

Multiple Choice Questions based Multi-Interest Policy Learning for Conversational Recommendation

Yiming Zhang*

Tongji University
China
2030796@tongji.edu.cn

Lingfei Wu*

JD.COM Silicon Valley
Research Center
USA
lwu@email.wm.edu

Qi Shen

Tongji University
China
1653282@tongji.edu.cn

Yitong Pang

Tongji University
China
1930796@tongji.edu.cn

Zhihua Wei†

Tongji University
China
zhihua_wei@tongji.edu.cn

Fangli Xu

Squirrel AI Learning
USA
fxu02@email.wm.edu

Bo Long

JD.COM
China
bo.long@jd.com

Jian Pei

Simon Fraser University
Canada
jpei@cs.sfu.ca

ABSTRACT

Conversational recommendation system (CRS) is able to obtain fine-grained and dynamic user preferences based on interactive dialogue. Previous CRS assumes that the user has a clear target item, which often deviates from the real scenario, that is for many users who resort to CRS, they might not have a clear idea about what they really like. Specifically, the user may have a clear single preference for some attribute types (e.g. brand) of items, while for other attribute types (e.g. color), the user may have multiple preferences or even no clear preferences, which leads to multiple acceptable attribute instances (e.g. black and red) of one attribute type. Therefore, the users could show their preferences over items under multiple combinations of attribute instances rather than a single item with unique combination of all attribute instances. As a result, we first propose a more realistic conversational recommendation learning setting, namely Multi-Interest Multi-round Conversational Recommendation (MIMCR), where users may have multiple interests in attribute instance combinations and accept multiple items with partially overlapped combinations of attribute instances. To effectively cope with the new CRS learning setting, in this paper, we propose a novel learning framework, namely Multiple Choice questions based Multi-Interest Policy Learning (MCMPL). In order to obtain user preferences more efficiently, the agent generates multiple choice questions rather than binary yes/no ones on specific attribute instance. Furthermore, we propose a union set strategy to select candidate items instead of existing intersection set strategy in order to overcome over-filtering items during the conversation. Finally, we design a Multi-Interest Policy Learning (MIPL) module, which utilizes captured multiple interests of the user to decide next action, either asking attribute instances or recommending items. Extensive experimental results on four datasets

demonstrate the superiority of our method for the proposed MIMCR setting. The implementation of our proposed models is publicly available at <https://github.com/ZYM6-6/MCMPL>.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval; Recommender systems.**

KEYWORDS

Conversational Recommendation, Reinforcement Learning, Graph Representation Learning

ACM Reference Format:

Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Bo Long, and Jian Pei. 2022. Multiple Choice Questions based Multi-Interest Policy Learning for Conversational Recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*, April 25–29, 2022, Virtual Event, Lyon, France. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3485447.3512088>

1 INTRODUCTION

Conversational recommendation system (CRS) aims to obtain fine-grained and dynamic user preferences and make successful recommendations through conversations with users [13, 40]. In each conversation turn, CRS can select different actions [12] based on user feedback, either asking attributes or recommending items. Since it is able to explicitly obtain user preferences and has the advantage of conducting explainable recommendation, CRS has become one of the hot topics in current research.

Various methods [7, 11, 18, 48] have been proposed to improve the performance of CRS based on different problem settings. In this work, we focus on the multi-round conversational recommendation (MCR) setting [8, 13, 15], which is the most realistic CRS setting so far. The system focuses on whether asking attributes or recommending items in each turn, and adjusts actions flexibly via user feedback to make successful recommendations with fewer turns.

Despite the success of MCR in recent years, the assumption of the existing MCR [13], that the user preserves clear preferences towards all the attributes and items, may often deviate from the real scenario. For the user who resorts to CRS, he might not have a clear idea about what he really likes. Specifically, the user may have a clear single preference for some **attribute types** (e.g., color) of items,

*Both authors contributed equally to this research.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9096-5/22/04...\$15.00
<https://doi.org/10.1145/3485447.3512088>

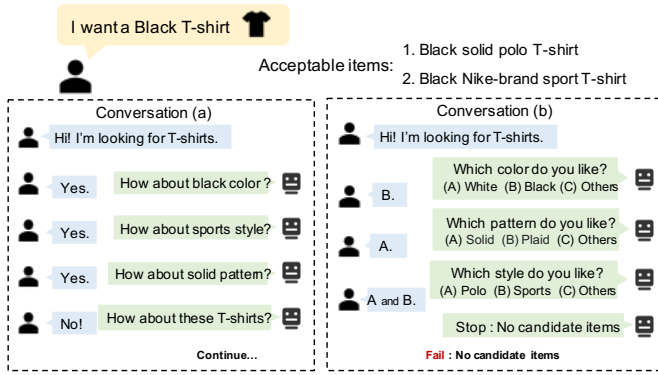


Figure 1: Examples of MIMCR scenario.

while for other attribute types (e.g., brand), the user might have multiple preferences or even no clear preferences. With the guidance of CRS, he may accept multiple **attribute instances** (e.g., red and black) of one attribute type. In addition, different combinations of these attribute instances are generally associated with different items. Therefore, the user could show his preferences over items under multiple combinations of attribute instances rather than a single item with unique combination of all attribute instances.

To this end, we extend the MCR to a more realistic scenario, namely Multi-Interest Multi-round Conversational Recommendation (MIMCR), in which users may have multiple interests in attribute instance combinations and accept multiple items with partially overlapped combinations of attribute instances. As shown in Figure 1, the user wants a black T-shirt. For the attribute types such as "style" or "brand", he can accept one or more instances. He shows interest in the combinations of "Nike-brand" and "sports", as well as "solid" and "polo" respectively. The user could accept a "black solid polo" T-shirt or a "black Nike-brand sports" T-shirt. The task will be completed as CRS successfully recommends one of them.

Existing works may encounter three significant limitations under the MIMCR scenario. First, current CRS frameworks often employ binary questions [13], which is concise but unable to elicit user interests effectively. As shown in the conversation (a) in Figure 1, although the user accepts all of the attribute instances asked by CRS, the combination of them does not point to any target items the user prefers. Moreover, since the CRS agent has asked attribute instance "sports", it will hardly ask "polo" (the user favors). This is the result of the mutual exclusion of attribute instances with the same attribute type in the current CRS system design. On the other hand, enumerating all choices (associated with each attribute instances) [13, 23, 46] are not practical since there may be too many attribute instances to be shown and answered by the user. Second, as shown in the conversation (b) in Figure 1, CRS can efficiently obtain user preferences by using multiple choice questions. However, the existing methods utilize the intersection set strategy to select items that are associated with all accepted attribute instances, which could easily lead to the over-filter of user preferred candidate items as the conversation progresses. Finally, the existing methods simply model user's intentions in a uniform manner, while neglecting the diversity of user interests, which will often fail to identify the user's multiple interests through the combinations of attribute instances.

