Towards Hierarchical Policy Learning for Conversational Recommendation with Hypergraph-based Reinforcement Learning

Sen Zhao[a,b], Wei Wei *[a,b], Yifan Liu[a,b], Ziyang Wang[a,b], Wendi Li[a,b], Xianling Mao[c], Shuai Zhu[d], Minghui Yang[d], and Zujie Wen[d]

[a]Cognitive Computing and Intelligent Information Processing (CCIIP) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology
[b]Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL)
[c]Beijing Institute of Technology
[d]Ant Financial Services Group

Abstract

Conversational recommendation systems (CRS) aim to timely and proactively acquire user dynamic preferred attributes through conversations for item recommendation. In each turn of CRS, there naturally have two decision-making processes with different roles that influence each other: 1) **director**, which is to select the follow-up **option** (*i.e.,* ask or recommend) that is more effective for reducing the action space and acquiring user preferences; and 2) **actor**, which is to accordingly choose **primitive actions** (*i.e.,* asked attribute or recommended item) that satisfy user preferences and give feedback to estimate the effectiveness of the director's option. However, existing methods heavily rely on a unified decision-making module or heuristic rules, while neglecting to distinguish the roles of different decision procedures, as well as the mutual influences between them. To address this, we propose a novel **D**irector-**A**ctor **H**ierarchical **C**onversational **R**ecommender (DAHCR), where the director selects the most effective option, followed by the actor accordingly choosing primitive actions that satisfy user preferences. Specifically, we develop a dynamic hypergraph to model user preferences and introduce an intrinsic motivation to train from weak supervision over the director. Finally, to alleviate the bad effect of model bias on the mutual influence between the director and actor, we model the director's option by sampling from a categorical distribution. Extensive experiments demonstrate that DAHCR outperforms state-of-the-art methods.

1 Introduction

Conversational recommendation systems (CRS) aim to dynamically learn the user's preferences by iteratively interacting with the user. Existing works have explored various settings of conversational recommendation from the perspective of either dialogue systems [Li *et al.*, 2018] or recommendation systems [Lei *et al.*, 2018]. In this work, we focus on the setting of multi-round conversational recommendation (MCR) [Sun and Zhang, 2018], which aims to recommend the target item to the user by iteratively asking attributes and recommending items in the limited turns.

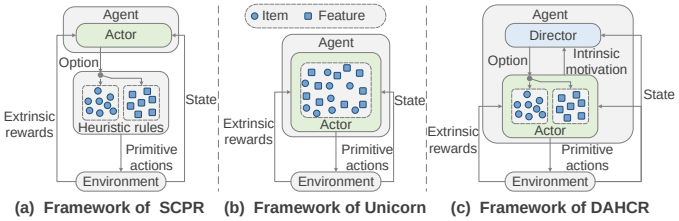

Figure 1: Illustration of policy learning frameworks for CRS, including a framework of the outsourcing strategy (SCPR), a framework of the unified strategy (Unicorn), and our proposed DAHCR with the Director-Actor framework.

For each turn in CRS, the system naturally includes two essential decision-make procedures, when to recommend (*i.e.,* ask or recommend), and what to talk about (*i.e.,* the specific attribute/items). Early works [Lei *et al.*, 2020a; Sun and Zhang, 2018] develop policy learning for a subset of decision procedures and outsource the other procedures to heuristic rules (SCPR as illustrated in Figure 1 (a)). These works isolate strategies for different decisions and make policy learning hard to converge due to their lack of mutual influence during training. To solve this problem, Deng *et al.* [2021] and Zhang *et al.* [2022] develop unified policy learning frameworks (Unicorn as illustrated in Figure 1 (b)) which unify the aforementioned two separated decision-make processes as a selection from the action space consisting of items and attributes. Despite effectiveness, the unified strategy brings out issues to be solved: (i) The unified strategy complicates the action selection of the CRS strategy by enlarging the action space and introducing data bias into the action space due to the imbalance in the number of items and attributes. As illustrated in Figure 1 (b), the action space is enlarged with all the items and attributes, and the strategy will prefer to select items when the number of items is larger. (ii)

*Corresponding author: weiw@hust.edu.cn

The unified strategy ignores the different roles of the two decision procedures, leading to the sub-optimal CRS strategy.

In the real scenario of CRS, the two decision procedures have different roles which are mutually influenced. As illustrated in Figure 1 (c), the decision procedure of when to recommend works as a **director**, which should select the option (*i.e.,* ask or recommend) that is more effective for reducing the action space and acquiring user preferences (*e.g.,* avoid recommending when the user's preference is not certain enough) to guide the latter procedure. The latter procedure works as an **actor**, which should accordingly choose the primitive action (*i.e.,* the specific attribute/items) that satisfies the user's preference and gives feedback to evaluate the effectiveness of the director's option. The director's option limits the actor's action space to either attributes or items, which reduces the action space and avoids the data bias introduced by the imbalance in the number of items and attributes.

There remain three challenges in modeling these two roles and their mutual influence. The first challenge is weak supervision. The extrinsic rewards from the environment in each turn estimate the user's preference for the actor's primitive actions, but fail to estimate the effectiveness of the director's option, which is weakly supervised by the final-turn result (*i.e.,* success or failure). The second challenge is user preference modeling. In the scenario of CRS, the user likes/dislikes items since they satisfy some attributes, which is a three-order relation (*i.e.,* user-attribute-item). To specify the attributes that motivate the user to like/dislike the item, we should model user preferences with such high-order relations. The third challenge is the bad effect of model bias [Battaglia *et al.*, 2018; Tarvainen and Valpola, 2017] on the mutual influence between director and actor. Specifically, the director's bias may lead to bad options that will filter out more efficient actions for the actor. And the actor's bias can result in false feedback that will disturb the convergence of the director.

