User-Centric Conversational Recommendation with Multi-Aspect User Modeling

Shuokai Li*,†
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
lishuokai18z@ict.ac.cn

Ruobing Xie†
WeChat Search Application Department, Tencent, China.
ruobingxie@tencent.com

Yongchun Zhu*
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
zhuyongchun18s@ict.ac.cn

Xiang Ao*
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
aoxiang@ict.ac.cn

Fuzhen Zhuang
Institute of Artificial Intelligence, Beihang University
SKLSDE, School of Computer Science, Beihang University
zhuangfuzhen@buaa.edu.cn

Qing He*,§
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
heqing@ict.ac.cn

Abstract

Conversational recommender systems (CRS) aim to provide high-quality recommendations in conversations. However, most conventional CRS models mainly focus on the dialogue understanding of the current session, ignoring other rich multi-aspect information of the central subjects (i.e., users) in recommendation. In this work, we highlight that the user's historical dialogue sessions and look-alike users are essential sources of user preferences besides the current dialogue session in CRS. To systematically model the multi-aspect information, we propose a User-Centric Conversational Recommendation (UCCR) model, which returns to the essence of user preference learning in CRS tasks. Specifically, we propose a historical session learner to capture users' multi-view preferences from knowledge, semantic, and consuming views as supplements to the current preference signals. A multi-view preference mapper is conducted to learn the intrinsic correlations among different views in current and historical sessions via self-supervised objectives. We also design a temporal look-alike user selector to understand users via their similar users. The learned multi-aspect multi-view user preferences are then used for the recommendation and dialogue generation. In experiments, we conduct comprehensive evaluations on both Chinese and English CRS datasets. The significant improvements over competitive models in both recommendation and dialogue generation verify the superiority of UCCR. The source code will be available on https://github.com/lisk123/UCCR.

CCS Concepts

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Natural language generation**.

Keywords

conversational recommender system; multi-aspect; user modeling

ACM Reference Format:
Shuokai Li, Ruobing Xie, Yongchun Zhu, Xiang Ao, Fuzhen Zhuang, Qing He. 2022. User-Centric Conversational Recommendation with Multi-Aspect User Modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 11-15 July 2022, Madrid, Spain* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532074

1 Introduction

In recent years, great efforts have been made to develop the conversational recommender systems (CRS) [5, 12, 22, 32, 33, 39], which aim to provide high-quality recommendations for users in conversations. Generally, CRS methods can be roughly divided into a recommender module and a dialogue module [2, 39]. The *dialogue module* converses with users through natural language. In contrast, the *recommender module* learns user preferences based on the dialogue contents, and provides appropriate recommendations for users. For the generative CRS [18, 39], the recommended items are naturally integrated into the natural language replies and given to users. Different from traditional recommender systems [3], CRS mainly captures user preferences according to the current dialogue session, and thus should handle both natural language understanding and user modeling [6]. Currently, it has also been widely used in various real-world scenarios, such as intelligent voice assistants (e.g., "Siri") and customer services on E-commerce platforms.

To model users' preferences and provide high-quality recommendations, lots of CRS models focus on better natural language

* The authors are at the Key Lab of Intelligent Information Processing of Chinese Academy of Sciences. Xiang Ao is also at Institute of Intelligent Computing Technology, Suzhou, China.
† Shuokai Li and Ruobing Xie have equal contributions.
§ Qing He is the corresponding author.

SIGIR '22, 11-15 July 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532074

understandings. Some works enhance the dialogue representation learning with more sophisticated encoders [14, 39]. Other methods also introduce useful external information such as knowledge graphs (KGs) [2, 39] and user reviews [18]. However, most of them pay too much attention to the current dialogue session and only learn the preferences reflected by the session (although we admit that it is indeed an important source of user preferences), ignoring the central subjects in CRS, i.e., **users**. In practical systems, users usually have various multi-aspect features such as user's historical dialogue sessions and user profiles besides the current session, which could help to provide more comprehensive understandings of users from different perspectives. However, there is seldom work that focuses on user-centric preference learning in CRS.

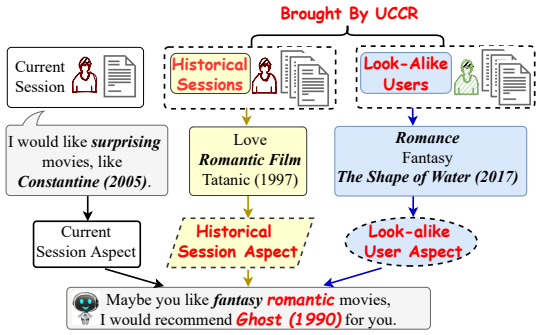


Figure 1: An example of the multi-aspect user information. UCCR introduces the historical dialogue sessions and look-alike users to CRS for user-centric preference learning.