To effectively address the aforementioned challenges, we propose a novel framework named Multiple Choice questions based Multi-Interest Policy Learning (MCM IPL) for MIMCR. In order to obtain user preferences more efficiently, our method generates attribute type-based multiple choice questions. As the conversation (b) in Figure 1, the user can flexibly select the attribute instances he likes or the option "Others" if he likes none. To avoid over-filtering items, we propose a union set strategy to select candidate items. In particular, we select the items satisfying at least one of the accepted attribute instances as the candidate items. Moreover, we develop a Multi-Interest Policy Learning (MIPL) module to decide the next action, either asking or recommending items. In details, we construct a current graph based on the conversation state, and a global graph based on the historical user-item interactions and the global item-attribute instance correlations. Based on the representation learned by graph neural network (GNN), we iteratively capture multiple interests of the user. Finally, the next action will be decided based on the policy learning with the multi-interest representations.

The contributions of this work are summarized as follows:

- We extend existing CRS to a more realistic scenario setting named MIMCR, which comprehensively takes into account the incompleteness and diversity of user's interests.
- For the MIMCR scenario, we propose the MCM IPL framework with more appropriate strategies to generate questions and select candidate items. Furthermore, our method iteratively extracts the user's multiple interests based on the current state and historical global information, to decide the next action via policy learning.
- We adapt four datasets for MIMCR, and extensive experimental results on these datasets show the superiority of our method.

2 RELATED WORKS

2.1 Conversational Recommendation

Compared to existing sequential or social recommendation systems [21, 27, 41], Conversational Recommendation System (CRS) is an effective solution for dynamic user preference modeling and explainable recommendation, originated from task-oriented dialogue systems [14]. Through the conversations with users, CRS collects the user's preference and then generates recommendations directly. In recent years, various approaches [4, 11, 16, 22, 38, 43] based on deep learning and reinforcement learning (RL) have been proposed for CRS. Multi-Armed Bandits based methods [7, 18, 40] and meta-learning based methods [11, 48] solve the user cold-start problem and balance the exploration and exploitation trade-offs for CRS. Besides, some methods [26, 42, 47] focus on asking questions about items to obtain the users' preference. In addition, the approaches focusing on the dialogue ability [4, 17, 43], are more likely to understand user's preferences and intentions with the input of raw natural language, and automatically generate fluent responses.

The most realistic conversational recommendation setting proposed so far is multi-round conversational recommendation (MCR) [8, 13, 15, 39]. In MCR task, the system focuses on whether to ask attributes or make recommendations based on policy learning at each turn to hit the target item for fewer interaction turns to improve the user experience. In this work, we focus on the MCR problem.

2.2 Multi-round Conversational Recommendation

For multi-round conversational recommendation, a conversation strategy is essential in the interaction process. The key of the conversation strategy is to dynamically decide when to ask questions, and when to make recommendations. At current stage of research, several reinforcement learning (RL) based frameworks have been adopted into MCR to model the complex conversational interaction environment. For instance, EAR [13] utilizes latent vectors based on available information to capture the current state of MCR, and learns the proper timing to ask questions about attributes or to recommend. Furthermore, SCPR [15] models the MCR task as an interactive path reasoning problem on the knowledge graph (KG). It chooses attributes and items strictly following the paths, and reasons on KG to find the candidate attributes or items via user's feedback. KBQG [23] generates the clarifying questions to collect the user's preference of attribute types based on knowledge graph. UNICORN [8] proposes a unified reinforcement learning framework based on dynamic weighted graph for MCR, which unifies three decision-making processes. Moreover, some sophisticated conversational strategies try to lead dialogues [36], which can introduce diverse topics and tasks in MCR [16, 19, 30, 33, 35, 37, 44].

However, these works all ignore a more realistic scenario in which users may accept multiple items with partially overlapped attributes. Therefore, we propose a new scenario named MIMCR to fill this gap. Furthermore, we develop a novel framework namely MCMPL to tackle the existing challenges.

3 DEFINITION AND PRELIMINARY

Although the multi-round conversational recommendation (MCR) scenario [8, 13, 15] is the most realistic CRS setting proposed so far, the assumption proposed by MCR [13], that the user preserves clear preferences towards all the attributes and items, still deviates from real scenario. In this work, we assume the user's preference for items is incomplete when resorting to CRS. Specifically, the user has clear single preferences for some attribute types, while for other attribute types, his preference might be various or vague. With the guidance of CRS, he may accept multiple attribute instances with the same type, which results in that the user may show interests in over items under different attribute instance combinations. Therefore, we propose a new scenario named **Multi-Interest Multi-round Conversational Recommendation (MIMCR)**.

In this scenario, we define the sets of users and items as \mathcal{U} and \mathcal{V} , respectively. And we also separately define the sets of attribute types and instances as \mathcal{C} and \mathcal{P} . Each $v \in \mathcal{V}$ is associated with a set of attribute instances \mathcal{P}_v . Each $p \in \mathcal{P}$ has its corresponding attribute type $c_p \in \mathcal{C}$. In each episode, there is a set \mathcal{V}_u of items that are acceptable to the user $u \in \mathcal{U}$. The set is represented as follows:

$$\mathcal{V}_u = \{v_1, v_2, \dots, v_{N_u}\} \quad (1)$$

where N_u is the number of acceptable items, $\mathcal{P}_1 \cap \mathcal{P}_2 \cap \dots \cap \mathcal{P}_{N_u} = \mathcal{P}_{same} \neq \emptyset$ and $\mathcal{P}_i \neq \mathcal{P}_j$. A conversation session is initialized by user u specifying an attribute instance $p_0 \in \mathcal{P}_{same}$ he clearly prefers. Then, the agent selects to ask questions about attribute instances or to recommend items based on policy learning. The CRS will update the conversational state based on the user feedback. The process will repeat until at least one acceptable item is successfully recommended to the user or the system reaches the maximum number of turn T .

4 FRAMEWORK

We propose Multiple Choice questions based Multi-Interest Policy Learning (MCMPL), a novel framework for MIMCR. The goal of our framework is to learn the policy network $\pi(s_t|a_t)$ to maximize the expected cumulative rewards as: $\pi^* = \operatorname{argmax}_{\pi \in \Pi} \mathbb{E} [\sum_{t=0}^T r_t]$, where s_t denotes the current state, a_t denotes the action taken by the agent and the r_t is intermediate reward. On the whole, the process of our framework in one turn can be decomposed into three steps: user modeling, consultation and transition.

4.1 User Modeling

We firstly encode the state s_t , which contains all the conversational information of the prior $t-1$ turns. The current state includes six components: $s_t = \{u, \mathcal{P}_u^{(t)}, \mathcal{P}_{rej}^{(t)}, \mathcal{V}_{rej}^{(t)}, \mathcal{P}_{cand}^{(t)}, \mathcal{V}_{cand}^{(t)}\}$. Previous methods [8, 13, 15] for MCR only extract the user's interest from the current state, ignoring the complements of historical interactions to the current user's preference. To this end, we construct a current graph and a global graph to jointly learn user representations. Moreover, we develop an iterative multi-interest extractor to obtain multiple interests of the user, which will be discussed in subsection 5.1.