To overcome the aforementioned challenges, we propose a **D**irector-**A**ctor **H**ierarchical **C**onversational **R**ecommender (DAHCR) with the director to select the option (*i.e.,* ask or recommend) that is more effective for reducing the action space and acquiring user preferences, followed by the actor accordingly choosing primitive actions (*i.e.,* specific items/attributes) that satisfy user preferences. To train from weak supervision over the director's option effectiveness, we develop and introduce an intrinsic motivation (*i.e.,* the actor's feedback) [Chentanez *et al.*, 2004] into our Director-Actor framework to estimate the effectiveness of the director's options. Furthermore, to model user preferences, we develop a dynamic hypergraph [Feng *et al.*, 2019] with each high-order relation (*i.e.,* user-attribute-item) specifying an attribute that motivates the user to like/dislike the item. Finally, to alleviate the bad effect of the model bias in the mutual influence, we model the director's option by sampling from a categorical distribution with Gumbel-softmax [Pei *et al.*, 2022]. Extensive experiments on two real-world datasets show that our method can outperform the state-of-the-art methods.

In a nutshell, this work makes the following contributions:

- We emphasize the roles of the Director and Actor in the two decision procedures for CRS, and the mutual influence between them.
- We propose a novel Director-Actor Hierarchical conversational recommender with intrinsic motivation to train from weak supervision and a dynamic hypergraph to learn user preferences from high-order relations. To alleviate the bad effect of model bias on the mutual influence between director and actor, DAHCR models the director's options by sampling from a categorical distribution with Gumbel-softmax.
- We conduct extensive experiments on two benchmark datasets, and DAHCR effectively improves the performance of conversational recommendation.

2 Related Work

Different from traditional recommendation systems [Zhao *et al.*, 2022; Wang *et al.*, 2022] that predict the user's preference based on his/her historical behaviors, conversational recommendation systems (CRS) [Priyogi, 2019; Xie *et al.*, 2021; Zhou *et al.*, 2020] aim to communicate with the user and recommend items based on the attributes explicitly asked during the conversation. Various efforts have been conducted to explore the challenges in CRS which can mainly be categorized into two tasks: dialogue-biased CRS studies the dialogue understanding and generation [Chen *et al.*, 2019; Kang *et al.*, 2020; Liu *et al.*, 2020], and recommendation-biased CRS explores the strategy to consult and recommend [Christakopoulou *et al.*, 2016; Christakopoulou *et al.*, 2018; Sun and Zhang, 2018; Lei *et al.*, 2020a]. This work focuses on the multi-round recommendation-biased CRS (MCR) [Lei *et al.*, 2020a] which focuses on the setting where the MCR aims to recommend the target item to the user by iteratively asking attributes and recommending items in limited turns.

For each turn in CRS, the system naturally includes two make-decision procedures, when to recommend (*i.e.,* ask or recommend), and what to talk about (*i.e.,* which attribute/item to inquire/recommend). Early works for the MCR improve the strategies of when and what attributes to ask, while the decision of which item to recommend is made by external heuristic rules. EAR [Lei *et al.*, 2020a] utilizes latent vectors to capture the current state of MCR, and employs policy gradient to improve the strategy of deciding when to ask questions about attributes and which attribute to ask. To reduce the action space in policy learning, SCPR [Lei *et al.*, 2020b] improves the strategy to only decide whether to ask or recommend and develops external path reasoning methods to decide which attribute to ask or which item to recommend. These works, however, isolate strategies for different problems and make the policy learning of these strategies hard to converge. To solve this problem, Unicorn [Deng *et al.*, 2021] unifies the two decision procedures as the selection from the candidate action space consisting of items and attributes. Specifically, Unicorn proposes a graph-based Markov Decision Process (MDP) environment to choose actions from the candidate action space. MCMIPL [Zhang *et al.*, 2022] further considers the user's multiple interests in the unified strategy and develops a multi-interest policy learning module. Despite effectiveness, these works ignore the variant roles of different decision procedures, which may lead to sub-optimal CRS strategies.

3 The Proposed Model

We first introduce the problem definition of multi-turn conversational recommendation (MCR). Next, we introduce the framework and the model of our proposed Director-Actor Hierarchical Conversational Recommender (DAHCR).

3.1 Problem Formulation

In this section, we formulate the problem of multi-turn conversational recommendation (MCR), which aims to recommend the target item to the user by asking attributes and recommending items in the limited turns of the conversation.

Specifically, let $V = \{v_1, v_2, \cdots, v_M\}$ denotes the item set. For each item v, there exists an attribute set P_v associated with the item. At the beginning of each conversation, a user u initializes the conversation session with a target item v^* and an attribute that belongs to the target item $p_0 \in P_{v^*}$. The candidate item set V_{cand} is formed with the items associated with p_0 and the candidate attribute set P_{cand} is constructed by the attributes associated with the items in the candidate item set V_{cand}. Then at each turn t, MCR can either ask the user an attribute $p_t \in P_{cand}$ or recommend a certain number of items (*e.g.,* the top ten items) $V_t \subseteq V_{cand}$ to the user. According to the target item v^* and its associated attributes P_{v^*}, the user will choose to accept or reject the proposal of MCR. Based on the user's feedback, MCR will update the candidate attribute set P_{cand} and the candidate item set V_{cand}. The conversation will continue until the max turn T and the recommendation is successful if the target item v^* is recommended within T.

