LLM-based Conversational Recommendation Agents with Collaborative Verbalized Experience

Yaochen Zhu
uqp4qh@virginia.edu
University of Virginia
Charlottesville, VA, USA

Harald Steck
hsteck@netflix.com
Netflix Inc.
Los Gatos, CA, USA

Dawen Liang
dliang@netflix.com
Netflix Inc.
Los Gatos, CA, USA

Yinhan He
nee7ne@virginia.edu
University of Virginia
Charlottesville, VA, USA

Nathan Kallus
nkallus@netflix.com
Netflix Inc. & Cornell University
New York, NY, USA

Jundong Li
jundong@virginia.edu
University of Virginia
Charlottesville, VA, USA

Abstract

Large language models (LLM) have demonstrated impressive zero-shot capabilities in conversational recommender systems (CRS). However, effectively utilizing historical conversations remains a significant challenge. Current approaches either retrieve few-shot examples or extract global rules to augment the prompt for LLM-based CRSs, which fail to capture the implicit and preference-oriented knowledge. To address the above challenge, we propose LLM-based **C**onversational **R**ecommendation **A**gents with Collaborative **V**erbalized **E**xperience (**CRAVE**). CRAVE starts by sampling trajectories of LLM-based CRS agents on historical queries and establishing verbalized experience banks by reflecting the agents' actions on user feedback. Additionally, we introduce a collaborative retriever network finetuned with content-parameterized multinomial likelihood on query-items pairs to retrieve preference-oriented verbal experiences for new queries. Furthermore, we developed a debater-critic agent (DCA) system where each agent maintains an independent collaborative experience bank and works together to enhance the CRS recommendations. We demonstrate that the open-ended debate and critique nature of DCA benefits significantly from the collaborative experience augmentation with CRAVE.

1 Introduction

Conversational recommender systems (CRS) aim to recommend items based on dialogue with users [10]. Compared with traditional recommender systems (RS) that leverage static historical interactions or item content to suggest new items, a CRS allows users the critical freedom to express their preferences in natural language, which attracts more attention in both academia and industry [9].

A CRS requires modeling user preference based on multiple rounds of dialogues between the user and system. Traditional methods train sequential models such as RNNs [4] or transformers [22] on historical conversations with groundtruth items, where external databases (e.g., item/word level knowledge graphs) are often introduced as the prior knowledge [3, 32]. Afterward, more efforts have been devoted to finetuning pretrained language models (PLM), e.g., GPT-2, LLAMA-2, such that prior knowledge gained from large external corpora can be utilized for better item/dialogue understanding [7, 24]. However, these PLMs are comparatively small in scale, where the reasoning ability is still limited. Recently, CRSs with large language models (LLM) with hundreds of billions of parameters, such as GPT-4o [14], have gained more attention. These models encompass substantial knowledge and have unprecedented reasoning ability on user preference based on dialogues, where [9] show that zero-shot LLM substantially outperforms both traditional methods and finetuned PLMs.

Despite the success of LLMs as zero-shot CRSs, utilizing the implicit knowledge in historical conversations and user feedback still remains a great challenge. First, most LLMs are large black-box models, which preclude model finetuning with historical conversations [9]. One naive strategy is to retrieve conversation-feedback pairs as few-shot examples in the prompt [8]. However, the *semantic gap* between the conversation and user preference is substantial, making it difficult (even for the LLMs) to derive generalizable recommendations for new conversations in an in-context manner [5]. To bridge the semantic gap, summarizing implicit knowledge by reflecting recommendations on historical conversations with user feedback, as proposed in verbal reinforcement learning (VRL), [11, 17, 29, 31] appears as a promising strategy. However, unlike typical VRL tasks such as question answering (QA) that assume LLMs can have multiple attempts for each question with external feedback, conversational recommendations for each user query are usually one-time. Consequently, [27] adapted VRL to CRS by summarizing rules via sequentially reflecting on the reasoning and recommendations of an LLM-based CRS for all the training samples, which are shared across all test queries and included as part of the instruction in the prompt. Nevertheless, these global rules often fail to account for the personalized preferences of different user queries, which are critical for CRSs.

Furthermore, all the methods introduced above focus on a single LLM with chain-of-thought (CoT) reasoning [26]. However, this might be inherently limited for the recommendation task, as [25] found that LLMs tend to think convergently (i.e., generate one reasoning and stick stringently to it), while for CRS, it is important to promote *divergent* thoughts on different potential aspects of user preference, as the user queries are usually very vague in CRS (otherwise the users will directly search for the item instead of exploring with the system). Recently, LLM debate has been introduced to address such limitations [2]. However, the tasks that LLM debate focuses on (e.g., QA) usually have clear answers rigorously derived with logic/math reasoning from the question, whereas the answers for CRS are more vague and personalized, resembling an open-end debate without definite answers. Therefore, it is especially crucial to derive valuable experiences from historical conversations and user feedback to guide each debater to contribute new perspectives on user preferences based on the user query and the reasoning of other debaters. In addition, since there are no definite answers nor multiple attempts for the CRS debate, it would also be challenging to comprehensively evaluate the reasoning and recommendations from different debaters.

To address the above-mentioned multi-faceted challenges, we propose **CRAVE**, i.e., LLM-based **C**onversational **R**ecommendation **A**gents with Collaborative **V**erbalized **E**xperience, aiming to leverage implicit, personalized, and agent-specific experiences reflected from historical conversations and user feedback to augment LLMs for better recommendations. Specifically, CRAVE starts by sampling trajectories of LLM-based CRS agents on historical conversations and establishing verbal experience banks by reflecting the agents' actions on the user feedback. Afterward, a collaborative retriever network finetuned on query-item pairs with multinomial likelihood is introduced to retrieve preference-oriented collaborative verbal experiences for each new query. Furthermore, we develop a debater-critic agent (DCA) system to tackle the convergent thinking issue of chain-of-thought CRS agent, where each agent in DCA maintains an independent collaborative experience bank and works together to enhance the CRS recommendations. We find that the open-end debate and critique nature of the DCA system benefits significantly from collaborative experience augmentation with CRAVE.