4.2 Consultation

Once the system finishes the user modeling step, it will move to the consultation step, with the purpose to decide whether to ask attribute instances or to recommend items. To make the next action more profitable and recommend successfully with the fewer turns, we employ a reinforcement learning (RL) method based on the extracted multiple interests of the user to learn the policy. The action space includes all candidate items and candidate attribute instances. However, in the real world, the number of items and attribute instances is very large, which severely limits the efficiency of CRS. To improve the efficiency, we sample K_v items and K_p attribute instances as action space \mathcal{A}_t . We develop a novel dueling Q-network [34] to calculate the Q-value of each action in \mathcal{A}_t . If CRS decides to ask a question, our method will select K_a attribute instances in \mathcal{A}_t with the same attribute type to generate *attribute type-based multiple choice questions*. The user can choose zero (the option "Others" as shown in conversation (b) of Figure 1), one, or more attribute instances with the given attribute type. If CRS decides to recommend items, the system will select K items in \mathcal{A}_t to recommend. We will discuss the details of sampling strategies and policy learning in subsection 5.2.

4.3 Transition

When the user responds to the action of agent, the transition step will be triggered. This step will transition the current state to the next state s_{t+1} . If the user responds to the question, attribute instance sets that the user accepts and rejects in this turn can be defined as $\mathcal{P}_{cur_acc}^{(t)}$ and $\mathcal{P}_{cur_rej}^{(t)}$ respectively. Some components are updated by $\mathcal{P}_{cand}^{(t+1)} = \mathcal{P}_{cand}^{(t)} - \mathcal{P}_{cur_rej}^{(t)} - \mathcal{P}_{cur_acc}^{(t)}$, $\mathcal{P}_{rej}^{(t+1)} = \mathcal{P}_{rej}^{(t)} \cup \mathcal{P}_{cur_rej}^{(t)}$ and $\mathcal{P}_u^{(t+1)} = \mathcal{P}_u^{(t)} \cup \mathcal{P}_{cur_acc}^{(t)}$. When the user is recommended items, if the set $\mathcal{V}_{rec}^{(t)}$ of recommended items are all rejected, the next state can be updated by $\mathcal{V}_{rej}^{(t+1)} = \mathcal{V}_{rej}^{(t)} \cup \mathcal{V}_{rec}^{(t)}$. Otherwise, this conversation session ends. Finally, we need to update the candidate item set $\mathcal{V}_{cand}^{(t+1)}$ based on the user's feedback. Previous works [8, 15]

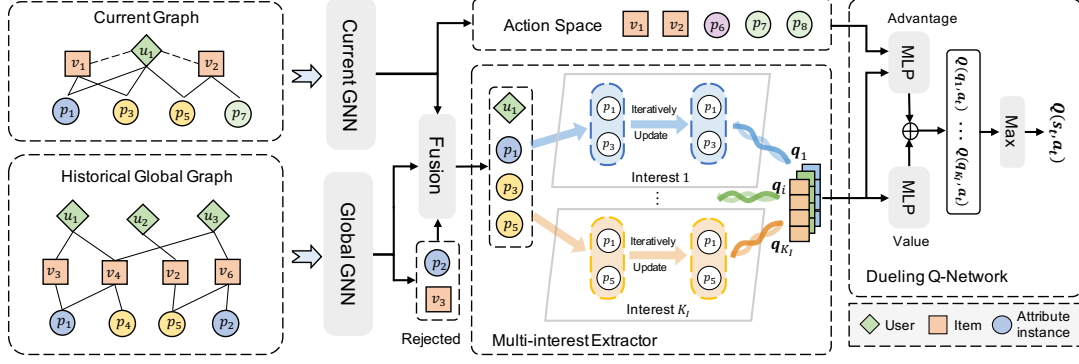


Figure 2: The overview of Multi-Interest Policy Learning (MIPL).

update candidate items based the intersection set strategy, that is, only the items satisfying all the accepted attribute instances in $\mathcal{P}_u^{(t+1)}$ remain, which obviously deviates from the scenario. In fact, the user might not prefer the combination of all attribute instances, but rather part of them. To this end, we propose the *attribute instance-based union set strategy* to update $\mathcal{V}_{cand}^{(t+1)}$ as follows:

$$\mathcal{V}_{cand}^{(t+1)} = \{v | v \in \mathcal{V}_{p_0} - \mathcal{V}_{rej}^{(t+1)} \text{ and } \mathcal{P}_v \cap \mathcal{P}_u^{(t+1)} \neq \emptyset \text{ and } \mathcal{P}_v \cap \mathcal{P}_{rej}^{(t+1)} = \emptyset\} \quad (2)$$

where \mathcal{V}_{p_0} is the item set in which all items are associated to attribute instance p_0 which initializes the conversation session. In this way, we can get the next state, which will be updated as $s_{t+1} = \{u, \mathcal{P}_u^{(t+1)}, \mathcal{P}_{rej}^{(t+1)}, \mathcal{V}_{rej}^{(t+1)}, \mathcal{P}_{cand}^{(t+1)}, \mathcal{V}_{cand}^{(t+1)}\}$.

4.4 Reward

In this work, five kinds of rewards are defined following [8, 15], namely, (1) r_{rec_suc} , a strongly positive reward when the recommendation succeeds, (2) r_{rec_fail} , a strongly negative reward when the recommendation fails, (3) r_{ask_suc} , a slightly positive reward when the user accepts an asked attribute instance, (4) r_{ask_fail} , a negative reward when the user rejects an asked attribute instance, (5) r_{quit} , a strongly negative reward if the session reaches the maximum number of turns. In addition, since our method asks multiple choice questions, we design the reward from the user's feedback on a question in the form of sum as $r_t = \sum_{\mathcal{P}_{cur_acc}^{(t)}} r_{ask_suc} + \sum_{\mathcal{P}_{cur_rej}^{(t)}} r_{ask_rej}$.

5 MULTI-INTEREST POLICY LEARNING

In this section, we detail the design of Multi-Interest Policy Learning (MIPL) module. As shown in Figure 2, to obtain more comprehensive user representations, we establish a current graph to capture user current preferences, and a global graph to capture long-term preferences. Based on the learned node representations of the two graphs, we propose an iterative multi-interest extractor to model user's preferences for different combinations of attribute instances. Moreover, we design a new duelling Q-network [34] to decide the next action based on the extracted multiple interests.

5.1 Multi-interest Encoder

5.1.1 GNN-based Representation Fusion. The existing methods [8, 13, 15] capture user preferences based on the current conversation state, which might cause user preferences to be incomplete due to the limited number of turns. In addition, only the current

conversation information is not enough to capture the correlation of attribute instances. Therefore, we construct a current graph based on the conversation state, and a historical global graph based on the historical user-item interactions and the global item-attribute instance correlations. We employ GNNs to learn the node representations of two graphs separately and utilize gating mechanism for fusion.