3.2 DAHCR Framework

As illustrated in Figure 1 (c), we propose the Director-Actor hierarchical conversational recommendation policy Learning, a novel framework for MCR. At each time step t, the director chooses an option $o_t \in \mathcal{O}$, and the actor chooses the primitive action $a_t \in \mathcal{A}_{t|o_t}$ accordingly. Consequently, the state is updated to s_{t+1} with the transition $T(s_{t+1}|s_t, o_t, a_t)$. The user's feedback $f_t \in \{acc, rej\}$, the extrinsic reward $r_t^a \in \mathcal{R}^a$, and intrinsic motivation $r_t^o \in \mathcal{R}^o$ are given according to s_t, o_t and a_t. Specifically, the main components of DAHCR $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, T, \mathcal{R}^o, \mathcal{R}^a \rangle$ are defined as:

State $\mathcal{S}$. The current state contains three components, including the interactive history $\mathcal{I}^t$, the related nodes $\mathcal{N}^t$, and the hypergraph $\mathcal{G}^t$ among the user and related nodes:

$$s_t = [\mathcal{I}^t, \mathcal{N}^t, \mathcal{G}^t], \tag{1}$$

where $\mathcal{I}^t = [(a_j^1, f_j)|j = 1, 2 \cdots t-1]$. The related nodes $\mathcal{N}^t = \{u\} \cup P_{acc}^t \cup P_{rej}^t \cup V_{rej}^t \cup V_{cand}^t$ contains the user u, the accepted attributes P_{acc}^t, the rejected attributes P_{rej}^t, the rejected items V_{rej}^t, and the candidate items V_{cand}^t.

Options $\mathcal{O}$. Based on the state s_t of the current turn, the director should choose an option $o_t \in \mathcal{O}$, where $\mathcal{O} = \{ask, rec\}$ denotes whether to ask or recommend.

Primitive Actions $\mathcal{A}$. Based on the director's option o_t and the state s_t, the actor selects the primitive action $a_t \in \mathcal{A}_{t|o_t}$, where $\mathcal{A}_{t|ask} = P_{cand}^t$ is the candidate attributes to ask and $\mathcal{A}_{t|rec} = V_{cand}^t$ means the candidate items to recommend.

Transitions T. Follows previous works [Lei *et al.*, 2020b; Deng *et al.*, 2021], the state s_{t+1} updates based on the user's response:

$$\begin{cases} P_{acc}^{t+1} = P_{acc}^t \cup a_t, & \text{if } o_t = ask, f_t = acc \\ P_{rej}^{t+1} = P_{rej}^t \cup a_t, & \text{if } a_t = ask, f_t = rej \\ V_{rej}^{t+1} = V_{rej}^t \cup a_t, & \text{if } a_t = rec, f_t = rej \end{cases}, \tag{2}$$

$$V_{cand}^t = V_{P_{acc}^t} \setminus V_{rej}^t,\ P_{cand}^t = P_{V_{cand}^t} \setminus (P_{acc}^t \cup P_{rej}^t), \tag{3}$$

where $V_{P_{acc}^t}$ denotes the items that satisfy all the accepted attributes and $P_{V_{cand}^t}$ denotes all the attributes that belong to the candidate items. Follows Lei *et al.* [2020b], items that have rejected attributes are not eliminated from V_{cand}^t.

Extrinsic Rewards $\mathcal{R}^a$. Extrinsic rewards are special signals passed from the environment to the agent, and guide the agent to select user-preferred actions. Five kinds of rewards are designed as previous works [Lei *et al.*, 2020b; Deng *et al.*, 2021]: (1) $r_{acc|rec}^a$, a strongly positive reward when recommend successfully; (2) $r_{rej|rec}^a$, a slightly negative reward when the recommended items are rejected; (3) $r_{acc|ask}^a$, a slightly positive reward when the asked attribute is accepted; (4) $r_{rej|ask}^a$, a slightly negative reward when the asked attribute is rejected; (5) r_{quit}^a, a strong negative reward when the maximum turn reaches.

Intrinsic Motivation $\mathcal{R}^o$. The intrinsic motivation is passed from the actor to the director to estimate the effectiveness of the director's option. Since recommending is an inefficient action when the user's preference is not certain enough (*i.e.,* the target item is outside the top ten in the actor's ranking list of items), we assign a positive reward r_+^o to the option of ask and a negative reward r_-^o to the option of rec in this situation. Inversely, when the user's preference is certain, r_+^o and r_-^o are assigned to the option of rec and ask, respectively.

3.3 DAHCR Policy Learning

State encoder. The interactive history $\mathcal{I}^t = [(o_j, f_j)|j \in \{1, 2 \cdots t-1\}]$ between DAHCR and the user contains the director's historical options o_j and the user's feedback of accepting or rejecting $f_j \in \{acc, rej\}$. With the embedding of the interactive history $\{\mathbf{X}_h^1, \mathbf{X}_h^2, \cdots, \mathbf{X}_h^{t-1}\}$, the interactive state $\mathbf{s}_h^t$ is obtained with GRU networks [Cho *et al.*, 2014]:

$$\mathbf{s}_h^t = GRU(\mathbf{s}_h^{t-1}, \mathbf{X}_h^{t-1}). \tag{4}$$

The interactive state $\mathbf{s}_h^t$ encodes the historical interaction between the agent and the user, and is excepted to guide DAHCR to learn CRS strategy to decide whether the user's preference is certain enough (*e.g.,* the recommended attribute is accepted for several turns) for recommending items.