2 Problem Formulation

In this section, we define the CRS problem studied in this paper. Let $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items, respectively. A conversation between a user and the system is denoted as $\{(u_t, s_t, \mathcal{I}_t)\}_{t=1}^{T}$, where at the t-th turn, $u_t \in \{\texttt{User}, \texttt{System}\}$ generates an utterance s_t, and $\mathcal{I}_t \subseteq \mathcal{I}$ denotes the set of mentioned items. When $u_T = \texttt{System}$, we have groundtruth items $\mathcal{I}_T^{gt}$ (i.e., items with positive feedback) for the previous conversation $c = \{(u_t, s_t, \mathcal{I}_t)\}_{t=1}^{T-1}$. We denote the historical conversations as $C_{train} = \{c_1, c_2, \dots, c_{N_{train}}\}$. For a test conversation $c^{te} = \{(u_t^{te}, s_t^{te}, \mathcal{I}_t^{te})\}_{t=1}^{T'-1}$, the aim of this paper is to generate a ranked item list $\hat{\mathcal{I}}_{T'}^{te}$ from the catalog $\mathcal{I}$ by an LLM agent system with personalized verbal experience obtained from C_{train}, such that $\hat{\mathcal{I}}_{T'}^{te}$ best matches the groundtruth items in $\mathcal{I}_{T'}^{te}$ (if $\mathcal{I}_{T'}^{gt,te} \neq \emptyset$ and $u_{T'} = \texttt{System}$).

3 Methodology

3.1 Conversation Recommendation Agents

Conversational recommendation agents (CRA) is defined as a set of agents $\mathcal{A} = \{A_1, \dots, A_{N_A}\}$ that each A_i associates with an LLM-based policy $\pi_i(a|c, r_{-i})$, where c denotes the conversation, r_{-i} denotes the response history from agents in $\mathcal{A}$ before agent i takes action in the current step, and a is the action taken based on c and r_{-i} (typically involving reasoning and item recommendations).

3.1.1 ***Chain-of-Thought Agent.*** The simplest form of CRA is composed of only one chain-of-thought (COT) LLM agent. In this case, $\mathcal{A} = \{A_{cot}\}$ and the policy $\pi_{cot}(a|c, r_{-cot})$ reduces to $\pi_{cot}(a|c)$ where action a reasons with the user's preference based on the conversation c and makes recommendations accordingly. Despite the efficiency, COT can lead to the convergent thinking issue [25], which may not be able to provide good recommendations when user preferences are vague from the query.

3.1.2 ***Debater-Critic Agent System.*** In addition to the COT agent A_{cot}, we introduce a debater-critic agent (DCA) system for CRA that encourages divergent thinking on the user queries and generates recommendations that better cover the true user preferences. In DCA, LLM agents in $\mathcal{A}$ are divided into two parts, i.e., the *debaters* $\mathcal{D} = \{A_1, \dots, A_{N_A - 1}\}$ that sequentially evaluates the reasoning and recommendations of the previous debaters, and the *critic* $Q = A_{N_A}$ that judges the reasoning and recommendations of all the debaters and provides the final recommendation list. Here, $A_1 \in \mathcal{D}$ is a COT agent that starts the debate.

Nevertheless, given the open-end nature of the CRS task, a zero-shot DCA without prior experience may struggle to provide meaningful debates that maximally cover the user interests, and the critic may fail to effectively evaluate the reasoning and recommendations from the debaters. To address this challenge, we propose leveraging the training set C_{train} to gain verbal experience that can be collaboratively retrieved based on the conversation s, thereby guiding $\mathcal{D}$ and Q toward more effective debating and critiquing, respectively.

3.2 Collaborative Verbalized Policy Learning

To leverage the historical conversations C_{train} to improve the actions of the agents, traditional reinforcement learning (RL) generally parameterizes each policy $\pi_i(a|c, r_{-i})$ with a learnable neural network and optimizes it with gradient-based methods such as policy gradient [18]. However, since π_i in CRA is based on a black-box LLM, we adapt the verbal reinforcement learning (VRL) by adaptively augmenting policy $\pi_i(a|c, r_{-i})$ with an agent-specific retrieved experience $e(c; \mathcal{E}_i)$, i.e.,

$$a \sim \pi_i\left(a|c, r_{-i}; e(c, \mathcal{E}_i)\right), \tag{1}$$

where $\mathcal{E}_i$ is the *experience bank* for agent i that stores the verbalized experience associated with all the training conversations in C_{train} in natural language, and e is the *collaborative retrieval network* that selects the verbalized experiences in $\mathcal{E}_i$ based on **user preference similarity** of test conversation c with training conversations.

3.2.1 ***CRA Trajectory Sampling.*** To establish the experience bank $\mathcal{E}_i$ for agent i, we first collect trajectories of policy $\pi_i(a|c, r_{-i})$ on training conversations. To gain good experience, existing VRL methods typically require sampling multiple trajectories for each sample until the task succeeds [29]. However, since it is challenging to exactly identify all the groundtruth items for CRS due to the large item catalog and subtle user preference, we sample the trajectory only once for each c, which we empirically show can already establish a good experience bank. Specifically, for the j-th training sample, the trajectory for the COT agent A_{cot} can be sampled as:

$$r_{j,c} \sim \pi_i(a|c_j) = \Phi\left(T_c^f, F_c^f, c_j\right). \tag{2}$$

Here, T_c^f is the task-specific prompt that instructs the LLM (which we denote as Φ) to reason with user preference based on the conversation c_j and make recommendations accordingly, and F_c^f is the format instruction that guides the LLM agent to output its reasoning and recommendations that can be easily processed with string

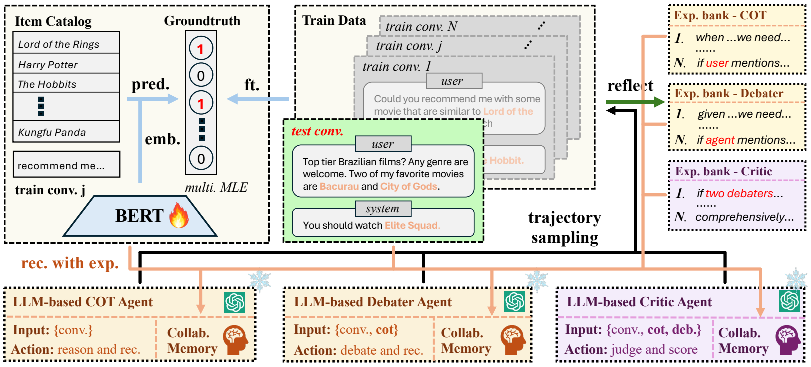


Figure 1: Overview of the proposed CRAVE framework for CRS and its three components.