Current Graph Representation. Following [8], we construct a weighted graph based on the t -th turn state of an episode as $\mathcal{G}_u^{(t)} = (\mathcal{N}^{(t)}, \mathcal{E}^{(t)})$, where $\mathcal{N}^{(t)} = \{u\} \cup \mathcal{P}_u^{(t)} \cup \mathcal{P}_{cand}^{(t)} \cup \mathcal{V}_{cand}^{(t)}$. For the edge weight $\mathcal{E}_{i,j}^{(t)}$, we consider three cases: (1) The weight of edge between the user and each accepted attribute instance is 1; (2) The weight of edge between each attribute instance and the associated item is 1; (3) The weight of edge between the user and each item is $w_v^{(t)}$, which indicates the coarse matching score of the item v to the current state: $w_v^{(t)} = \sigma(\mathbf{e}_u^T \mathbf{e}_v + \sum_{p \in \mathcal{P}_u^{(t)}} \mathbf{e}_p^T \mathbf{e}_v - \sum_{p \in \mathcal{P}_{rej}^{(t)}} \mathbf{e}_p^T \mathbf{e}_v)$,

where $\sigma(\cdot)$ is the sigmoid function, \mathbf{e}_u , \mathbf{e}_v and $\mathbf{e}_p \in \mathbb{R}^d$ are the initial embedding of user, item and attribute instance.

We employ a L_c -layer GCN [10] to capture the connectivity between nodes of $\mathcal{G}_u^{(t)}$ and obtain higher-quality node representations in the current state. We define the initial embedding \mathbf{e}_n of node n as $\mathbf{e}_n^{(0)}$, and $\mathbf{e}_n^{(l)}$ as the output node embedding of l -th layer. The calculation method of $l+1$ -th layer is as follows:

$$\mathbf{e}_n^{(l+1)} = \text{ReLU}(\sum_{j \in \mathcal{N}_n^{(t)}} \frac{\mathbf{W}_c^{(l+1)} \mathbf{e}_j^{(l)}}{\sqrt{\sum_i \mathcal{E}_{n,i}^{(t)} \sum_i \mathcal{E}_{j,i}^{(t)}}} + \mathbf{e}_n^{(l)}) \quad (3)$$

where $\mathcal{N}_n^{(t)}$ denotes the set of neighbor nodes of node n in the turn t , $\mathbf{W}_c^{(l+1)} \in \mathbb{R}^{d \times d}$ is trainable parameters. We define the output of the last layer $\mathbf{e}_n^{(L_c)}$ as the final embedding \mathbf{e}_n^c of the node.

Global Graph Representation. We use the historical interactions between users and items as well as the correlation between items and attribute instances to establish a heterogeneous global graph $\mathcal{G}_g = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \mathcal{U} \cup \mathcal{V} \cup \mathcal{P}$ and $\mathcal{E} = \mathcal{E}_{u,v} \cup \mathcal{E}_{p,v}$. The edge $(u, v, r_{u \sim v}) \in \mathcal{E}_{u,v}$ denotes the user u has interacted the item v . And the edge $(p, v, r_{p \sim v}) \in \mathcal{E}_{p,v}$ denotes that the item v is associated with the attribute instance p .

We employ a L_g -layer Global Graph Neural Network (GGNN) [5, 6, 25] to extract long-term interests of users, and global correlations of items and attribute instances. The initial input embeddings of the first layer are $\mathbf{s}_u^{(0)} = \mathbf{e}_u$, $\mathbf{s}_v^{(0)} = \mathbf{e}_v$ and $\mathbf{s}_p^{(0)} = \mathbf{e}_p$. Let $\mathbf{s}_u^{(l)}$, $\mathbf{s}_v^{(l)}$ and $\mathbf{s}_p^{(l)}$ denote the output representations of nodes after the propagation

of l -th layer. For the $l + 1$ -th layer of GGNN, we model different edge types separately. For the edge in $\mathcal{E}_{u,v}$, we adopt the calculation method as follow:

$$s_{u \sim v}^{(l+1)}(n) = b_g^{(l+1)} + \sum_{i \in \mathcal{N}_{r_{u \sim v}}(n)} \frac{\mathbf{W}_g^{(l+1)} s_i^{(l)}}{\sqrt{|\mathcal{N}_{r_{u \sim v}}(i)| |\mathcal{N}_{r_{u \sim v}}(n)|}} \quad (4)$$

where $\mathcal{N}_{r_{u \sim v}}(n)$ denotes the neighbor nodes of node n with the edge type $r_{u \sim v}$, $\mathbf{W}_g^{(l+1)}$ and $b_g^{(l+1)}$ are trainable parameters. For the edge in $\mathcal{E}_{p,v}$, we adopt the same method as Equation 4 to get $s_{p \sim v}^{(l+1)}(n)$.

For the user u and attribute instance p , we utilize ReLU function to activate semantic messages to obtain output node embeddings: $s_u^{(l+1)} = \text{ReLU}(s_{u \sim v}^{(l+1)}(u))$, $s_p^{(l+1)} = \text{ReLU}(s_{p \sim v}^{(l+1)}(p))$. Since item v is connected by both two kinds of edges, we accumulate different messages propagated by different types of edges and update the representation: $s_v^{(l+1)} = \text{ReLU}(\text{mean}(s_{u \sim v}^{(l+1)}(v), s_{p \sim v}^{(l+1)}(v)))$. We define the output of the last layer $s_n^{(L_g)}$ as the final node embedding s_n^g .

We apply the gating mechanism to fuse the embeddings of nodes which belong to both graphs $\mathcal{G}_u^{(t)}$ and \mathcal{G}_g as follows:

$$g = \sigma(\mathbf{W}_{gated} [\mathbf{e}_n^c \parallel \mathbf{s}_n^g]), \mathbf{v}_n = g \cdot \mathbf{s}_n^g + (1 - g) \cdot \mathbf{e}_n^c, \quad (5)$$

where \parallel is the concatenate operation, $\mathbf{W}_{gated} \in \mathbb{R}^{d \times d}$ is trainable parameter and $\sigma(\cdot)$ is the sigmoid function.

5.1.2 Iterative Multi-interest Extractor. In CRS scenario, since the user's interest is diversity, we use multi-attention mechanism to model the user u and attribute instances accepted by u . The multi-interest embeddings of user can be obtained through the combination of attribute instances with different weights. Inspired by [3, 24, 32], we adopt the iterative update rule to adjust the weights of attribute instances with M iterations more precisely.