To learn the user's preference for the specific attributes and items, we build a dynamic hypergraph $\mathcal{G}_u^{(t)} = (\mathcal{N}^{(t)}, \mathcal{H}^{(t)}, \mathbf{A}^{(t)})$, including: (1) the set of related nodes $\mathcal{N}^t = \{u\} \cup P_{acc}^t \cup P_{rej}^t \cup V_{rej}^t \cup V_{cand}^t$; (2) a hyperedge set $\mathcal{H}^{(t)}$, whose element $h \in \mathcal{H}^{(t)}$ denotes a hyperedge between the user, an attribute and items. In our case, for each attribute $p \in \mathcal{N}^{(t)}$, we define a hyperedge h_p corresponding to the attribute p; (3) a $|\mathcal{N}^{(t)}| \times |\mathcal{H}^{(t)}|$ adjacent matrix $\mathbf{A}^{(t)}$

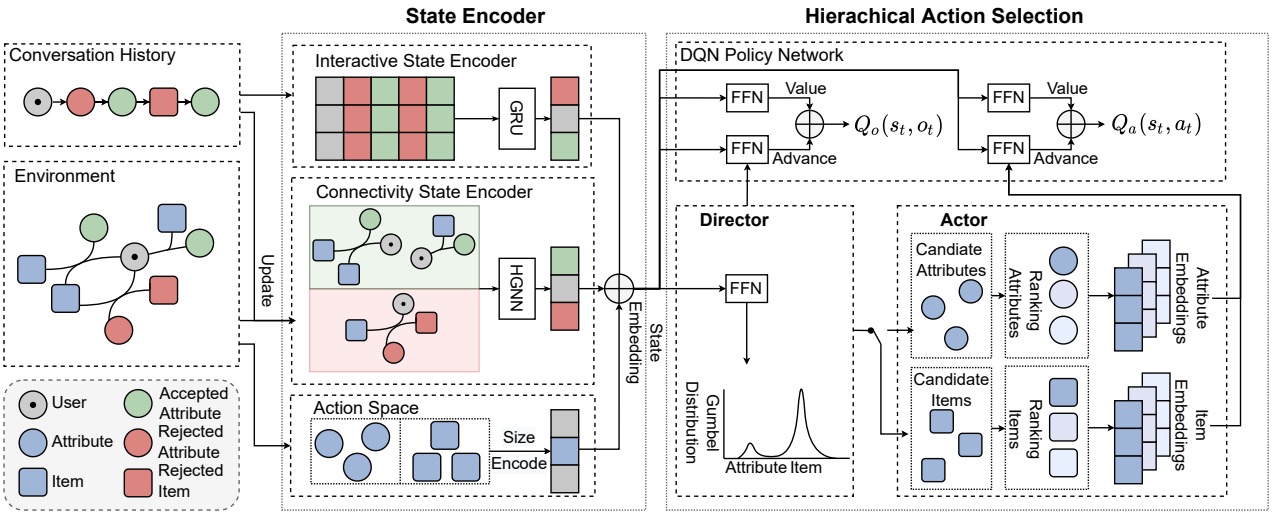

Figure 2: The overview of Director-Actor Hierarchical Conversational Recommender (Best view in color).

which denotes the weighted edge between each node and hyperedge, with entries denoted as:

$$A_{i,j}^{(t)} = \begin{cases} 1, & \text{if } n_i = u, p_{h_j} \in \mathcal{P}_{acc}^{(t)} \\ -1, & \text{if } n_i = u, p_{h_j} \in \mathcal{P}_{rej}^{(t)} \\ \frac{1}{|\mathcal{V}_{h_j}^{(t)}|}, & \text{if } n_i \in \mathcal{V}_{h_j}^{(t)} \\ 1, & \text{if } n_i = p_{h_j} \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where p_{h_j} denotes the attribute corresponding to the hyperedge h_j, and $\mathcal{V}_{h_j}^{(t)}$ indicates items that satisfy the attribute p_{h_j}.

To take advantage of the connectivity information from the dynamic hypergraph, we employ hypergraph neural networks [Xia *et al.*, 2022] to refine the node representation with structure and connectivity information. Firstly, we aggregate information propagated from nodes to related hyperedges:

$$\boldsymbol{H} = \boldsymbol{D}_h^{-1} \boldsymbol{A}^T \boldsymbol{E} \boldsymbol{W}_n, \tag{6}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{N}^{(t)}| \times d}$ denotes the initial embedding of related nodes $\mathcal{N}^{(t)}$, $\boldsymbol{W}_n \in \mathbb{R}^{d \times d}$ is the weight matrix, and $\boldsymbol{D}_h$ is the diagonal matrix denoting the degree of the hyperedges, which is defined as the number of nodes connected by hyperedges.

During the conversation, the hyperedges are successively generated when the user accepts or rejects the asked attribute. Moreover, the higher-level interactions between different hyperedges are also important in learning user preferences. To model the sequential information and hyperedge-wise feature interactions, higher-level hypergraph layers further pass messages through the interactions between hyperedges as:

$$\boldsymbol{H}_f^l = \text{MHSA}_f(\boldsymbol{H}_f^{l-1}, \boldsymbol{H}_f^{l-1}, \boldsymbol{H}_f^{l-1}), \tag{7}$$

where $f \in \{acc, rej\}$, and $\text{MHSA}(\cdot)$ indicates the multi-head self attention [Vaswani *et al.*, 2017]. Finally, we aggregate the information from the hyperedges to refine the nodes' representations $\mathbf{\Gamma}_l$ and then obtain the connectivity state s_g^t:

$$s_g^t = \sum_l \mathbf{\Gamma}_l(u), \mathbf{\Gamma}_l = \text{ReLU}(\mathbf{A} \cdot \boldsymbol{H}^l). \tag{8}$$

With the interactive state s_h^t and the connectivity state s_g^t, the final state is obtained by:

$$s^t = s_h^t \oplus s_g^t \oplus s_{len}^t, \tag{9}$$

where n_i denotes the node in the hypergraph, s_{len}^t encodes the size of the candidate item and attribute set by dividing the length $|V_{cand}|$ and $|P_{cand}|$ into ten-digit binary features [Lei *et al.*, 2020a], since it is also an important basis for deciding to ask or recommend (*e.g.,* the recommendation is easier to be successful when the candidate items' size is small).