split functions. Similarly, the trajectory sampling process for the DCA system can be formulated as follows:

$$r_{j,i} \sim \pi_i(a|s_j, r_{j,<i}) = \Phi\left(T_i^f, F_i^f, r_{j,<i}, c_j\right), \tag{3}$$

where $r_{j,1} = r_{j,c}$ is the response of the COT agent that starts the debate, and r_{j,N_A} is the judgment provided by the critic Q. DCA supports an arbitrary number of debaters with arbitrary debate rounds. However, due to the efficiency constraint of CRS (as we cannot ask the user to wait too long for the LLMs' debate), we consider only a one-round two-debater system and empirically show that it can already substantially improve over the COT agent.

In Eq. (3), the task specific prompt T_2 instructs the second debater A_2 to find issues in the reasoning and recommendations of the first COT agent A_{cot} and address the problems by first providing the correct reasoning on user preference and based on it making new recommendations. T_3 instructs the critic Q to comprehensively evaluate the reasoning of the two debaters and provide numerical scores (in the range of $[-2, 2]$) to judge the quality of the items recommended by the debaters.

3.2.2 *Verbalized Experience Collection.*

It is extremely challenging for the COT agent and the DCA system to reason over user preference based on conversations *in a zero-shot manner* as both have no prior experience on the actions that lead to good recommendations. Fortunately, items with positive feedback, i.e., $\mathcal{I}_j^{gt}$, are available for the historical conversations $c_j \in C$, which provide weak external guidance for the agents to reflect on their own actions. This allows for the summarization of useful experiences to guide their future actions when seeing similar conversations. For the COT agent A_{cot}, the reflection-based experience collection process can be formulated as follows:

$$e_{j,c} = \Phi\left(T_c^b, F_c^b, c_j, r_{j,c}, \mathcal{I}_j^{gt}\right), \tag{4}$$

where the task-specific prompt T_c^b instructs the LLM Φ to reflect on the action $r_{j,c}$ based on both the conversation c_j and the groundtruth items $\mathcal{I}_j^{gt}$. Specifically, the LLM is asked to assess the correctness of the reasoning and recommendations given in $r_{j,c}$, provide the rationale for the assessment, and finally offer useful experiences that can be applied to similar conversations in the future. Similarly, the experience collection process for agent i in the DCA system can be formulated as follows:

$$e_{j,i} = \Phi\left(T_i^b, F_i^b, r_{j,<i}, c_j, r_{j,i}, \mathcal{I}_j^{gt}\right), \tag{5}$$

where the prompt T_i^b is similar to that in Eq. (4). Compared to Eq. (4), Eq. (5) also considers the trajectory of previous agents, i.e., $r_{j,<i}$ when reflecting on the action $r_{j,i}$ with conversation c_j and groundtruth items $\mathcal{I}_j$. The final experience bank for DCA can be established as $\mathcal{E}_i = \{e_{j,i}\}_{j=1}^{N_{train}}$.

3.2.3 *Collaborative Experience Retrieval.*

After establishing the experience bank $\mathcal{E}_i$, we introduce the collaborative retrieval model $e(s, \mathcal{E}_i)$ defined in Eq. (1). The backbone model for e is a Sentence-BERT [15], which finetunes a BERT model $\mathbf{\Psi}$ on document pair-wise similarity. However, user queries with good semantic similarity do not necessarily share similar preferences. Therefore, we further finetune $\mathbf{\Psi}$ with *collaborative similarity*, such that the embeddings of conversations that share similar preferences become similar to each other. The user preference similarity of two queries c_i and c_j can be measured by the normalized overlap of groundtruth:

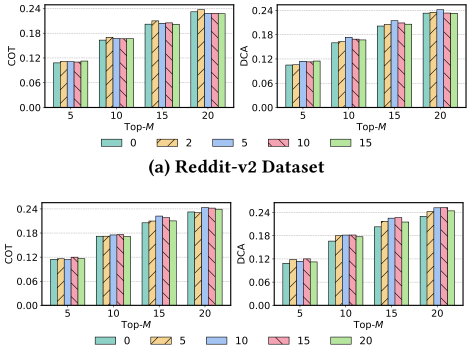
$$O_{i,j} = |\mathcal{I}_i^{gt} \cap \mathcal{I}_j^{gt}| / |\mathcal{I}_i^{gt} \cup \mathcal{I}_j^{gt}|. \tag{6}$$

However, $O_{i,j}$ simply counts the number of overlaps and ignores the item content, which is critical for conversational recommendations. Instead, we propose a novel *collaborative finetuning* strategy that encourages the embeddings of c_j to be simultaneously similar to the content of all the groundtruth items in $\mathcal{I}_j^{gt}$. This is achieved by maximizing the item content-parameterized multinomial likelihood on all the $(c_j, \mathcal{I}_j^{gt})$ pairs in C_{train} as follows:

$$\mathcal{I}_j \sim \text{multi}\left(\text{softmax}\left(\mathbf{\Psi}(s_j)\right), \mathbf{W}), |\mathcal{I}_j^{gt}|\right), \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathcal{I}| \times d} = \mathbf{\Psi}(T)$, and $T = \left[t_1 \dots t_{|\mathcal{I}|}\right]$ represents the content of all the catalog items described in natural language. For movies, we directly use the movie title as t_k, as the BERT model $\mathbf{\Psi}$ already gained sufficient knowledge of the movies through pretraining. d is the dimension of the content embeddings. During finetuning, we monitor the overlap of groundtruth items (Eq. (6)) of validation conversations with top-K retrieved training conversations. The collaborative retrieval network e is formulated as:

$$e\left(c^{te}, \mathcal{E}_i\right) = \left\{e_{k,i} | k \in \text{top}_K(\text{sim}(\hat{\mathbf{\Psi}}(c^{te}), \hat{\mathbf{\Psi}}(c_k)))\right\}, \tag{8}$$



(a) Reddit-v2 Dataset

(b) Redial Dataset

Figure 2: Performance of CRAVE with the COT agent and DCA system w.r.t. different number of retrieval K.

where the top_K function selects the indices of training conversations with the top K collaborative similarity judged by $\hat{\boldsymbol{\Psi}}$.