Previous works rarely consider items or attribute instances rejected by users, which can complement the current preferences of the user effectively. Therefore, we first fuse the global embeddings of the rejected items and attribute instances with the user's embedding:

$$\hat{\mathbf{v}}_u = \mathbf{v}_u + \mathbf{W}_u \left(\frac{1}{|\mathcal{N}_{rej}|} \sum_{n \in \mathcal{N}_{rej}} s_n^g \right) \quad (6)$$

where $\mathcal{N}_{rej} = \mathcal{V}_{rej} \cup \mathcal{P}_{rej}$, and $\mathbf{W}_u \in \mathbb{R}^{d \times d}$ is trainable parameters. Then, we define K_I attention networks for K_I interests. Based on accepted attribute instance embeddings $[\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N]$ and user embedding $\hat{\mathbf{v}}_u$, the initial iteration calculation method of each attention network to obtain the interest embedding $\mathbf{q}_k^{(1)}$ is as follows:

$$\mathbf{q}_k^{(1)} = \sum_{n=1}^N \alpha_{k,n}^{(1)} \mathbf{v}_n, k \in \{1, \dots, K_I\} \quad (7)$$

$$\alpha_{k,n}^{(1)} = \frac{\exp(\mathbf{h}_k^T \sigma(\mathbf{W}_k(\hat{\mathbf{v}}_u \parallel \mathbf{v}_n)))}{\sum_{n'=1}^N \exp(\mathbf{h}_k^T \sigma(\mathbf{W}_k(\hat{\mathbf{v}}_u \parallel \mathbf{v}_{n'})))} \quad (8)$$

where \mathbf{h}_k and \mathbf{W}_k are trainable metrics. The m -th iteration precisely adjusts the weights $\alpha_{k,n}^{(m)}$ based on the $m - 1$ -th iteration results:

$$\mathbf{q}_k^{(m)} = \sum_{n=1}^N \alpha_{k,n}^{(m)} \mathbf{v}_n \quad (9)$$

$$\alpha_{k,n}^{(m)} = \frac{\exp(\mathbf{h}_k^T \sigma(\mathbf{W}_k(\mathbf{q}_k^{(m-1)} \parallel \mathbf{v}_n))) + \alpha_{k,n}^{(m-1)}}{\sum_{n'=1}^N \exp(\mathbf{h}_k^T \sigma(\mathbf{W}_k(\mathbf{q}_k^{(m-1)} \parallel \mathbf{v}_{n'}))) + \alpha_{k,n'}^{(m-1)}} \quad (10)$$

where \mathbf{h}_k and \mathbf{W}_k are parameters shared with the previous iterations. We define the output $\{\mathbf{q}_1^{(M)}, \mathbf{q}_2^{(M)}, \dots, \mathbf{q}_{K_I}^{(M)}\}$ of M -th iteration as the final multi-interest embeddings $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{K_I}\}$.

5.2 Action Decision Policy Learning

A large action search space will bring a great negative impact on the efficiency of the system. Following [8], we select K_v items and K_p attribute instances as candidate action space \mathcal{A}_t . For candidate items to be recommended, we consider how well they match the current state. We select top- K_v items into the action space based on $w_v^{(t)}$. For attribute instances, we also select top- K_p attribute instances based on $w_p^{(t)}$ as: $w_p^{(t)} = \sigma(\mathbf{e}_u^T \mathbf{e}_p + \sum_{p' \in \mathcal{P}_u^{(t)}} \mathbf{e}_p^T \mathbf{e}_{p'} - \sum_{p \in \mathcal{P}_{rej}^{(t)}} \mathbf{e}_p^T \mathbf{e}_{p'})$

Inspired by [8], we design an improved dueling Q-network [34] to determine the next action. Following the standard assumption that delayed rewards are discounted by a factor of γ per timestep, we define the Q-value $Q(s_t, a_t)$ as the expected reward based on the state s_t and the action a_t . Based on the obtained K_I interest representations according to the current state s_t , we calculate each score between action a_t and each interest, and take the maximum value as Q-value:

$$Q(s_t, a_t) = \max_k (f_{\theta_V}(\mathbf{q}_k) + f_{\theta_A}(\mathbf{q}_k, a_t)), k \in \{1, \dots, K_I\} \quad (11)$$

where $f_{\theta_V}(\cdot)$ and $f_{\theta_A}(\cdot)$ are separate multi-layer perceptions (MLP). The optimal Q-function $Q^*(s_t, a_t)$ achieves the maximum expected reward by the optimal policy π^* , following the Bellman [1] equation:

$$Q^*(s_t, a_t) = \mathbb{E}_{s_{t+1}} \left[r_t + \gamma \max_{a_{t+1} \in \mathcal{A}_{t+1}} Q^*(s_{t+1}, a_{t+1} | s_t, a_t) \right] \quad (12)$$

The CRS firstly selects the action with the max Q-value. If the selected action points to an item, the system will recommend top- K items with the highest Q-value to the user. If the selected action points to an attribute instance p , the system will generate *attribute type-based multiple choice questions* to ask user. To be specific, the system will decide a attribute type c , and select top- K_a attribute instances whose corresponding attribute type is c with the highest Q-value. Then the user can choose which of the attribute instances he likes or dislikes. We propose two strategies to decide the attribute type c : (1) Top-based strategy. We select the attribute type corresponding to the attribute instance with the highest Q-value. (2) Sum-based strategy. For each attribute type, we sum the Q-values of its corresponding attribute instances to obtain the attribute type level score, and select the attribute type with the highest score. During the experiment, we mainly use Top-based strategy, and the other strategy will be compared in the ablation study.

5.3 Model Training

For each turn, the agent will receive the reward r_t based on the user's feedback. According to user feedback, we can update the state s_{t+1} and action space \mathcal{A}_{t+1} . We define a replay buffer \mathcal{D} following [8], which stores the experience $(s_t, a_t, r_t, s_{t+1}, \mathcal{A}_{t+1})$. To train our model, we sample mini-batch experiences from the replay buffer \mathcal{D} and define a loss function as follows:

$$\mathcal{L} = \mathbb{E}_{(s_a, a_t, r_t, s_{t+1}, \mathcal{A}_{t+1}) \sim \mathcal{D}} [(y_t - Q(s_t, a_t; \theta_Q, \theta_M))^2] \quad (13)$$

where θ_M is the set of parameters to capture multi-interest embeddings, $\theta_Q = \{\theta_V, \theta_A\}$, and y_t is the target value, which is based on the optimal Q-function as follows:

$$y_t = r_t + \gamma \max_{a_{t+1} \in \mathcal{A}_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_Q, \theta_M) \quad (14)$$

Due to the overestimation bias in original DQN, we employ the double DQN [29] to copy a target network Q' as a periodic from the online network to train the model following [8, 45].

6 EXPERIMENTS

To fully demonstrate the superiority of our method, we conduct experiments to verify the following four research questions (RQ):

- **(RQ1)**: Compared with the state-of-the-art methods, does our framework achieve better performance?
- **(RQ2)**: What are the impacts of key components on performance?
- **(RQ3)**: How do the settings of hyper-parameters (such as the number of interests K_I) affect our framework?
- **(RQ4)**: How can our framework effectively extract multiple interests in different attribute instance combinations?

Table 1: Statistics of datasets.