In this work, we attempt to emphasize users and polish the model's ability on user-centric preference learning in CRS. As in Fig. 1, the user preferences in real-world CRS could be mainly extracted from three aspects: (1) the user's **current dialogue session**, which is the main information widely adopted by conventional CRS models. (2) The user's **historical dialogue sessions**, which stores user's historical preferences from multiple views. This historical information is beneficial since users tend to have similar preferences with their historical behaviors, which is inspired by the idea of item-CF [24]. (3) The user's **look-alike users**, which could be retrieved by the relevance of user profiles or user historical behaviors. It learns users' preferences via their similar users under the instruction of user-CF [36]. The newly-introduced information on historical dialogue sessions and look-alike users is beneficial especially when the current session contains little information.

However, incorporating multi-aspect user information in CRS is non-trivial, since it is challenging to decide how much we should learn from historical and look-alike features without confusing the current session modeling. Differing from users in classical recommender systems, users in CRS will actively interact with the system via natural language. Hence, their user intentions are more explicit and definite according to the current sessions, and thus the historical and look-alike features should be considered under the constraints of the current user intentions. We hope to smartly utilize multi-aspect features, successfully capturing both the basic *fantasy* intention from the current session and the hidden *romantic* preference from the historical and look-alike features in Fig. 1.

In light of the observations above, we propose a novel **User-Centric Conversational Recommendation (UCCR)** framework to jointly model user's multi-aspect information in CRS. Specifically, UCCR learns the multi-aspect user preferences mainly from three information sources, including the user's current dialogue session, historical dialogues sessions, and look-alike users. UCCR mainly consists of four parts: (1) We first design a historical session learner to capture users' diverse preferences in their historical sessions besides learning from the current session. Precisely, we extract multi-view user preferences from the dialogues, including the word-level semantic view, entity-level knowledge view, and item-level consuming view. The correlations between the current and historical information are also considered in historical preference learning. (2) We propose a multi-view preference mapper to learn the intrinsic correlations among different views in the current/historical sessions. The main idea is that two views of a user should be more relevant, since they reflect similar preferences of the same user. We design three self-supervised cross-view objectives between these views as supplements to the supervised losses, which enables a more sufficient training of user multi-view preferences. (3) For the look-alike user aspect, we refer to the similar users' preferences as a user-CF based supplement to the target user's understanding. User basic profiles and user historical behaviors, which are essential sources of personalization, could be used for the user similarity calculation. A temporal look-alike user selector is designed for more precise user generalization. (4) Finally, multi-aspect user-centric modeling is conducted to jointly encode multi-aspect multi-view user preferences into the final user representation.

Through UCCR, these multi-aspect features are properly incorporated under the guiding ideology of user-centric modeling. Compared with conventional CRS models that focus on current session understanding, our UCCR comprehensively understands users from multiple aspects (current dialogue session, historical dialogue sessions, and look-alike users) and multiple views (word, entity, and item views), which returns to the essence of user understanding in recommendations. We summarize the contributions of this work as follows:

- We emphasize the user-centric modeling in CRS, and systematically highlight and verify the significance of historical dialogue sessions and look-alike users, returning to the essence of user understanding in CRS. To the best of our knowledge, we are the first to jointly model current dialogue session, historical dialogue sessions, and look-alike users via a user-centric manner in CRS.
- We propose a set of techniques to precisely extract useful user preferences related to the current user intentions from multiple views, including a historical session learner, a multi-view preference mapper, and a temporal look-alike user selector.
- UCCR achieves the best performances against SOTA baselines on both the dialogue and recommendation parts in two real-world datasets. Extensive model analyses and ablation tests also help to better understand multi-aspects user information in CRS, shedding light on real-world applications.

2 Preliminaries

Background of CRS. Modern CRS methods [2] aim to provide high-quality items through a multi-turn dialogue with users. Thus they consist of two major components, namely recommendation

module and dialogue generation module [41]. The *dialogue generation module* aims to generate utterances and converse with users. Each dialogue session may consist of multiple turns. In each dialogue *turn*, the dialogue module could either interact with users or recommend an item. The *recommendation module* aims to provide proper items according to the information in sessions.

There are mainly three objects in CRS, namely user mentioned entities, words, and items. *User mentioned entities* $e \in \mathcal{E}$ are entities in certain KG extracted from dialogues, which contain structural knowledge. In contrast, *words* $w \in \mathcal{W}$ reflect semantic knowledge in dialogues. In conventional CRS, *items* $d \in \mathcal{I}$ are recommended mainly via user preferences learned from user mentioned entities and words in the current dialogue sessions [39]. Note that in our setting, all items (e.g., movies) are also entities in $\mathcal{E}$ [14, 41].