3.3 Experience Augmented Generation

Finally, combining the collected experience bank $\mathcal{E}_i$ and the collaborative retrieval network e finetuned from $\boldsymbol{\Psi}$, for the COT agent A_{cot}, the recommendations for a test conversation c^{te} with verbalized collaborative experience can be formed as:

$$r_c^{te} \sim \pi_c \left(a|c^{te}, e_c^{te}\right) = \Phi\left(T_c^{fe}, F_c^{fe}, c^{te}, e_c^{te}\right), \tag{9}$$

where the task-specific prompt T_c^{fe} instructs the LLM Φ to *utilize the retrieved verbalized collaborative experience* $e_c^{te} = e(c^{te}, \mathcal{E}_c)$ to reason with user preference based on the conversation c^{te} and make recommendations accordingly. In addition, with the format instruction F_c^{fe}, the recommendation list $\hat{\mathcal{I}}_c^{te}$ can be directly extracted from the response r_c^{te}. For DCA, collaborative experience augmented responses can be formulated as follows:

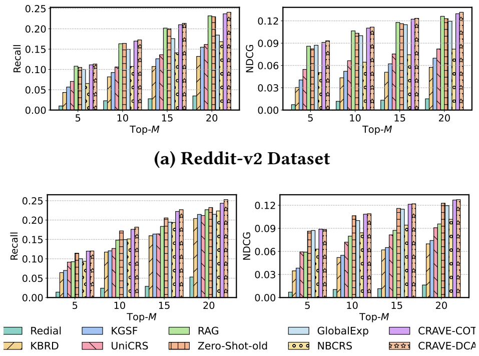
$$\begin{aligned} r_i^{te} \sim & \pi_i \left(a|c^{te}, r_{<i}^{te}, e_i^{te}\right) \\ &= \Phi\left(T_i^{fe}, F_i^{fe}, r_{<i}^{te}, c^{te}, e_i^{te}\right), \end{aligned} \tag{10}$$

where the instructions T_i^{fe} and F_i^{fe} are similar to those defined in Eqs. (3) and (9). Finally, after the experience augmented debate between the agents A_1 and A_2 the evaluation from the critic Q, item-score pairs are extracted from the response of Q, i.e., r_3^{te}, and the items are re-ranked by the quality score to form the final recommendation list $\hat{\mathcal{I}}_{dca}^{te}$.

4 Empirical Study

4.1 Datasets

We consider two widely-used real-world CRS datasets, i.e., the Redial dataset [13] and a movie-name corrected Reddit dataset [9] (which we name Reddit-v2). For both datasets, two people play the role of the user who seeks movies to watch and the system that responds with recommendations through conversations. For pre-processing, we keep the original training set and randomly split the original test set into a validation part to select the collaborative retrieval model and a test part for the final CRA evaluation. The statistics of the two datasets are summarized in Table 2 for reference.



(a) Reddit-v2 Dataset

(b) Redial Dataset

Figure 3: Comparison between CRAVE and various baselines on the Reddit-v2 and Redial datasets.

4.2 Performance w.r.t. Number of Retrieval

For CRAVE, the number of training samples from which the collaborative experiences are retrieved, i.e., K in Eq. (8), is an important hyperparameter. Therefore, we first explore the performance of CRAVE for both the COT agent and DCA system when K increases. The results are illustrated in Fig. 2. Fig. 2 shows that the performance of CRAVE first peaks and then drops as K gets larger. This is because when K is too small, insufficient experiences are retrieved from the bank, which fail to guide the CRA to take reasonable actions that lead to good recommendations. However, when K is too large, less relevant experiences may be retrieved, which may risk biasing the recommendations. In addition, we note that a smaller value of K leads to optimal performance on the Reddit-v2 dataset. This is probably because of its more diverse topics compared with the Redial dataset, where a small neighborhood keeps the retrieved most relevant to the test query. Finally, we note that DCA cannot outperform COT in a zero-shot manner. However, it benefits significantly from CRAVE and outperforms COT when collaborative experience is retrieved to facilitate the debate and the critique.

4.3 Comparison with Baselines

We now use the best K selected on the validation set and compare CRAVE with various state-of-the-art RNN-based, finetuned PLM-based, and LLM-based CRS baselines as follows:

- **Redial** [13] uses an RNN to model conversations and a denoising autoencoder to model items and generate recommendations.
- **KBRD** [3] introduces a relational graph neural network (GNN) on the DBpedia to model entities, and optimize similarity between tokens and entities to fuse semantics.
- **KGSF** [32] incorporates ConceptNet to model the conversations, with mutual information maximization w.r.t. entity KG embeddings to fuse the entity information.
- **UniCRS** [24] introduces a pretrained transformer, i.e., DialoGPT, to capture the context information w.r.t. the entity KG embeddings used for semantic fusion.
- **Zero-shot LLM** [9] directly inputs the historical dialogue with task-specific prompt and formats instructions for CRS without any retrieval from the external knowledge database.

- **RAG** [12] denotes the model that retrieves item-related sentences from a database of movie plots and metadata based on semantic similarity between the query and sentences.
- **NBCRS** [28] uses Sentence-BERT to retrieve training queries and take vote of groundtruths as recommendations. Neighbor size is selected based on the validation set.
- **GlobalExp** proposed in [27] goes through all training samples and summarized rules that are shared among all test queries.