Hierarchical Action Selection Strategy. After obtaining the state encoding the interactive history and the user's preference, we design a novel dueling Q-network to conduct policy learning under the hierarchical structure. Following the basic assumption that the delayed rewards are discounted by a factor of γ, we define the Q-value $Q_o(s_t, o_t)$ and $Q_a(s_t, a_t)$ as the expected reward for the director's option o_t and the actor's action a_t based on the state s_t:

$$Q_o(s_t, o_t) = f_{\theta_V}^o(s_t) + f_{\theta_A}^o(s_t, o_t), \tag{10}$$

$$Q_a(s_t, a_t | o_t) = P^*(s_t, o_t)(f_{\theta_V}^a(s_t) + f_{\theta_A}^a(s_t, a_t)), \tag{11}$$

where the value function $f_{\theta_V}^o(\cdot)$ $f_{\theta_V}^a(\cdot)$, and the advantage function $f_{\theta_A}^o(\cdot)$ $f_{\theta_A}^a(\cdot)$ are four separate multi-layer perceptions (MLP). $P^*(s_t, o_t)$ controls the action space of the actor's actions by masking the Q-value of actions a_t according to the director's option o_t. To realize a differentiable discrete sample of the director's option and alleviate the bad effect of model bias on the mutual influence between the director and actor, we model o_t by sampling from a categorical distribution with Gumbel-softmax as:

$$P^*(s_t, o_t) = \frac{exp((log(P(s_t, o_t)) + \epsilon)/\tau)}{\sum_{o \in \mathcal{O}} exp((log(P(s_t, o)) + \epsilon)/\tau)}, \tag{12}$$

where $\epsilon = -log(-log(x))$ and x is sampled from $Uniform(0, 1)$. The temperature parameter τ controls the bias and variance of the likelihood distribution. When τ is

larger, the likelihood is smoother with more variance and less bias. $P(s_t, o_t)$ is calculated with the softmax function as:

$$P(s_t, o_t) = \frac{exp(Q_o(s_t, o_t))}{\sum_{o \in \mathcal{O}} Q_o(s_t, o)}, \quad (13)$$

where $\mathcal{O} = \{ask, rec\}$ denotes the space for the director's option. Gumbel-Softmax is used to alleviate the bias when lacking policy learning in the mutual influence between Director and Actor: (1)Biases (e.g., bad options generated in the early learning of Director) caused by Director may filter out of efficient actions for the Actor. Hence, we consider sampling options from a Gumbel distribution, which may pave the way to explore the right action for Actor; (2)Bias (e.g., false feedback) caused by Actor may affect the convergence of Director, and thus to alleviate the issue, we consider to optimize the Gumbel distribution, rather than the deterministic function of Director's option.

The optimal Q-function with the maximum expected reward $Q_o^*(s_t, o_t)$ and $Q_a^*(s_t, a_t)$ for the director's option and the actor's primitive action are achieved by optimizing the hierarchical policy π_o and π_a, follows the Bellman function [Bellman and Kalaba, 1957] as:

$$Q_o^*(s_t, o_t) = \mathbb{E}_{s_{t+1}}[r_t^o + \gamma \max_{o_{t+1} \in \mathcal{O}} Q_o^*(s_{t+1}, o_{t+1}|o_t)], \quad (14)$$

$$Q_a^*(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r_t^a + \gamma \max_{a_{t+1} \in \mathcal{A}_{t+1|o_{t+1}}} Q_a^*(s_{t+1}, a_{t+1}|a_t)], \quad (15)$$

where $\mathcal{A}_{t+1|o_{t+1}}$ denotes the action space of the actor's primitive actions according to the director's option (*i.e.,* $\mathcal{A}_{t+1|ask}$ denotes all the candidate attributes and $\mathcal{A}_{t+1|rec}$ denotes all the candidate items).

Model training. For each turn, the agent will get the intrinsic motivation r_t^o to the director's option, the extrinsic reward r_t^a to the actor's primitive action, and the candidate actions space $\mathcal{A}_{t+1}$ is updated according to the user's feedback. We define a replay buffer $\mathcal{D}$ following Deng *et al.* [2021], which stores the experience $(s_t, o_t, a_t, r_t^o, r_t^a, s_{t+1}, \mathcal{A}_{t+1})$. For the training procedure, we sample mini-batch from the buffer and optimize the model with loss function as follows:

$$\mathcal{L}_1(\theta_Q) = \mathbb{E}_{(s_t, o_t, r_t^o, s_{t+1}) \sim \mathcal{D}}[(y_t^o - Q_o(s_t, o_t; \theta_Q))^2], \quad (16)$$

$$\mathcal{L}_2(\theta_Q) = \mathbb{E}_{(s_t, a_t, r_t^a, s_{t+1}, \mathcal{A}_{t+1}) \sim \mathcal{D}}[(y_t^a - Q_a(s_t, a_t; \theta_Q))^2]. \quad (17)$$