Notions of our UCCR. In real-world CRS, a user may have multiple dialogue sessions with the system. We organize user dialogue sessions in chronological order. For a user u from $\mathcal{U}$ having T dialogue sessions, we have the following definitions:

Definition 1: Current Dialogue Session. We regard the T-th session as the current dialogue session that we should recommend for. In the current (dialogue) session, when recommending items at a certain turn, all t user mentioned entities $C_e = \{e_1^T, ..., e_t^T\}$ of the current session before this turn are viewed as the *current entities*. For words, the definition of *current words* C_w is the same as C_e.

Definition 2: Historical Dialogue Sessions. We call all previous sessions before the current session as historical dialogue sessions. It also includes the *historical entities* $\mathcal{H}_e = \{\mathcal{H}_e^1, \cdots, \mathcal{H}_e^{T-1}\}$ and *historical words* $\mathcal{H}_w$ extracted from all $T-1$ sessions. Besides, the previous recommended items in historical sessions are viewed as *historical preferred items* $\mathcal{H}_d$. Precisely, $\mathcal{H}_e^j = \{e_1^j, ...e_{t_j}^j\}$ includes all t_j user mentioned entities of the j-th historical dialogue session, and similar as $\mathcal{H}_w$ and $\mathcal{H}_d$. We should double clarify that our proposed historical dialogue session is completely different from the dialog/conversation history used in [2, 18, 39], for their "historical" information locates in the historical sentences (turns) of the current dialogue session. To the best of our knowledge, we are the first to highlight the significance of historical dialogue sessions in generative CRS.

Definition 3 (Look-alike Users). The look-alike users refer to similar users. The user similarity can be calculated from multiple perspectives, such as user profiles and historical behaviors. In UCCR, we rely on the historical words, entities, and items for look-alike users learning. In light of user-CF, the look-alike users may have similar tastes, thus could enhance the user representations, which is especially effective when users only have sparse information learned from the current or historical dialogue sessions.

3 Method

In this section, we propose our User-Centric Conversational Recommendation (UCCR) to the CRS task. Unlike conventional CRS methods [2, 18, 39] that merely focus on the current dialogue session, our UCCR jointly models multi-aspect user features including (1) current dialogue session, (2) historical dialogue sessions, and (3) look-alike users for comprehensive user understandings.

The user-centric framework works as follows: First, in both current and historical sessions, we jointly consider word, entity, and item views to model user current and historical preferences. The historical session learner is specially designed to effectively extract useful information related to the current user intention (Sec. 3.1 and 3.2). Second, we propose the multi-view preference mapper to learn the intrinsic correlations among words, entities, and items in both current and historical sessions via multiple self-supervised objectives (Sec. 3.3). For look-alike users, as user preferences are changing dynamically, we also design a temporal look-alike user selector to find more appropriate similar users (Sec. 3.4). Finally, the overall user preferences are learned by jointly considering multi-aspect and multi-view user features (Sec. 3.5). The overview illustration of our UCCR framework is shown in Fig. 2.

3.1 Current Session Learner

We first introduce how to encode the user features from the current dialogue session (i.e., current words C_w and current entities C_e).

3.1.1 Current Entity Learner Following [2, 39], we use the widely-used knowledge graph DBpedia [11] as the entity source. It stores factual knowledge triples $< e_1, r, e_2 >$, where $e_1, e_2 \in \mathcal{E}$ are entities, and $r \in \mathcal{R}$ is the relation. Note that the user mentioned entities are pre-marked and fixed via DBpedia in our CRS datasets [14, 41]. As entity relation is important to consider, following [2, 39], we adopt the powerful R-GCN [25] to encode the structural triple information into entity representations as follows:

$$\boldsymbol{v}_e^{l+1} = \sigma\Big(\sum_{r\in\mathcal{R}}\sum_{e'\in\mathcal{N}_e^r}\frac{1}{\mathrm{Z}_{e,r}}W_r^l\boldsymbol{v}_{e'}^l + W^l\boldsymbol{v}_e^l\Big), \tag{1}$$

where $\boldsymbol{v}_e^l \in \mathbb{R}^d$ is the l-th layer's representation of entity e, and $\mathcal{N}_e^r$ is the one-hop neighbor set of e under the relation r. W_r^l and W^l are trainable weights of layer l, and $\mathrm{Z}_{e,r}$ is a normalization factor. For convenience, we use the last layer's representation $\boldsymbol{v}_e^L$ as the entity representation $\boldsymbol{v}_e$. Via Eq. 1, the current entities $C_e = \{e_1^T, \cdots, e_t^T\}$ are transformed into an entity matrix $\boldsymbol{V}_e = \{\boldsymbol{v}_{e_1^T}, \cdots, \boldsymbol{v}_{e_t^T}\}$. Next, following [2, 39], we incorporate the self-attention mechanism to aggregate the entity matrix $\boldsymbol{V}_e$ according to the importance of entities. The final current entity representation $\boldsymbol{r}_e^c$ is built as:

$$\begin{aligned}\boldsymbol{r}_e^c &= \text{R-GCN}(C_e) = \mathcal{F}(\boldsymbol{\mu}_e(\boldsymbol{V}_e)^{\mathrm{T}}),\\ \boldsymbol{\mu}_e &= \text{Softmax}(\boldsymbol{b}_e\text{Tanh}(W_e\boldsymbol{V}_e)),\end{aligned} \tag{2}$$

where $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ is a linear transformation, $\boldsymbol{b}_e \in \mathbb{R}^{1\times d}$ and $W_e \in \mathbb{R}^{d\times d}$ are trainable weights.