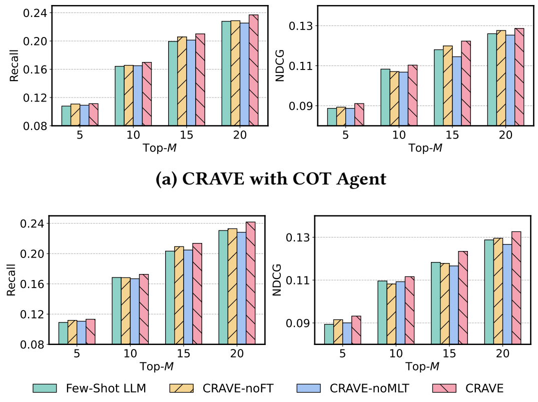
The results are illustrated in Fig. 3. From the figure we can find that, although not finetuned on any CRS-specific data, the zero-shot LLM is already the strongest baseline that outperforms Redial, KBRD, KGSF, and UniCRS, where laborious training is required to achieve good performance. We also observe that NBCRS, i.e., a simple neighborhood-based method, outperforms most traditional methods. Furthermore, we note that naively retrieving the movie content with RAG to augment the prompt does not help, which is probably due to the large semantic gap between the content and user preference. In addition, we also observe that the inclusion of global rules summarized from the training data may also degrade the performance of the zero-shot LLM, as CRS is a highly query-dependent task, and applying the same rules for all the queries may not cater to the users' personalized preferences. Augmented with verbalized collaborative experience gained via reflection on user feedback on historical recommendations, we can see in Fig. 3 that the proposed CRAVE outperforms all the baselines.

4.4 Ablation Study

We conduct the ablation study to demonstrate the effectiveness of the ***(i)*** verbalized experience collection module and the ***(ii)*** collaborative experience retrieval module in CRAVE. To answer the research questions, we design the following baselines:

- **Few-shot LLM** directly retrieves the query-groundtruth pairs from C_{train} to augment the test query instead of summarizing the experiences by reflecting on the actions on C_{train}.
- **CRAVE-noFT** directly uses pretrained Stella-400M model for experience retrieval instead of finetuning it with item content parameterized multinomial likelihood defined in Eq. (7).
- **CRAVE-noMLT** uses the normalized overlap metric in Eq. (7) as the pairwise loss for conversations to finetune the retrieval network, without considering item content information.

For all the ablation models, the best number of shots is determined by the validation set the same way as CRAVE. The results are illustrated in Fig. 4. From Fig. 4, we can first note the favorable comparison of CRAVE with the few-shot LLM baseline. This further demonstrates the effectiveness of the reflection-based agent-specific experiences introduced in CRAVE, as LLMs may fail to directly generalize from the in-context demonstrations due to the large semantic gap between the conversations and user preferences. In addition, the substantial improvement of CRAVE over the CRAVE-noFT and CRAVE-noMLT variants further shows the effectiveness of the collaborative retrieval network in retrieving user-preference-oriented experiences for the conversational recommendation agents. In addition, we note that finetuning the retrieval network with a non-content based metric (i.e., CRAVE-noMLT) actually decreases the performance compared to the variant with no finetuning at all (i.e., CRAVE-noMLT), which highlights the importance of both collaborative and content information for the LLM-based CRSs.



(a) CRAVE with COT Agent

(b) CRAVE with DCA System

Figure 4: Comparison between CRAVE and various ablation models on the Reddit-v2 dataset.

Table 1: Comparison of in-list recommendation similarity (*inv. diversity*) between CRAVE for COT agent and DCA system.

Methods	Reddit-v2		Redial	
	cont. ↓	collab. ↓	cont. ↓	collab. ↓
CRAVE-COT	0.505	0.431	0.465	0.457
CRAVE-DCA	0.458	0.425	0.449	0.454

5 Diversity Analysis

Finally, we compare the recommendation diversity of CRAVE with COT and DCA backbones in Table 1. Specifically, we consider two aspects of in-list recommendation diversity, i.e., content diversity and collaborative diversity. We use the average in-list pairwise similarity of Stella-400M embeddings for the titles of recommended movies (averaged over all test conversations) to measure the content diversity. Additionally, we assess collaborative similarity with movie embeddings derived by training an EASE [20] model on item co-mentions in the training conversations. However, due to the sparsity of item co-mention data, this is not as sensitive as the content metric. The results indicate that DCA augmented with verbal experience not only improves recommendation accuracy but also enhances diversity compared with the COT agent.

6 Conclusions

In this paper, we introduced CRAVE, i.e., conversational recommendation agents with verbalized collaborative experience. We show that valuable experience for the agents can be gained from the training set via trajectory sampling and self-reflection, which is largely unexplored for previous LLM-based CRS. We also find that the improvement is especially evident for the debater-critic system as CRS, which resembles an open-end debate without definite groundtruth. In addition, we show that the collaborative retriever network, which is a fine-tuned BERT model that encodes preference similarity of conversations, plays a key role in CRAVE.

References

[1] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. Dbpedia: A nucleus for a web of open data. In *International Semantic Web Conference* (2007), Springer, pp. 722–735.

[2] Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., and Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate. In *ICLR* (2024).

[3] Chen, Q., Lin, J., Zhang, Y., Ding, M., Cen, Y., Yang, H., and Tang, J. Towards knowledge-based recommender dialog system. In *EMNLP* (2019).

[4] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Workshop* (2014).

[5] Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., et al. A survey on in-context learning. In *EMNLP* (2024), pp. 1107–1128.

[6] Fang, J., Gao, S., Ren, P., Chen, X., Verberne, S., and Ren, Z. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135* (2024).

[7] Feng, Y., Liu, S., Xue, Z., Cai, Q., Hu, L., Jiang, P., Gai, K., and Sun, F. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212* (2023).

[8] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[9] He, Z., Xie, Z., Jha, R., Steck, H., Liang, D., Feng, Y., Majumder, B. P., Kallus, N., and McAuley, J. Large language models as zero-shot conversational recommenders. In *CIKM* (2023).

[10] Jannach, D., Manzoor, A., Cai, W., and Chen, L. A survey on conversational recommender systems. *CSUR 54*, 5 (2021), 1–36.

[11] Kim, G., Baldi, P., and McAleer, S. Language models can solve computer tasks. In *NeurIPS* (2024).

[12] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS* (2020), pp. 9459–9474.

[13] Li, R., Ebrahimi Kahou, S., Schulz, H., Michalski, V., Charlin, L., and Pal, C. Towards deep conversational recommendations. In *NeurIPS* (2018).

[14] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[15] Reimers, N. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[16] Rendle, S. Factorization machines. In *ICDM* (2010), pp. 995–1000.

[17] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS* (2024).

[18] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML* (2014), Pmlr, pp. 387–395.

[19] Speer, R., Chin, J., and Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI* (2017).

