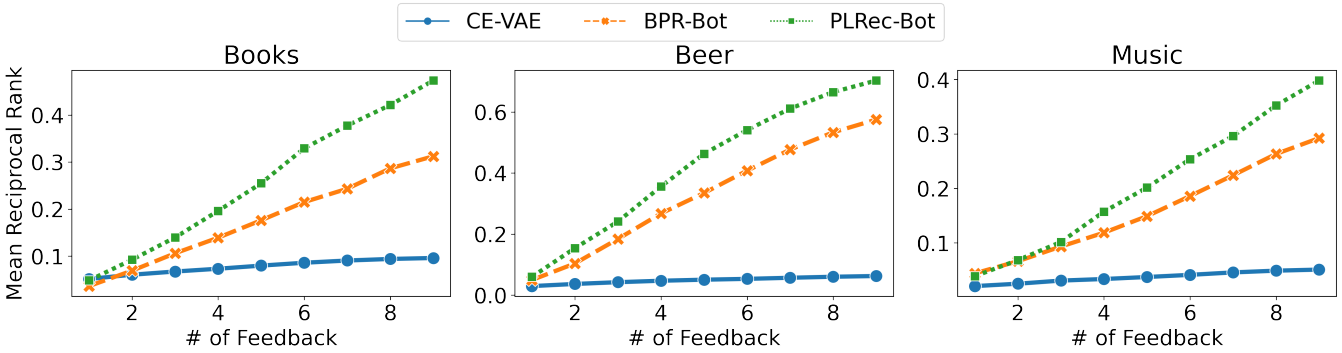


Figure 6: Mean Reciprocal Rank (MRR) vs. pieces of user feedback received, comparing the best baseline (CE-VAE, blue circles) against BPR-Bot (orange crosses) and PLRec-Bot (green squares). Users are able to give much more useful feedback when presented with rationales for both of our models, improving MRR faster than CE-VAE.

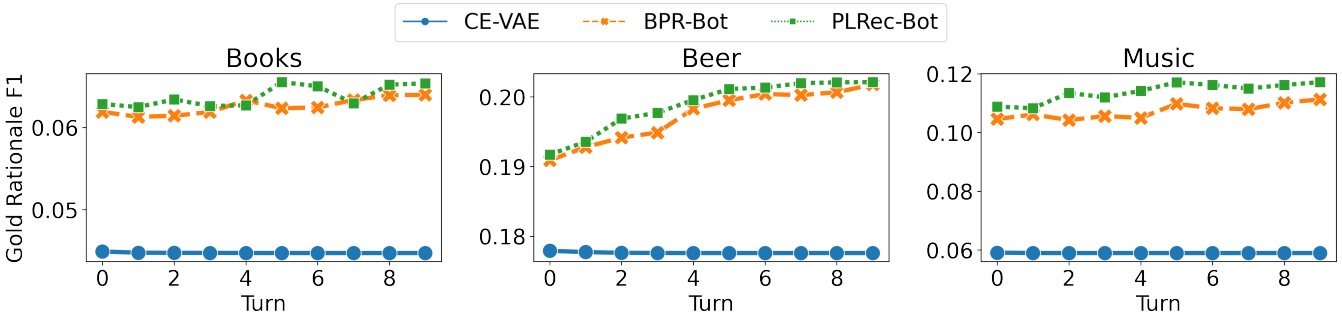


Figure 7: F1 score of rationales surfaced by conversational recommender systems compared to the user's ground truth rationales of the target item. When comparing CE-VAE (blue circles) to models trained with our bot-play framework—BPR-Bot (orange crosses) and PLRec-Bot (green squares)—our models more accurately infer the user's session preferences, and can improve their accuracy with each piece of user feedback.

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.

We note that models are able to better refine rationales in domains with more dense expression of subjective rationales per user review (Table 3). 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. On the other hand, this may reflect how our models suggest more rationales than users typically reveal, in order 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. We observe substantial [12] inter-annotator agreement, with Fleiss κ [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 4.