3.1.2 Current Semantic Learner The structural knowledge in entities reflects user preferences accurately, but lacks generalization. According to [39], the semantic information of words could effectively improve the ability of preference generalization. Following [39], we adopt an external lexical dataset ConceptNet [27] to bring in prior semantic information. The semantic similarities from the dataset are used to build edges between words. Precisely, given the words in the current dialogue session $C_w = \{w_1^T, ..., w_t^T\}$, we first leverage GCN to learn the embeddings of current words. Then the current semantic representation $\boldsymbol{r}_w^c = \text{GCN}(C_w)$ is calculated via self-attention similar as Eq. (2).

3.2 Historical Session Learner

In this section, we introduce the historical information to improve user preference learning. However, directly calculating historical

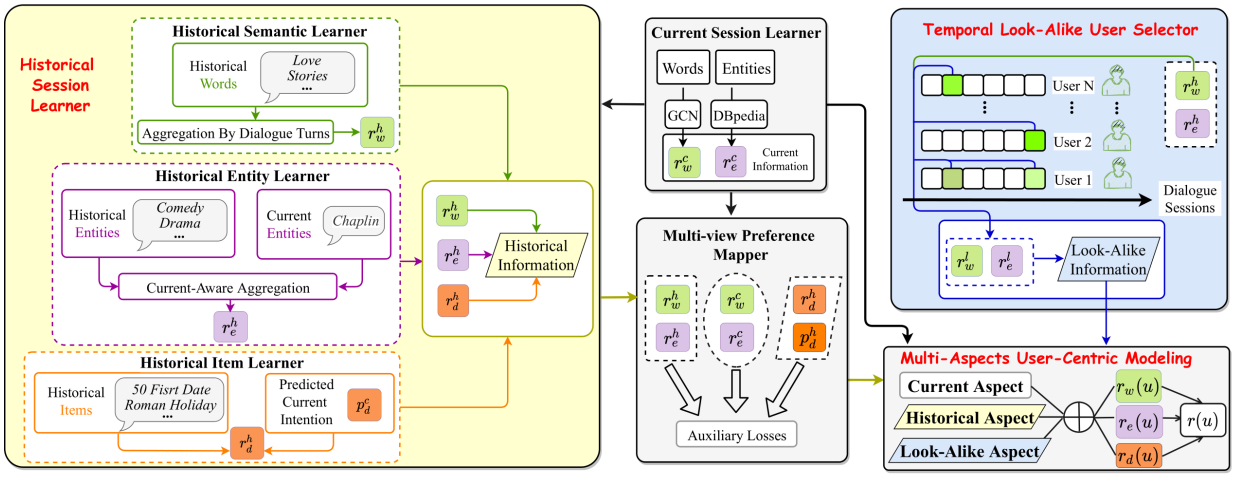


Figure 2: The overview of our model UCCR. First, the multiple views information is encoded by the historical and current session learners. Second, the multi-view preference mapper further explores the correlations between views. Next, the temporal look-alike users selector provides another aspect feature. Finally, these aspects are fused by the user-centric modeling module.

session information as the current session in Sec. 3.1 will lead to poor performances, since the gap between user's historical and current intentions may confuse the current recommendation. We believe that the historical sessions should better work as supplements under the constraints of users' current intentions, providing additional information to complement user preferences. Hence, we propose a multi-view historical session learner to capture current-related historical information from entity, semantic, and item views.

3.2.1 Historical Entity Learner For the historical entities $\mathcal{H}_e = \{\mathcal{H}_e^1, ..., \mathcal{H}_e^{T-1}\}$, we first learn an entity-level session representation for each dialogue $\mathcal{H}_e^j$, and then aggregate the $T-1$ sessions according to the similarities between historical and current sessions to learn the historical entity representation $\boldsymbol{r}_e^h$.