[20] Steck, H. Embarrassingly shallow autoencoders for sparse data. In *WWW* (2019), pp. 3251–3257.

[21] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., and Gomez, A. N. Attention is all you need. In *NeurIPS* (2017).

[23] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *ICML* (2008), pp. 1096–1103.

[24] Wang, X., Zhou, K., Wen, J.-R., and Zhao, W. X. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *KDD* (2022), pp. 1929–1937.

[25] Wang, Y., Liu, Z., Zhang, J., Yao, W., Heinecke, S., and Yu, P. S. Drdt: Dynamic reflection with divergent thinking for llm-based sequential recommendation. *arXiv preprint arXiv:2312.11336* (2023).

[26] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS* (2022).

[27] Xi, Y., Liu, W., Lin, J., Chen, B., Tang, R., Zhang, W., and Yu, Y. Memocrs: Memory-enhanced sequential conversational recommender systems with large language models. In *CIKM* (2024), pp. 2585–2595.

[28] Xie, Z., Wu, J., Jeon, H., He, Z., Steck, H., Jha, R., Liang, D., Kallus, N., and McAuley, J. Neighborhood-based collaborative filtering for conversational recommendation. In *RecSys* (2024), pp. 1045–1050.

[29] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. In *ICLR* (2023).

[30] Zhang, Z., Bo, X., Ma, C., Li, R., Chen, X., Dai, Q., Zhu, J., Dong, Z., and Wen, J.-R. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501* (2024).

[31] Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J., and Huang, G. Expel: Llm agents are experiential learners. In *AAAI* (2024), pp. 19632–19642.

[32] Zhou, K., Zhao, W. X., Bian, S., Zhou, Y., Wen, J.-R., and Yu, J. Improving conversational recommender systems via knowledge graph based semantic fusion. In *KDD* (2020), pp. 1006–1014.

Appendix

A Related Work

A.1 LLM Agent with Verbalized Experience

The simplest form of verbal experience for LLM agents is memorization [30], i.e., documents [8] or few-shot examples [5] retrieved based on semantic relevancy w.r.t. the prompt. However, in-context learning on the retrieved memories can be challenging for tasks that require complex reasoning. To bridge the semantic gap, verbal reinforcement learning (VRL) was proposed to enhance the experiences by self-reflection on the LLM agent's actions based on external feedback. For instance, RCI [11] iteratively prompts the LLM to critique and improve upon its previous output until the answer is correct, whereas Reflexion [17] gains experience by reflecting on the failures. These methods typically require multiple attempts on a single test query, which is not feasible for CRS. Recently, EXPEL [31] was proposed to retrieve experiences across tasks based on task-task semantic similarity. However, naive semantic similarity cannot be directly applied to CRS due to the significant semantic gap between the conversation and user preferences. To the best of our knowledge, the only work that utilizes verbalized experience for LLM-based CRS is the global memory introduced in [27], which iteratively reflects on reasoning and recommendations of an LLM-based CRS on the training samples to learn global rules that can be augmented in the prompt to support future recommendations.

A.2 Conversational Recommender System

Generally, CRS consists of two main modules: ***(i)*** conversation and ***(ii)*** recommendation [6, 10]. This paper focuses on the recommendation aspect of CRS, i.e., suggesting new items to users based on their previous conversations with the system. Traditional methods [24, 32] rely on training sequential models, such as RNNs [4] or transformers [22], to understand the conversations and integrate them with the item information learned from traditional recommendation models [16, 23]. These approaches often incorporate external knowledge databases, such as DBPedia [1] and ConceptNet [19], as the item/word prior knowledge. Recently, pretrained language models (PLM) [21] have gained more attention in CRS research, as they encapsulate prior knowledge of both natural language and items through pretraining on external corpora, which is beneficial for both conversation and item modeling in CRS [24]. Recently, large language models (LLM) with hundreds of billions of parameters, e.g., GPT-4o [14], have emerged as the strongest baseline in CRS. [9] demonstrated that these LLMs achieve excellent zero-shot recommendations, significantly outperforming both traditional and finetuned PLM-based methods. The aim of this paper is to further enhance the strongest zero-shot LLMs by developing a collaborating LLM-based CRS agent system that incorporates collaborative experience gained through self-reflection on historical conversations.

A.3 Implementation Details

Previous works found that the state-of-the-art GPT-4o [14] achieves the best zero-shot performance on both Redial and Reddit-v2 datasets [9]. In our experiment, we use GPT-4o as the backbone for the policies of the agents and show that CRAVE can further improve the zero-shot performance by forming a CRA system with GPT-4o with augmented collaborative experience. For the collaborative retrieval network, we leverage the 400M Stella model[1] as the backbone and finetune it as Eq. (7) for 7 epochs with the learning rate $1e^{-5}$. The normalized overlap metric defined in Eq. (6) on the validation set was monitored during training (see Fig. 5 for the training dynamics averaged over ten independent training runs of the collaborative retrieval network), where the best $\hat{\Psi}$ is saved and used for retrieving verbalized collaborative experience in the following sections.

Table 2: Statistics of Redial and Reddit-v2 datasets.

dataset	#items	#train	#val/test
Redial	8,010	186,546	11,564
Reddit-v2	20,193	108,989	7,565

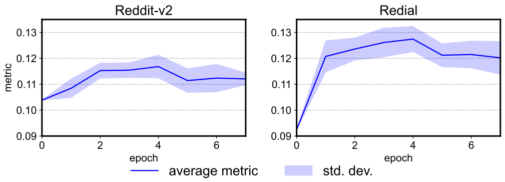


Figure 5: Dynamic of finetuning collaborative retrieval network with content-parameterized mult. likelihood.

B Prompts Used in the Main Paper

In this section, we provide the task-specific prompt and format instructions that we defined in the main paper for trajectory sampling, experience reflection, and recommendation stages of CRAVE.

Eq. (2): COT Agent, Trajectory Sampling

T_c^f: Pretend that you are a movie recommender system. Here is the user's query: $\{c_j\}$

F_c^f: Specifically, after writing down your reasoning, write #### to mark the beginning of your recommendation list. Then, list EXACTLY 20 movie recommendations, each on a new line with no extra sentences.