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* 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, they are more

Table 4: Session-level human evaluation via ACUTE-EVAL. 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 statistically significant with $p < 0.05$ via binomial test.

BPR-Bot vs.	Useful		Inform.		Know.		Adaptive		PLRec-Bot vs.	Useful		Inform.		Know.		Adaptive	
	W	L	W	L	W	L	W	L		W	L	W	L	W	L	W	L
Ablation (BPR)	**78***	10	**73***	11	**68***	15	**85***	5	Ablation (PLRec)	**86***	5	**78***	7	**74***	8	**81***	9
CE-VAE	**83***	9	**74***	10	**63***	16	**81***	8	CE-VAE	**87***	7	**79***	11	**77***	12	**83***	10

Table 5: Cold-start user study results. 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.

	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	**0.79 ± 0.24***	**0.88 ± 0.18**	**0.78 ± 0.23***	**69%***

likely to trust the system and continue to interact with the agent in a meaningful way [28].

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).

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.

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], with normalized scores for each question shown in Table 5. **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 provide 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, and it confirms that our critiquing approach to conversational recommendation reflects a realistic and appealing human interaction paradigm.

7 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, Amsterdam, 209–219. 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, Montreal, 515–521. 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, Portland, 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. https://doi.org/10.1109/64.608186

[5] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *KDD*. ACM, San Francisco, 815–824. 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, Orlando, 193–200. 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. *Educational and psychological measurement* 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, Hong Kong, 1951–1961. 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, Montreal, 1224–1234. https://doi.org/10.1109/ICCV48922.2021.00128

[10] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*. ICLR, Banff, 1–9. 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. 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. 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, Houston, 304–312. 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, Virtual, 2073–2083. 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, Virtual, 13–22. 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. arXiv:1909.03087 http://arxiv.org/abs/1909.03087

[17] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *NeurIPS*. NeurIPS, Montreal, 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, Virtual, 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, Taipei, 2535–2541. 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, Virtual, 1269–1278. https://doi.org/10.1145/3397271.3401091

[21] Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. In *ICDM*. IEEE Computer Society, Brussels, 1020–1025. https://doi.org/10.1109/ICDM.2012.110

[22] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. ACM, Santiago, 43–52. https://doi.org/10.1145/2766462.2767755

[23] 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, Hong Kong, 188–197. https://doi.org/10.18653/v1/D19-1018

[24] 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.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, Montreal, 452–461.

[26] 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, New York, 3854–3860. http://www.ijcai.org/Abstract/16/542

[27] Rashmi R. Sinha and Kirsten Swearingen. 2002. The role of transparency in recommender systems. In *CHI*. ACM, Minneapolis, 830–831. https://doi.org/10.1145/506443.506619

[28] Nava Tintarev and Judith Masthoff. 2011. Designing and Evaluating Explanations for Recommender Systems. In *Recommender Systems Handbook*. Springer, New York, 479–510.

[29] Amos Tversky and Itamar Simonson. 1993. Context-Dependent Preferences. *Management Science* 39, 10 (1993), 1179–1189.

[30] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: explaining recommendations using tags. In *IUI*. ACM, Sanibel Island, 47–56. https://doi.org/10.1145/1502650.1502661

[31] Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *RecSys*. ACM, Vancouver, 86–94. https://doi.org/10.1145/3240323.3240369

[32] Pontus Wärnestål. 2005. Modeling a dialogue strategy for personalized movie recommendations. In *Beyond Personalization Workshop*. ACM, San Diego, 77–82.

[33] Ga Wu, Kai Luo, Scott Sanner, and Harold Soh. 2019. Deep language-based critiquing for recommender systems. In *RecSys*. ACM, Copenhagen, 137–145. https://doi.org/10.1145/3298689.3347009

[34] 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, Torino, 177–186. https://doi.org/10.1145/3269206.3271776

[35] 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, Gold Coast, 83–92. https://doi.org/10.1145/2600428.2609579

[36] 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, Virtual, 1006–1014.