$\mathcal{L}_1$ and $\mathcal{L}_2$ are alternatively optimized to teach DAHCR efficiently interacts with the user and predict the user's preference, where $\theta_Q = \{\theta_V, \theta_A\}$, and y_t^o, y_t^a are target values for the director's options and the actor's actions, which are based on the optimal value function as:

$$y_t^o = r_t^o + \gamma \max_{o_{t+1} \in \mathcal{O}} Q_o^*(s_{t+1}, o_{t+1}; \theta_Q), \quad (18)$$

$$y_t^a = r_t^a + \gamma \max_{a_{t+1} \in \mathcal{A}_{t+1}} Q_a^*(s_{t+2}, a_{t+1}; \theta_Q). \quad (19)$$

To alleviate the problem of overestimation bias, we adopt the double Q-learning [Van Hasselt *et al.*, 2016] to employ target networks $Q_o^{'}$ and $Q_a^{'}$ as period copies from the online networks to train the network following previous works [Deng *et al.*, 2021; Lei *et al.*, 2020b].

Dataset	LastFM	LastFM*	Yelp	Yelp*
Users	1,801	1,801	27,675	27,675
Items	7,432	7,432	70,311	70,311
Interactions	76,693	76,693	1,368,606	1,368,606
Attributes	33	8,438	29	590
Entities	9,266	17,671	98,605	98,576
Relations	4	4	3	3
Triples	138,215	228,217	2,884,567	2,533,827

Table 1: Statistics of datasets.

4 Experiments

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

- **RQ1**: Compared with the state-of-the-art methods, does our framework achieves better performance?
- **RQ2**: What are the impacts of key components on performance?
- **RQ3**: How do hyper-parameters settings (such as the layer number of hypergraph neural networks) affect our framework?

4.1 Experiment Setting

Datasets. For better comparison, we follow previous works to conduct experiments[1] on LastFM, LastFM* for music artist recommendation and Yelp, Yelp* for the business recommendation. The statistics of datasets are illustrated in Table 1.

- **LastFM** [Lei *et al.*, 2020a]: LastFM is designed to evaluate the binary question scenario for the music artist recommendation, where the user gives preference towards an attribute using yes or no. Following Lei *et al.* [2020a], we manually merge relevant attributes into 33 coarse-grained attributes.
- **Yelp** [Lei *et al.*, 2020a]: Yelp is designed for enumerated questions for the business recommendation, where the user can select multiple attributes under one category.
- **LastFM*** and **Yelp*** [Lei *et al.*, 2020b]: Following Lei *et al.* [2020b], we construct the datasets with original attributes and pruning off the attributes with frequency < 10, named LastFM* (containing 8438 attributes) and Yelp* (containing 590 attributes) separately.

Metrics.

Following previous studies [Lei *et al.*, 2020a; Lei *et al.*, 2020b; Deng *et al.*, 2021], we adopt three widely used metrics for conversational recommendation: SR@t, AT, and hDCG. Success rate (SR@t) is adopted to measure the cumulative ratio of successful recommendations by the turn t. Average turns (AT) is used to evaluate the average number of turns for all sessions. hDCG@(T, K) is used to additionally evaluate the ranking performance of recommendations. Therefore, the higher SR@t and hDCG@(T, K) indicate better performance, while the lower AT means an overall higher efficiency.

[1] https://github.com/Snnzhao/DAHCR

Models	LastFM			LastFM*			Yelp			Yelp*		
	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG
Abs Greedy	0.222	13.48	0.073	0.635	8.66	0.267	0.264	12.57	0.145	0.189	13.43	0.089
Max Entropy	0.283	13.91	0.083	0.669	9.33	0.269	0.921	6.59	0.338	0.398	13.42	0.121
CRM	0.325	13.75	0.092	0.580	10.79	0.224	0.923	6.25	0.353	0.177	13.69	0.070
EAR	0.429	12.88	0.136	0.595	10.51	0.230	0.967	5.74	0.378	0.182	13.63	0.079
SCPR	0.465	12.86	0.139	0.709	8.43	0.317	0.973	5.67	0.382	0.489	12.62	0.159
UNICORN	0.547	11.57	0.176	0.798	7.58	0.412	0.985	5.33	0.397	0.522	11.55	0.174
MCMIPL	0.633	11.54	0.191	0.839	6.89	0.412	0.981	5.65	0.387	0.552	11.31	0.178
DAHCR	**0.712**†	**10.83**†	**0.213**†	**0.925**†	**6.31**†	**0.431**†	**0.992**	**5.16**†	**0.400**†	**0.626**†	**11.02**†	**0.192**†

Table 2: Experimental results.† represents the improvement of DAHCR over all baselines is statistically significant with p-value < 0.01. hDCG indicates hDCG@(15, 10). SR@15 and hDCG are the higher the better, while AT is the lower the better.

Implementation Details.

Following previous works [Deng *et al.*, 2021; Lei *et al.*, 2020b], we adopt TransE [Bordes *et al.*, 2013] from OpenKE [Han *et al.*, 2018] to pretrain the embedding of nodes in the constructed graph with the training set. The embedding size and the hidden size are set as 64 and 100. The temperature parameter τ is set to be 0.7 for Yelp* and 0.3 for the other three datasets. The layer number of hypergraph neural networks is selected from 1, 2, 3, and 4. We set the intrinsic motivations as: $r^o_+ = 1, r^o_- = -1$. The settings of the extrinsic rewards are the same as previous works [Lei *et al.*, 2020a; Lei *et al.*, 2020b; Deng *et al.*, 2021]: $r^a_{rec\_acc} = 1, r^a_{rec\_rej} = -0.1, r^a_{ask\_acc} = 0.01, r^a_{ask\_rej} = -0.1, r^a_{quit} = -0.3$. In the training procedure, the size of the experience replay buffer is 50,000, and the batch size is 128. The learning rate and the L_2 norm regularization are set to be 1e-4 and 1e-6, with Adam optimizer. The discount factor γ is set as 0.999. Our experiment is conducted on Nvidia RTX 3090 graphic cards equipped with an AMD r9-5900x CPU (32GB Memory).