Different from other recommender systems, users in CRS expect high-quality recommendations mainly based on the current session. The current entities are very important to reflect users' current intentions (refer to Sec. 4.4). Hence, we attempt to extract user's entity-view preferences related to the current entities from historical entities in the historical entity learner. Specifically, given the j-th historical entity set $\mathcal{H}_e^j = \{e_1^j, ..., e_t^j\}$, we use the same R-GCN and self-attention in Eq. (1) and (2) to learn the j-th session's entity representation $\boldsymbol{h}_e^j$. Next, the similarity between these learned entity representations and the current entity $\boldsymbol{r}_e^c$ is considered to weigh $T-1$ historical sessions' entity representations. The final historical entity representations $\boldsymbol{r}_e^h$ is learned as:

$$\begin{aligned} \boldsymbol{r}_e^h &= \text{Agg}(\boldsymbol{r}_e^c, \boldsymbol{h}_e^1, ..., \boldsymbol{h}_e^{T-1}) = \sum_{j=1}^{T-1} \varphi(\boldsymbol{h}_e^j, \boldsymbol{r}_e^c)\, \boldsymbol{h}_e^j, \\ \varphi(\boldsymbol{h}_e^j, \boldsymbol{r}_e^c) &= \text{Softmax}(\boldsymbol{h}_e^j W_s \boldsymbol{r}_e^c \,/\, \lambda_e), \end{aligned} \tag{3}$$

where $\text{Agg}(\cdot)$ denotes the attention-based weighted aggregation of historical entities. Through Eq. (3), the current-related historical entity-view information is successfully extracted.

3.2.2 Historical Semantic Learner We also attempt to learn historical semantic-view information correlated to the current user intention. Differing from entities, we highlight the temporal factors for modeling historical words. The semantic information close to the current session is prone to be more important than previous sessions. Hence, considering simplicity and efficiency, using temporal factors is enough to well extract useful current-related semantic information from historical words.

Concretely, given the j-th historical word set $\mathcal{H}_w^j = \{w_1^j, ..., w_t^j\}$, we first use the same GCN to learn the word representations $\boldsymbol{V}_w^j = \{\boldsymbol{v}_{w_1^j}, ..., \boldsymbol{v}_{w_t^j}\}$ like Section 3.1.2. Then the j-th session representation is calculated as:

$$\boldsymbol{h}_w^j = \mathcal{F}(\sum_{m=1}^{t} s(w_m^j)\, \boldsymbol{v}_{w_m^j}), \tag{4}$$

where $s(w_m^j) = \text{Softmax}(1, 2, ..., t)[m]$ is the importance of w_m^j according to the dialogue turn. Through this, a larger m (i.e., a latter word w_m^j) will have a larger weight, which follows the assumption that the latter words are supposed to have more useful information. Then for each user, his/her all historical session representations are also aggregated by the temporal factors: $\boldsymbol{r}_w^h = \sum_{j=1}^{T-1} s(\boldsymbol{h}_w^j)\, \boldsymbol{h}_w^j$, where j is the index of the dialogue sessions.

3.2.3 Historical Item Learner Here we consider the historical items, which is also an important source for user modeling, as the goal of CRS is to predict the user preference on items. Considering the j-th historical items $\mathcal{H}_d^j = \{d_1^j, ...d_t^j\}$, We also use R-GCN and self-attention mechanisms in Eq. (1) and (2) to learn the j-th session's item representation $\boldsymbol{h}_d^j = \text{R-GCN}(\mathcal{H}_d^j)$. Since the current item is unknown, we use the combination of current words and entities $\boldsymbol{p}_d^c$ to represent the current user intention, which is defined as:

$$\begin{aligned} \boldsymbol{p}_d^c &= g(\boldsymbol{r}_w^c, \boldsymbol{r}_e^c) = \tau \cdot \boldsymbol{r}_w^c + (1-\tau) \cdot \boldsymbol{r}_e^c, \\ \tau &= \sigma(W_g \text{Concat}(\boldsymbol{r}_w^c, \boldsymbol{r}_e^c)), \end{aligned} \tag{5}$$

where σ is the Sigmoid activation function. Finally, the historical item representation is $\boldsymbol{r}_d^h = \text{Agg}(\boldsymbol{p}_d^c, \boldsymbol{h}_d^1, ..., \boldsymbol{h}_d^{T-1})$ like Eq. (3), which directly reflect the historical consuming preferences of users.

We have also tried LSTM or GRU for the historical entity and item learner in the sequential manner. However, the computational cost is high and there is no improvement. A possible reason is that CRS is different from traditional sequential recommendations, as the current session contains the user's main intention, and is of vital importance. Thus, for the entity and item views that contain concrete structural knowledge, considering the similarity between historical and current sessions is more reasonable than temporal factors, and outperform direct sequential models.