Dataset	Yelp	LastFM	Amazon-Book	MovieLens
#Users	27,675	1,801	30,291	20,892
#Items	70,311	7,432	17,739	16,482
#Interactions	1,368,609	76,693	478,099	454,011
#Attribute instances	590	8,438	988	1,498
#Attribute types	29	34	40	24
#Entities	98,576	17,671	49,018	38,872
#Relations	3	4	2	2
#Triplets	2,533,827	228,217	565,068	380,016

6.1 Datasets

To evaluate the proposed method, we adapt two existing MCR benchmark datasets, named Yelp and LastFM. To evaluate our method more comprehensively, we also process two additional datasets. The statistics of these datasets are presented in Table 1.

- **Yelp and LastFM** [13]: For the Yelp, Lei et al. build a 2-layer taxonomy. We define the 29 first-layer categories as attribute types, and 590 second-layer categories as attribute instances. For the LastFM, we adopt original attributes as attribute instances. We utilize clustering to select 34 categories as attribute types.
- **Amazon-Book** [31]: We select entities and relations in knowledge graph (KG) as attribute instances and types, separately. To ensure data quality, we select entities associated with at least 10 items.
- **MovieLens**: MovieLens-20M¹ is a widely used recommendation benchmark dataset. We retain the user-item interactions with the rating > 3 . Similarly, we select entities in KG as attribute instances and relations as attribute types.

For each conversation episode, we sample N_v items with partially overlapped attribute instances as the acceptable items for the user.

6.2 Experiments Setup

6.2.1 User Simulator. Since MCR is a system based on interaction with users, we design a user simulator to train and evaluate it. Based on the scenario MIMCR, we adjust the user simulator adopted in [13]. We simulate a conversation session for each observed user-item set interaction pair (u, \mathcal{V}_u) . We regard each item $v_i \in \mathcal{V}_u$ as the ground-truth target item. The session is initialized by the simulated user specifying an attribute instance $p_0 \in \mathcal{P}_{same}$. Given a conversation, the simulated user's feedback of each turn follows the rules: (1) when the system asks a question, he will accept the attribute instances which are associated with any item in \mathcal{V}_u and reject others; (2) when the system recommends a list of items, he will accept it if the list contains at least one item in \mathcal{V}_u ; (3) We consider that user's patience will run out when the maximum number of turn T is reached. The simulated user will exit the system until he accepts the recommended item list or his patience runs out.

6.2.2 Baselines. To evaluate model performance, we compare our model with following six representative baselines:

- **Max Entropy** asks an attribute or recommends the top ranked items based on a certain probability [13].
- **Abs Greedy** [7] only recommends items in each turn and treats rejected items as negative examples to update the model.
- **CRM** [28] is originally designed for single-round CRS, which utilizes reinforcement learning to select next action. Following [13], we adapt CRM to MCR scenario.
- **EAR** [13] adopts a three stage solution called Estimation–Action–Reflection for MCR, and employs RL strategy to decide actions.
- **SCPR** [15] proposes a generic framework that models MCR as an interactive path reasoning problem on a graph, and employs the DQN [20] framework to select actions.
- **UNICORN** [8] proposes a unified policy learning framework, which develops a dynamic graph based RL to select action for each turn. It is the state-of-the-art (SOTA) method.

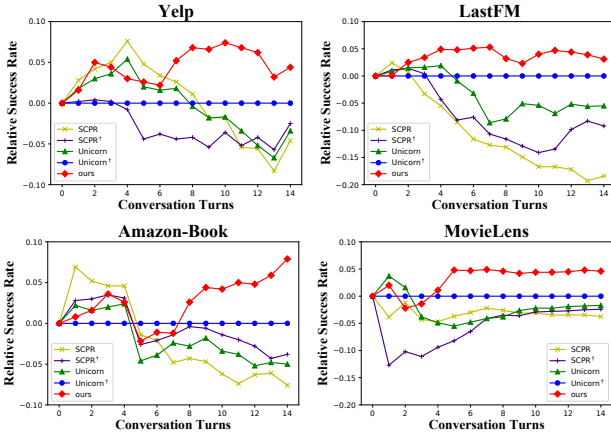
For a more comprehensive and fair performance comparison, we adapt SCPR and UNICORN as follows: (1) The system employs multiple choice questions to ask the user. When the system decides to ask the user, the agent will generate *attribute type-based multiple choice questions* as described in subsection 5.2. (2) The system selects candidate item set by the *attribute instance-based union set strategy* described in subsection 4.3. We name the two adapted methods SCPR[†] and UNICORN[†] respectively.

6.2.3 Parameters Setting. We randomly split each dataset for training, validation and test with the ratio of 7 : 1.5 : 1.5. The embedding dimension is set as 64, while the batch size as 128. We recommend top $K = 10$ items or ask $k_a = 2$ attribute instances in each turn. The maximum turn T of conversation is set as 15. We employ the Adam optimizer with the learning rate $1e - 4$. Discount factor γ is set to be 0.999. Following [8], we adopt TransE [2] via OpenKE [9] to pretrain the node embeddings in the constructed KG with the training set. We construct the global graph based on the training set. The numbers of current GNN layers L_c and global GNN layers L_g are set to be 2 and 1, respectively. We extract user's multiple interests with $M = 2$ iterations. For the action space, we select $K_p = 10$ attribute instances and $K_o = 10$ items. To maintain a fair comparison, we adopt the same reward settings: $r_{rec_suc} =$

¹<https://grouplens.org/datasets/movielens/>

Table 2: Performance comparison of different models on the four datasets. hDCG stands for hDCG@(15, 10).

Models	Yelp			LastFM			Amazon-Book			MovieLens		
	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG
Abs Greedy	0.195	14.08	0.069	0.539	10.92	0.251	0.214	13.50	0.092	0.752	4.94	0.481
Max Entropy	0.375	12.57	0.139	0.640	9.62	0.288	0.343	12.21	0.125	0.704	6.93	0.448
CRM	0.223	13.83	0.073	0.597	10.60	0.269	0.309	12.47	0.117	0.654	7.86	0.413
EAR	0.263	13.79	0.098	0.612	9.66	0.276	0.354	12.07	0.132	0.714	6.53	0.457
SCPR	0.392	12.65	0.140	0.659	9.36	0.307	0.390	11.72	0.144	0.799	4.39	0.529
UNICORN	0.404	12.39	0.146	0.788	7.56	0.355	0.416	11.68	0.155	0.819	4.28	0.568
SCPR [†]	0.413	12.45	0.149	0.751	8.52	0.339	0.428	11.50	0.159	0.812	4.03	0.547
UNICORN [†]	0.438	12.28	0.151	0.843	7.25	0.363	0.466	11.24	0.170	0.836	3.82	0.576
Our Model	0.482	11.87	0.160	0.874	6.35	0.396	0.545	10.83	0.223	0.882	3.61	0.599

**Figure 3: Comparisons at Different Conversation Turns.**

1, $r_{rec_fail} = -0.1$, $r_{ask_suc} = 0.01$, $r_{ask_fail} = 0.1$, $r_{quit} = -0.3$. We set the maximum number N_o of acceptable items as 2. Other settings are explored in the hyper-parameter analysis.