Eq. (3): Debater Agent, Trajectory Sampling

T_2^f: Pretend you are a movie recommender system. Here is a user's query: $\{c_j\}$. Below is the reasoning and recommendation list from another movie recommender system: $\{r_{j,1}\}$. Evaluate the reasoning for any potential issues. Even if the reasoning is sound, the provided recommendations may not align well with it. Analyze these aspects and provide your corrected reasoning and recommendations.

[1] huggingface.co/dunzhang/stella_en_400M_v5.

F_2^f : After completing your reasoning, write #### to indicate the start of your recommendation list. Then, list EXACTLY 20 movie recommendations, each on a new line with no extra sentences.

Eq. (3): Critic Agent, Trajectory Sampling

T_3^f : You are a judge for a debate on movie recommendations for the user query: $\{c_j\}$. The debate between two movie recommender systems is as follows: Movie Recommender System 0: $\{r_{j,1}\}$ \n\n Movie Recommender System 1: $\{r_{j,2}\}$ \n\n Your task is to reflect on both movie recommender systems and comprehensively critique the reasoning and recommendations from each side. After providing your analysis, generate scores for each movie from both recommender systems to indicate the quality of the recommendation. Use the following scale: -2 for very bad, -1 for bad, 0 for neutral, 1 for good, and 2 for very good.

F_3^f :Write #### to mark the beginning of your judgment on the recommendation list. Then, list the movies from both sides with their scores in the format: movie_name####score, each on a new line with no extra sentences.

Eq. (4): COT Agent, Experience Reflection

T_c^b: You are evaluating a movie recommender system. Assess the reasoning and recommended movies based on the user query: $\{c_j\}$. Here is the reasoning and recommended movies from the system: $\{r_{j,c}\}$. Here are the ground truth movies the user wants to watch: $\{\mathcal{I}_j\}$. Determine if the reasoning and recommendations are successful by checking the consistency with the ground truth movies and the overlap with recommended movies.

F_c^b: First, provide your judgment: success/failure, followed by ####. Next, analyze why the reasoning/recommendations succeed or fail based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for making movie recommendations for similar user queries, based on your analysis.

Eq. (5): Debater Agent, Experience Reflection

T_2^b: You are evaluating a debate on movie recommendations. Assess the reasoning and recommended movies of Debater 1 based on the user query: $\{c_j\}$ and the initial argument by Debater 0: $\{r_{j,1}\}$. Here is the reasoning and recommended movies of Debater 1: $\{r_{j,2}\}$. Here are the ground truth movies the user wants to watch: $\{\mathcal{I}_j\}$ \n\n Determine if Debater 1's reasoning and recommendations are successful by checking the consistency with the ground truth movies and the overlap with recommended movies.

F_2^b: First, provide your judgment: success/failure, followed by ####. Next, analyze why Debater 1's reasoning/recommendations succeed or fail based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for Debater 1 when debating for similar user queries, based on your analysis.

Eq. (5): Critic Agent, Experience Reflection

T_3^b: You are evaluating a critique on a debate about movie recommendations. Assess the critique provided on the recommendations from two movie recommender systems based on the user query: $\{c_j\}$. Here is the reasoning and recommendations from Debater 0: $\{r_{j,1}\}$ \n\n Here is the reasoning and recommendations from Debater 1: $\{r_{j,2}\}$ \n\n Here is the critique you need to evaluate: $\{r_{j,3}\}$ \n\n Here are the ground truth movies the user wants to watch: $\{\mathcal{I}_j\}$. Determine if the critique's reasoning and recommendations are successful by checking consistency with the ground truth movies and overlap with highly scored movies.

F_3^b: First, provide your judgment: success/failure, followed by ####. Next, analyze why the critique's reasoning/scoring succeeds or fails based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for critiquing movie recommendations for similar user queries, based on your analysis.

Eq. (9): COT Agent, Exp. Augmented Gen.

T_c^{fe}: Pretend you are a movie recommender system. Here is a user's query $\{c^{te}\}$. When making recommendations, consider the following guidelines: $\{e_c^{te}\}$ \n\n

F_c^{fe}: First, provide your judgment: success/failure, followed by ####. Next, analyze why the critique's reasoning/scoring succeeds or fails based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for critiquing movie recommendations for similar user queries, based on your analysis.

Eq. (10): Debater Agent, Exp. Augmented Gen.

T_2^{fe}: Pretend you are a movie recommender system. Here is a user's query: $\{c^{te}\}$. Below is the reasoning and recommendation list from another movie recommender system: $\{r_1^{te}\}$. Evaluate the reasoning for any potential issues. Even if the reasoning is sound, the recommendations may not align well with it. Analyze these aspects and provide your corrected reasoning and recommendations. When doing evaluation and making recommendations, consider the following guidelines: $\{e_c^2\}$ \n\n

F_2^{fe}: After completing your reasoning, write #### to indicate the start of your recommendation list. Then, list EXACTLY 20 movie recommendations, each on a new line with no extra sentences.

Eq. (10): Critic Agent, Exp. Augmented Gen.

T_3^{fe}: You are a judge for a debate on movie recommendations for the user query: $\{c^{te}\}$. The debate between two movie recommender systems is as follows: Movie Recommender System 0: $\{r_1^{te}\}$ \n\n Movie Recommender System 1: $\{r_2^{te}\}$ \n\n Your task is to reflect on both movie recommender systems and comprehensively critique the reasoning and recommendations from each side. After providing your analysis, generate scores for each movie from both recommender systems to indicate the quality of the recommendation. Use the following scale: -2 for very bad, -1 for bad, 0 for neutral, 1 for good, and 2 for very good. Consider the following rules when you make the judgment: $\{e_c^3\}$ \n\n

F_3^{fe}: Write #### to mark the beginning of your judgment on the recommendation list. Then, list the movies from both sides with their scores in the format: movie_name####score, each on a new line with no extra sentences.

C Qualitative Analysis

In this section, we present a qualitative analysis of the retrieved experiences for CRAVE with the COT and DCA backbones, as shown in Tables 3 and 4. From these tables, we observe that the retrieved experiences maintain a good balance between generalization and catering to the specific preferences expressed in the query, which results in overall better recommendations compared to zero-shot LLMs.

Table 3: Qualitative analysis of the recommendations of zero-shot LLM and CRAVE for the COT Agent.