3.3 Multi-View Preference Mapper

Now we have the following representations: (1) current entities $\boldsymbol{r}_e^c$, (2) current words $\boldsymbol{r}_w^c$, (3) historical entities $\boldsymbol{r}_e^h$, (4) historical words $\boldsymbol{r}_w^h$ and (5) historical items $\boldsymbol{r}_i^h$. In this section, we design several self-supervised objectives to learn the intrinsic correlations between different views. Inspired by contrastive learning [19], for two view representations $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of a user, we suppose that $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ should be more similar than other users'. Given a batch of samples $\mathcal{B}$, we minimize the self-supervised loss to align the two views as follows:

$$\mathcal{L}_a(\boldsymbol{v}_1, \boldsymbol{v}_2) = \sum_{u \in \mathcal{B}} (1 - \text{sim}(\boldsymbol{v}_1^u, \boldsymbol{v}_2^u))^2 + \lambda_a \sum_{u,u' \in \mathcal{B}} (\text{sim}(\boldsymbol{v}_1^u, \boldsymbol{v}_2^{u'}))^2, \tag{6}$$

where $\text{sim}(\cdot,\cdot)$ is the cosine similarity function that measures the correlation between two views. Here, u' represents the negative users, which are all other users of batch $\mathcal{B}$ except for u.

Specifically, we have three alignment tasks: (1) $\boldsymbol{v}_1 = \boldsymbol{r}_w^c$, $\boldsymbol{v}_2 = \boldsymbol{r}_e^c$; (2) $\boldsymbol{v}_1 = \boldsymbol{r}_w^h$, $\boldsymbol{v}_2 = \boldsymbol{r}_e^h$; (3) $\boldsymbol{v}_1 = \boldsymbol{r}_d^h$, $\boldsymbol{v}_2 = \boldsymbol{p}_d^h$, $\boldsymbol{p}_d^h$ is the combination of historical words and entities (refers to Eq. (5)). Here we do not align the views between current and historical information, since the user's current intention is not always consistent with historical information in CRS. Instead, we have considered the correlations between historical and current information via current-aware aggregations and temporal factors in Eq. (3), (4) and (5).

3.4 Temporal Look-Alike User Selector

It is widely verified that users having similar historical behaviors are likely to share the same preferences [36]. This could be an effective supplement to the user modeling, especially when we can only learn very little from the historical and current sessions for cold-start [20, 42–44] users. However, the user preferences often evolve over dialogues dynamically [37]. Thus, simply learning one representation for each user is not proper. Here we not only search the look-alike users from the user set $\mathcal{U}$, but also consider the fine-grained user preference slices at every time points for all users.

Based on the observations, we design a new temporal look-alike user selection, considering user preference evolution in CRS. Concretely, we decompose users into multiple recommendation turns. Suppose that a user u' has K recommendation turns totally (if there are T historical sessions and each session has t recommendation turns, we have $K = t * T$ turns in total). In each recommendation turn, the user may have distinct preferences. Thus, for the word view, at turn k, we first calculate the corresponding historical word representation $\boldsymbol{r}_w^h(u'_k)$ and current word representation $\boldsymbol{r}_w^c(u'_k)$ for user u' at time k (noted as u'_k). Then the look-alike user's enhanced representation of user u from user u' is learned as:

$$\boldsymbol{r}_w^l(u, u') = \sum_{k=1}^{K} \delta(\text{sim}(\boldsymbol{r}_w^h(u), \boldsymbol{r}_w^h(u'_k)))\ \boldsymbol{r}_w^c(u'_k), \tag{7}$$

where $\delta(x) = \max(0, x - \delta_w)$ is a clip function to avoid too much noise, and δ_w is the threshold. $\boldsymbol{r}_w^l(u, u')$ is viewed as the contribution of u' on user u's current word modeling, and $\boldsymbol{r}_w^l(u, u')$ equals 0 when $\text{sim}(\boldsymbol{r}_w^h(u), \boldsymbol{r}_w^h(u'_k))$ is smaller than δ_w for all time points. For the entity view, the formalization of $\boldsymbol{r}_e^l(u, u')$ is the same as Eq. (7). $l_w(u, u')$ and $l_e(u, u')$ will be used as look-alike user supplements for u.

To calculate the look-alike users, for each epoch, we recompute the historical representations $\boldsymbol{r}_w^h(u)$ and $\boldsymbol{r}_e^h(u)$ for all users at all time points. During training, the similarity function in Eq. (7) is only used to distinguish look-alike users. For efficiency, we set a gradient block to $\text{sim}(\cdot,\cdot)$ to avoid too much computation.

3.5 Multi-Aspect User-Centric Modeling

In this section, we introduce the user-centric modeling from multi-aspects: current session, historical sessions, and look-alike users.

3.5.1 Multi-Aspect Entity View Modeling We first introduce the multi-aspect entity view modeling for user u to get the final entity-view representation $\boldsymbol{r}_e(u)$ via information from three aspects:

$$\boldsymbol{r}_e(u) = \boldsymbol{r}_e^c(u) + \alpha_h \boldsymbol{r}_e^h(u) + \alpha_s \sum_{u' \in \mathcal{U}} \boldsymbol{r}_e^l(u, u'), \tag{8}$$

where α_h and α_s is to regulate the amount of information learned from the historical and look-alike aspects, respectively.