6.2.4 Evaluation Metrics. Following previous studies on MCR [8, 13, 15], we utilize success rate (SR@ T) [28] to measure the cumulative ratio of successful recommendation with the maximum turn T , and average turn (AT) to evaluate the average number of turns. Besides, we adopt hDCG@(T, K) [8] to additionally evaluate the ranking performance of recommendations. For SR@ t and hDCG@(T, K), the higher value indicates better performance. While the lower AT means the overall higher efficiency.

6.3 Performance Comparison (RQ1)

The comparison experimental results of the baseline models and our models are shown in Table 2. We also intuitively present the performance comparison of success rate at each turn in Figure 3. Relative success rate denotes the difference between each methods and the most competitive baseline UNICORN[†], where the blue line of UNICORN[†] is set to $y = 0$ in the figures. For clear observation, we only report the result of four competitive baselines and our model. Based on the comparison in the table and figures, we can summarize our observations as follows:

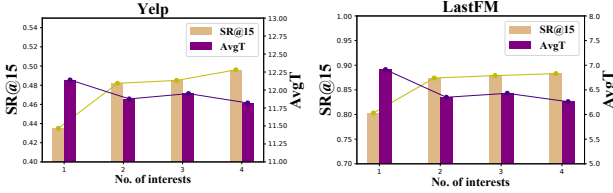
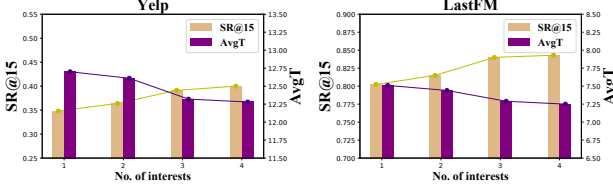
- Our framework outperforms all the comparison methods on four datasets. Compared with baselines, our method extends the form

Table 3: Results of the Ablation Study.

Models	Yelp			Amazon-Book		
	SR@15	AT	hDCG	SR@15	AT	hDCG
Ours	0.482	11.87	0.160	0.545	10.83	0.223
-w/o multi-interest	0.435	12.41	0.145	0.522	10.96	0.204
-w/o global graph	0.463	12.31	0.150	0.516	11.03	0.198
-binary questions	0.448	12.96	0.151	0.513	11.12	0.192
-intersection set strategy	0.414	12.29	0.145	0.438	11.81	0.159
-Sum-based strategy	0.467	11.94	0.152	0.529	11.01	0.217

of questions to attribute type-based multiple choice formula, eliciting user’s feedback of multi-acceptable items efficiently. Besides, the union set strategy can effectively avoid over-filtering items. Moreover, we extract multiple interests of the user from the accepted attribute instances by combining current preferences with historical interactions, instead of utilizing a mixed single state representation to decide the next action.

- Compared to the original version of SCPR and UNICORN, adapted SCPR[†] and UNICORN[†] achieve better performance, which indicates the effectiveness of above designs (multiple choice questions and union set strategy) for MIMCR. Nevertheless, our method still outperforms the adapted methods. We infer that the single user preference extracted by these baselines limits the ability to capture fine-grained user interests.
- Interestingly, we can find that original SCPR and UNICORN outperform adapted versions at the first few turns, but they fall quickly as the turn increases. Since original frameworks narrow the candidate item set following the intersection set strategy, and the acceptable items might not be filtered out when the number of accepted attribute instances is small, a smaller candidate item set can increase the probability of successful recommendation. As the number of conversations turn grows, the over-specific candidate item set over-filters out the acceptable items, which limits the subsequent improvement of these methods. On the contrary, our method achieves an outstanding performance in the latter stage of the conversation, where there are still comparatively generalized candidate items set and attributes space to avoid over-filtering.

Figure 4: Performance comparisons w.r.t. K_I with $N_0 = 2$.Figure 5: Performance comparisons w.r.t. K_I with $N_0 = 3$.

6.4 Ablation Studies (RQ2)

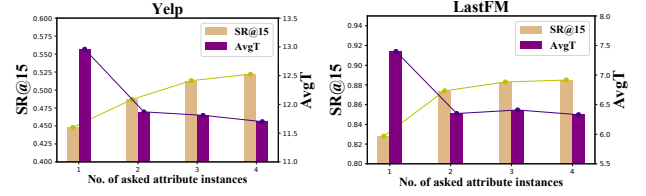
In order to verify the effectiveness of some key designs, we conduct a series of ablation experiments on the Yelp and Amazon-Book datasets. The results are shown in Table 3.

6.4.1 Impact of different modules. Firstly, we evaluate the effectiveness of different modules, including Iterative Multi-interest Extractor and Global Graph Representation. Specifically, we remove these critical modules of MCM IPL to observe performance changes. As can be seen in Table 3, the model performance decreases significantly without the Iterative Multi-interest Extractor, which suggests that multi-interest representation is more appropriate for MIMCR, compared to the mixed single-interest representation. Moreover, we can see that the removal of Global Graph Representation module also leads to poor performance, which indicates that the historical user representation is important for revealing latent user preferences and guiding the extraction of current multiple interests.

6.4.2 Impact of different strategies. We conduct some experiments to study the effectiveness of strategies. Specifically, we retain the binary question type ("-binary questions"), traditional candidate item filtering strategy ("-intersection set strategy"), separately. Meanwhile, we utilize the Sum-based strategy to decide the attribute type involved in questions. The binary question type version of our model performs worse than default setting, which demonstrates the efficiency of multiple choice question types for the conversational interaction. Besides, the intersection set strategy achieves inferior performance. It can be inferred that limitation of item selection strategy based on all accepted attribute instances will over-filter some user-acceptable items. While for the adjustment of sum-based strategy, the model still keeps competitive performance in all metrics for MIMCR, which indicates that this strategy can select suitable attribute types based on user interests.

6.5 Hyper-parameter Sensitivity Analysis (RQ3)

6.5.1 Impact of Interests Number. Since interest number K_I is closely related to maximum number N_0 of acceptable items. We explore the hyper-parameter K_I in the case of the maximum number N_0 of acceptable items is 2 and 3 respectively. As we can see from Figure 4 and Figure 5, with the increase of interest number K_I , the performance of our methods improves. In addition, when the interest

Figure 6: Performance comparisons w.r.t. K_a .