4.2 Baselines

To demonstrate the effectiveness of the proposed DAHCR, we choose state-of-the-art methods for comparison. Specifically, we first choose two rule-based methods, three reinforcement learning-based methods that outsource part of decision procedures, and then two reinforcement learning (RL)-based method that unifies two decision procedures.

- **Max Entropy [Lei *et al.*, 2020a].** This method employs a rule-based strategy to ask and recommend. It chooses to select an attribute with maximum entropy based on the current state, or recommends the top-ranked item with certain probabilities.
- **Abs Greedy [Christakopoulou *et al.*, 2016].** This method only makes the item-recommendation actions and updates the model based on the feedback. It keeps recommending items until the successful recommendation is made or the pre-defined round is reached.
- **CRM [Sun and Zhang, 2018].** A RL-based method that records the users' preferences into a belief tracker and learns the policy deciding when and which attributes to ask based on the belief tracker.
- **EAR [Lei *et al.*, 2020a].** This method proposes a three-stage solution to enhance the interaction between the conversational component and the recommendation component.
- **SCPR [Lei *et al.*, 2020b]** This method models CRS as an interactive path reasoning problem. It prunes irrelevant candidate attributes by traversing attribute vertices on the graph based on user feedback.
- **UNICORN [Deng *et al.*, 2021]** A RL-based approach that unifies the two decision strategies. It learns graph-enhanced state representations for RL via graph neural networks.
- **MCMIPL [Zhang *et al.*, 2022]** A state-of-the-art approach to CRS that extends the unified conversational recommendation strategy with multi-interest representations of the user.

4.3 Performance Comparison (RQ1)

Overall performance.

From the overall performance of all methods reported in Table 2, we make the following observations:

- **Our proposed DAHCR achieves the best performance.** DAHCR significantly outperforms all the baselines by achieving a higher success rate and hDCG, and less average turn on four datasets. The reason for such improvement can be attributed to the following aspects: i) With the hierarchical strategy, our proposed DAHCR can reduce the action space and avoid the data bias introduced by the imbalance in the number of items and attributes. ii) The director and actor in DAHCR can well play their roles of choosing the effective option (*i.e.,* ask or recommend) and learning the user's preference over the candidate items and attributes. iii) The intrinsic motivation can well deal with the problem of weak supervision on the director's option effectiveness. iv) Modeling user preferences with dynamic hypergraph, DAHCR could specify the attributes that motivate the user to like/dislike the item with high-order relations. v) Modeling the director's option by sampling from a categorical distribution can well alleviate the bad effect of model bias on the mutual influence between the director and the actor.

- **Mutual influence of different decision procedures and the user's dynamic preference in the policy learning is important for CRS.** From Table 2, DAHCR, MCMIPL and UNICORN surpasses CRM, EAR and SCPR in terms of three metrics over four datasets. There are two reasons for this performance: (i) CRM, EAR and SCPR isolate the strategies for different decision procedures during the training process, which will make the conversational recommendation strategy hard to converge. This demonstrate the importance of the mutual influence between different decision procedures in the training process. (ii) Compared with other methods, DAHCR, MCMIPL and UNICORN use reinforcement learning to update the user's preference on the attributes and items, which leads to better performance. This proves the importance of learning the user's dynamic preference for CRS.

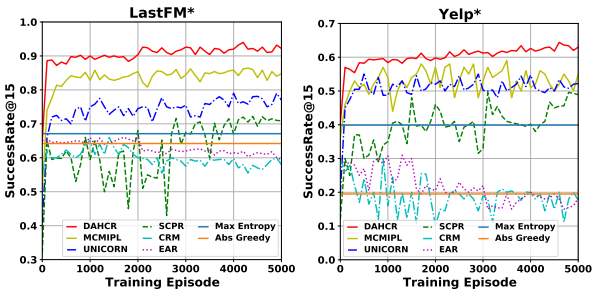

Figure 3: Test performance at different training epochs.

Training Efficiency.
Figure 3 shows the performance curves of the different methods tested on LastFM* and Yelp*, respectively. The test performance curves for the unsupervised methods Max Entropy and Abs Greedy are shown as two horizontal lines for comparison. It can be seen that DAHCR is far superior in converge speed and stability to all the baselines. The reason is that the hierarchical conversational recommendation strategy of DAHCR can realize more effective action choice and better learn the user's preference. In the Yelp*, where the action candidate space is larger, the performance of EAR and CRM does not improve much or even gets worse as the training process iterates. These results demonstrate the efficiency and effectiveness of the proposed DAHCR.

	LastFM*			Yelp*		
	SR@15	AT	hDCG	SR@15	AT	hDCG
DAHCR	**0.925**	**6.31**	**0.431**	**0.626**	**11.02**	**0.192**
(a) - w/o Hie.	0.842	6.88	0.415	0.551	11.35	0.176
(b) - w/o Hyper.	0.909	6.54	0.428	0.608	11.13	0.189
(c) - w/o Gumbel.	0.863	6.70	0.418	0.560	11.30	0.181
(d) - w/o Intrinsic.	0.887	6.63	0.424	0.570	11.31	0.183

Table 3: Results of the Ablation Study.