For the historical aspect, the coefficient α_h aims to balance the proportion of historical information incorporated into current session. Thus it is learned by the combination of historical and current representations:

$$\alpha_h = \mathcal{G}(\text{Concat}(\boldsymbol{r}_e^c(u), \boldsymbol{r}_e^h(u))) \ / \ \tau_e, \tag{9}$$

where $\mathcal{G}$ is a feed-forward network with Sigmoid activation function, and τ_e controls the value of α_h. For the temporal look-alike users representation $\boldsymbol{r}_e^l(u, u')$, we also incorporate the coefficient α_s. According to the performance on the validation set, we empirically set α_s to 1 for simplification.

3.5.2 Multi-Aspect Word View Modeling The word view representation also consists of three aspects like entity view. Similar with Eq. 8, $\boldsymbol{r}_w(u)$ is calculated as:

$$\boldsymbol{r}_w(u) = \boldsymbol{r}_w^c(u) + \beta_h \boldsymbol{r}_w^h(u) + \beta_s \sum_{u' \in \mathcal{U}} \boldsymbol{r}_w^l(u, u'), \tag{10}$$

and the details could refer to Section 3.5.1.

3.5.3 Multi-Aspect Item View Modeling As user's current preferred items are not available, the current aspect of item view is unknown and the recommender task is to predict the current item. Thus, the look-alike aspect is also not applicable to enhance item view user modeling (not applicable for current item). Thus the consuming

view is calculated via historical item representation $\boldsymbol{r}_d^h(u)$ as:

$$\begin{aligned} \boldsymbol{r}_d(u) &= \gamma_h \boldsymbol{r}_d^h(u), \\ \gamma_h &= \delta(\text{sim}(\boldsymbol{p}_d^h,\ \boldsymbol{p}_d^c)), \end{aligned} \tag{11}$$

where γ_h also balances the proportion of historical and current information. As the current consuming item is unknown, we use the combination of mentioned words and entities instead, and γ_h is calculated by the combination of $\boldsymbol{p}_d^h$ (historical user intent) and $\boldsymbol{p}_d^c$ (current user intent). To be noticed, the defined of $\boldsymbol{p}_d^h$ and $\boldsymbol{p}_d^c$ is the same as Eq. (5), $\delta(\cdot)$ and $\text{sim}(\cdot,\cdot)$ is the same as Eq. (7).

3.5.4 Multi-Aspect Multi-View Fusion With the multi-view representations learning, the user representation is calculated as:

$$\boldsymbol{r}(u) = g(\boldsymbol{r}_w(u),\ \boldsymbol{r}_e(u)) + \boldsymbol{r}_d(u), \tag{12}$$

where $g(\cdot,\cdot)$ is the combination of words and entities like Eq. (5).

3.6 Optimization

The learned user representation $\boldsymbol{r}(u)$ is leveraged to both provide high-quality recommendations and generate utterances.

3.6.1 Recommendation Objective The probability of recommending item d_i to user u is calculated by the user representation $\boldsymbol{r}(u)$:

$$\boldsymbol{p}_{rec}(u, d_i) = \text{Softmax}(\boldsymbol{r}(\boldsymbol{u})^\top \cdot \boldsymbol{d}_i), \tag{13}$$

where $\boldsymbol{d}_i$ is the representation of item d_i. Then we adopt cross-entropy loss to train the recommendation model:

$$\mathcal{L}_{rec} = -\sum_{u\in\mathcal{U}}\sum_{i=1}^{N_u} \log \boldsymbol{p}_{rec}(u, d_i) + \lambda_{CL} \sum_{(v_1,v_2)} \mathcal{L}_a(v_1, v_2), \tag{14}$$

where $\mathcal{L}_a(v_1, v_2)$ is the multi-view preference alignment loss.

3.6.2 Dialogue Generation Objective For dialogue generation, we adopt the standard seq2seq framework [29] following [2, 39]. Concretely, Transformer [30] is used as the base model for encoder and decoder, which consists of several multi-head attention layers and fully connected feed-forward layers.

Given the input utterances, the encoder first extracts the semantic feature and the decoder outputs a representation $\boldsymbol{q}$ for token generation. To incorporate the user preferences into token generation, we use $\boldsymbol{r}(u)$ as a bias feature:

$$\boldsymbol{p}_{dial}(y_t|y_1, ..., y_{t-1}) = \text{Softmax}(W^G\boldsymbol{q} + \mathcal{M}(\boldsymbol{r}(u))[y_t], \tag{15}$$

where $\mathcal{M}$ is a linear transformation which guarantees the dimension of $\mathcal{M}(\cdot)$ equals the vocabulary size. As $\boldsymbol{r}(u)$ is injected into $\boldsymbol{p}_{dial}$, the generated utterances satisfy the user needs better. Then the dialogue module is trained with the cross-entropy loss:

$$\mathcal{L}_{dial} = -\sum_{u\in\mathcal{U}}\sum_{t=2}^{N_t} \log(\boldsymbol{p}_{dial}(y_t|y_1, ..., y_{t-1})). \tag{16}$$

3.7 Motivations and Discussions on UCCR

In this work, we systematically consider the current session, historical sessions, and look-alike users in CRS. Compared with previous CRS methods, which only use the current session features or the "historical" sentences/turns in the current session, our UCCR performs a comprehensive understanding of users in CRS. Here we give detailed discussions on all model designs.