Acceptable Movies	Genres	Actors	Directors
Deadpool	comedy, action	Ryan Reynolds	Tim Miller
Forrest Gump	comedy, romance	Tom Hanks, Gary Sinise	Robert Zemeckis

Conversation	Multiple Interests
Hi! I'm looking for some American movies.	-
Which genres do you like? (A) Comedy (B) Action (C) Others	$I_1 = \{\text{American, Comedy}\}$ $I_2 = \{\text{American, Action}\}$
Which actors do you like? (A) Ryan Reynolds (B) Tom Hanks (C) Others	$I_1 = \{\text{American, Comedy, Tom Hanks}\}$ $I_2 = \{\text{Action, Ryan Reynolds, Tom Hanks}\}$
Which directors do you like? (A) Robert Zemeckis (B) James Cameron (C) Others	$I_1 = \{\text{Comedy, Tom Hanks, Robert Zemeckis}\}$ $I_2 = \{\text{Action, Ryan Reynolds, Tom Hanks}\}$
How about "Forrest Gump"?	I_1 hits "Forrest Gump"

Figure 7: A conversation generated by our framework. I_1 and I_2 denote two interests of the user, respectively.

number K_I exceeds the maximum number of acceptable items, the performance will hardly improve, which indicates that some interests may exist redundancy and point to the same user preferences.

6.5.2 Impact of Asked Attribute Instances Number. When asking users questions, the attribute instances number K_a included in a question affects model performance. As can be seen in Figure 6, the performance improves as the value of K_a increases, which indicates that the larger number of asked attribute instances in a turn, the more information the CRS obtains. However, if the value of K_a is too large, performance improvement is limited. That also indicates the most of extra attribute instances are invalid.

6.6 Case Study (RQ4)

To show the process of extracting the user's multiple interests, we present a conversation case generated by our framework from MovieLens dataset in Figure 7. We only show the attribute types and instances that are relevant to the questions. For each interest, we present attribute instances with high contribution rate, where the sum of their attention scores ≥ 0.8 . As can be seen, based on user's feedback of each turn as well as historical global information, our model extracts multiple interests in different attribute instances combinations. Finally, our method makes a successful recommendation based on one of the interest representations that perfectly matches user's preference.

7 CONCLUSION

In this work, we define a more realistic CRS scenario named MIMCR, in which the user may accepts one of multiple potential items instead of single target item in a conversation. Based on the scenario, we propose a novel framework MCM IPL, which generates multiple choice questions to collect user preferences, and utilizes union set strategy to select candidate items. In addition, we propose a MIPL module to exact multi-interest of the user to decide the next action. Extensive experimental results on four datasets demonstrate the superiority of our method in the proposed scenario.

ACKNOWLEDGMENTS

The work is partially supported by the National Nature Science Foundation of China (No. 61976160, 61976158, 61906137), Shanghai Science and Technology Plan Project (No. 21DZ1204800) and Technology research plan project of Ministry of Public and Security (Grant No. 2020JSYD01).

REFERENCES

- [1] Richard Bellman and Robert Kalaba. 1957. On the role of dynamic programming in statistical communication theory. *IRE Transactions on Information Theory* 3, 3 (1957), 197–203.
- [2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).
- [3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.
- [4] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 1803–1813.
- [5] Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement learning based graph-to-sequence model for natural question generation. In *ICLR*.
- [6] Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2020. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings. In *NeurIPS*.
- [7] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 815–824.
- [8] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified Conversational Recommendation Policy Learning via Graph-based Reinforcement Learning. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1431–1441.
- [9] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*. 139–144.
- [10] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [11] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsook Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1073–1082.
- [12] Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2020. Conversational recommendation: Formulation, methods, and evaluation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2425–2428.
- [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 *Proceedings of the 13th International Conference on Web Search and Data Mining*. 304–312.
- [14] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1437–1447.
- [15] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2073–2083.
- [16] Mike Lewis, Denis Yarats, Yann N. Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or No Deal? End-to-End Learning of Negotiation Dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2443–2453.
- [17] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems*, Vol. 31. 9725–9735.
- [18] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2021. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-start Users. *ACM Transactions on Information Systems* 39, 4 (2021), 1–29.
- [19] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards Conversational Recommendation over Multi-Type Dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1036–1049.
- [20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
- [21] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2021. Heterogeneous Global Graph Neural Networks for Personalized Session-based Recommendation. *arXiv preprint arXiv:2107.03813* (2021).
- [22] Bilih Priyogi. 2019. Preference elicitation strategy for conversational recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 824–825.
- [23] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to Ask Appropriate Questions in Conversational Recommendation. *arXiv preprint arXiv:2105.04774* (2021).
- [24] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 3859–3869.
- [25] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.
- [26] Anna Sepiarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 172–180.
- [27] Qi Shen, Lingfei Wu, Yitong Pang, Yiming Zhang, Zhihua Wei, Fangli Xu, and Bo Long. 2021. Multi-behavior Graph Contextual Aware Network for Session-based Recommendation. *arXiv preprint arXiv:2109.11903* (2021).
- [28] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st international acm SIGIR conference on research and development in information retrieval*. 235–244.
- [29] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.
- [30] Lingzhi Wang, Huang Hu, Lei Sha, Can Xu, Kam-Fai Wong, and Daxin Jiang. 2021. Finetuning Large-Scale Pre-trained Language Models for Conversational Recommendation with Knowledge Graph. *arXiv preprint arXiv:2110.07477* (2021).
- [31] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 950–958.
- [32] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.
- [33] Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for Good: Towards a Personalized Persuasive Dialogue System for Social Good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5635–5649.
- [34] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1995–2003.
- [35] Chi-Man Wong, Fan Feng, Wen Zhang, Chi-Man Vong, Hui Chen, Yichi Zhang, Peng He, Huan Chen, Kun Zhao, and Huajun Chen. 2021. Improving Conversational Recommendation System by Pretraining on Billions Scale of Knowledge Graph. *arXiv preprint arXiv:2104.14899* (2021).
- [36] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive Human-Machine Conversation with Explicit Conversation Goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3794–3804.
- [37] Wei Wu and Rui Yan. 2019. Deep Chat-Chat: Deep Learning for Chatbots. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1413–1414.
- [38] Zhihui Xie, Tong Yu, Canzhe Zhao, and Shuai Li. 2021. Comparison-based Conversational Recommender System with Relative Bandit Feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1400–1409.
- [39] Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, and Ji-Rong Wen. 2021. Adapting User Preference to Online Feedback in Multi-round Conversational Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 364–372.
- [40] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.

- [41] Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Ethan Chang, and Bo Long. 2021. Graph Learning Augmented Heterogeneous Graph Neural Network for Social Recommendation. *arXiv preprint arXiv:2109.11898* (2021).
- [42] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the 22nd ACM international conference on Information and Knowledge Management*. 1411–1420.
- [43] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1006–1014.
- [44] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System.. In *Proceedings of the 28th International Conference on Computational Linguistics*. 4128–4139.
- [45] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. 2020. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 179–188.
- [46] Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 881–890.
- [47] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation. In *WSDM*. 816–824.
- [48] Lixin Zou, Long Xia, Yulong Gu, Xiangyu Zhao, Weidong Liu, Jimmy Xiangji Huang, and Dawei Yin. 2020. Neural Interactive Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 749–758.