User Query	Recommendations and Groundtruth
User: Films with news headline montages. Anyone have suggestions of films that have a montage of news headlines and newscasters? Looking for some montage style references.	**COT (Zero-shot)**: Citizen Kane, Good Night, and Good Luck, The Social Network, All the President's Men **COT (CRAVE)**: **Network**, Citizen Kane, The Truman Show, Wag the Dog, Good Night, and Good Luck **Groundtruth:** Network
Exemplar Experiences **Broaden the thematic scope:** While focusing on media and advertising, consider films that offer a critical or satirical perspective on society, even if they don't directly address media themes. **Consider Tone and Style:** Pay attention to the tone and style of the films the user mentions, such as satire or drama, and ensure recommendations align with these preferences.	
User: Movies where a big-timer befriends a small-timer... then the small-timer gets into a major conflict with the big-timer. Some examples: Last King of Scotland, A Bronx Tale, and Gangs of New York.	**COT (Zero-shot)** The Devil's Advocate, The Departed, Training Day, Scarface, The Godfather **COT (CRAVE)** The Departed, **American Gangster**, Donnie Brasco, The Devil's Advocate, Training Day **Groundtruth:** American Gangster
Exemplar Experiences **Align with User's Examples:** Pay close attention to the examples provided by the user and ensure that recommendations closely match the tone, style, and themes of those examples. **Understand the Core Criteria:** Identify the core elements the user is interested in, such as unexpected temper, hero-type characters, and memorable scenes, and ensure recommendations embody these traits.	
User: Movies with/about unsuspected murderers. Like a character, main or not, who seems so innocent but ends up ruthlessly murdering someone or multiple people.	**COT (Zero-shot)** Psycho, Gone Girl, The Usual Suspects, **Primal Fear**, Scream **COT (CRAVE)** **Primal Fear**, Gone Girl, The Talented Mr. Ripley, Se7en, The Usual Suspects **Groundtruth:** Primal Fear
Exemplar Experiences **Clarify User Intent:** Ensure a clear understanding of whether the user is more interested in the procedural aspect of the investigation or the psychological exploration of the killer. **Balance Explicit and Implicit Interests:** While addressing the explicit request, also consider the user's implicit interests, which might be inferred from the examples they provide or their viewing history.	
User: The main character is accused of something and thrown in jail/imprisoned somewhere for a long time. Something like The Count of Monte Cristo.	**COT (Zero-shot)** The Count of Monte Cristo, V for Vendetta, The Green Mile, The Shawshank Redemption, Kill Bill: Vol. 1 **COT (CRAVE)** V for Vendetta, **Oldboy**, The Shawshank Redemption, Law Abiding Citizen, The Prestige **Groundtruth:** Oldboy
Exemplar Experiences **Consider User-Provided Examples:** Use examples as strong indicators of their preferences and ensure that similar movies are prioritized in the recommendations. **Include a Broader Range:** Include a broader range of movies that fit the theme, including lesser-known films that might align with the user's interests.	

Table 4: Qualitative analysis of the recommendations of zero-shot LLM and CRAVE for the DCA system.

<table>
<tr><th>User Query</th><th>Recommendations and Groundtruth</th></tr>
<tr><td><b>User:</b> Any films that will leave me feeling more intelligent?. Looking for some well-made films that will get me thinking, have me learning, leave me questioning. Any suggestions?</td><td><b>DCA (Zero-shot)</b>: Inception, The Matrix, Interstellar, Eternal Sunshine of the Spotless Mind, Arrival
<b>DCA (CRAVE)</b>: <b>Synecdoche</b>, <b>New York</b>, The Tree of Life, The Seventh Seal, The Double Life of Véronique, Solaris
<b>Groundtruth:</b> Synecdoche, New York</td></tr>
<tr><td colspan="2"><b>Exemplar Experiences from Debater 1</b>
<b>Understand User Preferences:</b> Pay close attention to the specific themes and types of movies the user is interested in, such as existential or philosophical themes, rather than focusing solely on science and technology.
<b>Diverse Themes:</b> Include a broader range of themes, such as personal growth, emotional depth, and interconnectedness, to better match the user's interests.

<b>Exemplar Experiences from Critic</b>
<b>Alignment with User Preferences:</b> Ensure that the recommendations align with the user's specific interests and preferences, as indicated by any ground truth or explicit requests.
<b>Diversity in Themes and Genres:</b> While maintaining a focus on the user's request (e.g., cerebral films), include a variety of themes and genres to capture different aspects of the user's interests.</td></tr>
<tr><td><b>User:</b> What movie feels like a warm blanket?. The most notable one for me is Harry Potter, but these days i like a little bit more grit. So lately Prometheus has topped it. Some others that come to mind for me are Blade Runner, the Planet of the Apes trilogy, LOTR, the Holiday, Hunger Games, True Detective. I'm realizing most of these are fantasy, but they don't have to be. I'm looking for that rainy day feeling that wraps you up. I get a cup of tea and bundle up and pretend i live in a city that ever gets cold, which i don't. I'd love to have some recommendations.</td><td><b>Critic (Zero-shot)</b>: Pan's Labyrinth, Children of Men, Annihilation, Ex Machina, The Shape of Water
<b>Critic (CRAVE)</b>: Pan's Labyrinth, Arrival, <b>The Lord of the Rings: The Fellowship of the Ring</b>, The Shape of Water, The Grand Budapest Hotel
<b>Groundtruth:</b> The Lord of the Rings: The Fellowship of the Ring</td></tr>
<tr><td colspan="2"><b>Exemplar Experiences from Debater 1</b>
<b>Understand User Preferences:</b> Pay close attention to the examples and preferences provided by the user to tailor recommendations that align with their desired vibe or theme.
<b>Emphasize Comfort and Warmth:</b> Prioritize films that are uplifting, heartwarming, and comforting, especially when the user is seeking a cozy experience.

<b>Exemplar Experiences from Critic</b>
<b>Understand User Preferences:</b> Carefully analyze the user's query and any examples they provide to understand their preferences. Look for themes, genres, or specific qualities that the user is seeking in their movie recommendations.
<b>Evaluate Alignment:</b> Assess how well the recommended movies align with the user's preferences. Consider whether the films match the themes, emotional tone, and atmosphere that the user is looking for.</td></tr>
</table>