4.4 Study of DAHCR (RQ2&RQ3)

Next we investigate the underlining mechanism of our DAHCR with four ablated models that remove the hierarchical framework, dynamic hypergraph, Gumbel-softmax, and intrinsic motivation, respectively. From the results in Table 3, we observe that:

- The performance of DAHCR suffers a significant degradation when replacing the hierarchical framework with the unified framework. This demonstrates the importance of modeling the variant roles of different decision procedures in CRS.
- The performance of DAHCR drops when replacing the dynamic hypergraph neural networks with graph neural networks. We attribute this to the importance of high-order relations (*i.e.,* user-attribute-item) in modeling user preferences for CRS.
- The model performs worse when removing the Gumbel-softmax. This result suggests that alleviating the bad effect of model bias on the mutual influence between the director and the actor is necessary. Our method that models the director's option by sampling from the categorical distribution with Gumbel-softmax can reasonably deal with such an effect.
- The performance of the model drops when removing the intrinsic motivation, which indicates the necessity of intrinsic motivation to train DAHCR from weak supervision on the director's effectiveness.

Figure 4 shows the experimental results by varying the layer number of hypergraph neural networks. Two-layer DAHCR performs better than one-layer DAHCR since it can capture high-order collaborative information in the dynamic hypergraph. But the performance of DAHCR does not always increase when the layer number increases. We attribute this to the noise that increases along with the hop of neighbors.

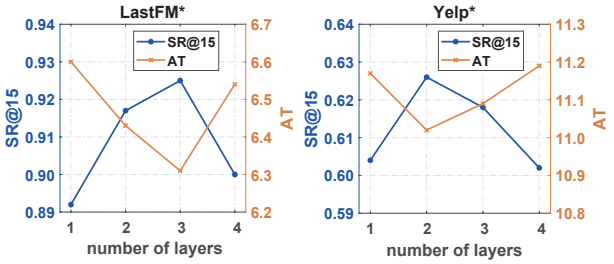

Figure 4: Impact of Layer Number(L)

5 Conclusion

In this work, we propose a Director-Actor Hierarchical Conversational Recommender (DAHCR) with the director to select the most effective option (*i.e.,* ask or recommend), followed by the actor accordingly choosing primitive actions that satisfy user preferences. The intrinsic motivation is designed for training from weak supervision on the director's effectiveness, a dynamic hypergraph is developed to learn user preferences from high-order relations, and Gumbel-softmax is employed to alleviate the bad effect of model bias on the mutual influence between director and actor. Experimental results on two real-world datasets show that the proposed DAHCR outperforms the state-of-the-art methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No.62276110, Grant No.61772076, in part by CCF-AFSG Research Fund under Grant No.RF20210005, and in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank the anonymous reviewers for their comments on improving the quality of this paper.

References

[Battaglia *et al.*, 2018] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[Bellman and Kalaba, 1957] Richard Bellman and Robert Kalaba. On the role of dynamic programming in statistical communication theory. *IRE Transactions on Information Theory*, 3(3):197–203, 1957.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[Chen *et al.*, 2019] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 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)*, pages 1803–1813, 2019.

[Chentanez *et al.*, 2004] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17, 2004.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Christakopoulou *et al.*, 2016] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems, 2016.

[Christakopoulou *et al.*, 2018] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. Q&r: A two-stage approach toward interactive recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 139–148, 2018.

[Deng *et al.*, 2021] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1431–1441, 2021.

[Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[Han *et al.*, 2018] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 139–144, 2018.

[Kang *et al.*, 2020] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y Lan Boureau, and Jason Weston. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 1951–1961. Association for Computational Linguistics, 2020.

[Lei *et al.*, 2018] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 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)*, pages 1437–1447, 2018.

[Lei *et al.*, 2020a] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 304–312, 2020.

[Lei *et al.*, 2020b] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, pages 2073–2083, 2020.

[Li *et al.*, 2018] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31, 2018.

[Liu *et al.*, 2020] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. Towards conversational recommendation over multi-type dialogs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1036–1049, 2020.

[Pei *et al.*, 2022] Jiahuan Pei, Cheng Wang, and György Szarvas. Transformer uncertainty estimation with hierarchical stochastic attention. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 2022.

[Priyogi, 2019] Bilih Priyogi. Preference elicitation strategy for conversational recommender system. In *Proceedings*

of the twelfth ACM international conference on web search and data mining, pages 824–825, 2019.

[Sun and Zhang, 2018] Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 235–244, 2018.

[Tarvainen and Valpola, 2017] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[Van Hasselt *et al.*, 2016] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2022] Ziyang Wang, Wei Wei, Xian-Ling Mao, Guibing Guo, Pan Zhou, and Sheng Jiang. User-based network embedding for opinion spammer detection. *Pattern Recognition*, 125:108512, 2022.

[Xia *et al.*, 2022] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 70–79, 2022.

[Xie *et al.*, 2021] Zhihui Xie, Tong Yu, Canzhe Zhao, and Shuai Li. 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*, pages 1400–1409, 2021.

[Zhang *et al.*, 2022] Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Bo Long, and Jian Pei. Multiple choice questions based multi-interest policy learning for conversational recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2153–2162, 2022.

[Zhao *et al.*, 2022] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. Multi-view intent disentangle graph networks for bundle recommendation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 2022.

[Zhou *et al.*, 2020] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1006–1014, 2020.