Current Session Learner. It is an important source to learn user preferences and capture user intentions in CRS. We simply follow the previous works [2, 39] to encode the current session, which is not the focused contribution of our user-centric modeling in UCCR.

Historical Session Learner. The historical sessions are completely different from the dialogue/conversation history used in previous CRS works [2, 39, 41], as their "historical" information is actually the historical turns of the current dialogue session. The usages of historical sessions in CRS are also largely different from the historical user-item interactions [1, 31] in traditional recommendations, since the recommendation in CRS is strongly constrained by the current user intentions, while traditional recommendations are not.

Thus, the main goal of historical sessions learner is: *learning current-related and regular information from historical sessions, without impeding the current session information.* To achieve this goal, we *model the relation between historical and current session.* For the structural historical entities and items, which are concrete objects, we directly use the current representations to filter the useful information. For the historical words, which contain general semantic knowledge, we simply use the temporal factor to weigh them. Both of two designs could effectively extract current-related information from historical sessions. This historical information is beneficial especially when the current session contains little information.

Temporal Look-alike User Selector. The main goal of temporal look-alike user selector is: *leveraging the similar users' current features to enhance the user modeling.* Here the similar users are calculated by the historical features. Moreover, in CRS, user preferences often change over dialogue sessions. Thus we take the user interest evolution into consideration, selecting accurate look-alike features from each time point. When both the current and historical information is little, the look-alike feature is a useful supplement.

Multi-Aspect User-Centric Modeling. To jointly add historical and look-alike features without confusing the current preference, we also consider the balance between the current and historical user preferences via our multi-aspect user-centric modeling. Thus, all three aspects could jointly provide a comprehensive user modeling.

4 Experiment

To validate the superiority of UCCR, we conduct extensive evaluations to answer the following research questions: **RQ1**: How does our UCCR perform on the recommendation and dialogue generation tasks compared with the state-of-the-art baselines? **RQ2**: What are the benefits of UCCR in cold-start scenarios? **RQ3**: How do different components of UCCR benefit its performance, i.e., different views and different aspects? **RQ4**: How do different hyper-parameter settings impact UCCR?

4.1 Experimental Settings

4.1.1 Datasets We conduct experiments on two widely-used public datasets collected from the real-world platforms, including both Chinese (TG-ReDial [41]) and English (ReDial [14]) languages. ReDial contains 10,006 dialogues consisting of 504 users related to 51,699 movies. TG-ReDial contains 10,000 dialogues consisting of

Table 1: The recommendation results. The marker * indicates that the improvement is statistically significant compared with the best baseline (t-test with p-value < 0.05).

Dataset	TG-ReDial						ReDial					
Method	HR@10	HR@50	MRR@10	MRR@50	NDCG@10	NDCG@50	HR@10	HR@50	MRR@10	MRR@50	NDCG@10	NDCG@50
SASRec	0.0048	0.0170	0.0011	0.0016	0.0019	0.0046	0.0418	0.1598	0.0385	0.0407	0.0473	0.0712
Text CNN	0.0052	0.0188	0.0015	0.0022	0.0029	0.0058	0.0733	0.1810	0.0438	0.0482	0.0576	0.0808
Bert	0.0098	0.0356	0.0027	0.0040	0.0051	0.0101	0.1499	0.2937	0.0683	0.0761	0.0813	0.1167
ReDial	0.0102	0.0370	0.0028	0.0041	0.0053	0.0107	0.1733	0.3359	0.0779	0.0841	0.0969	0.1351
KBRD	0.0141	0.0481	0.0045	0.0063	0.0072	0.0143	0.1827	0.3688	0.0784	0.0855	0.1004	0.1428
TG-ReDial	0.0168	0.0513	0.0061	0.0080	0.0088	0.0161	0.1893	0.3801	0.0801	0.0883	0.1032	0.1477
KGSF	0.0175	0.0543	0.0073	0.0088	0.0096	0.0175	0.2006	0.4034	0.0837	0.0932	0.1110	0.1556
KECRS	0.0113	0.0394	0.0033	0.0042	0.0057	0.0111	0.1772	0.3423	0.0780	0.0851	0.0983	0.1391
RevCore	0.0191	0.0581	0.0077	0.0093	0.0105	0.0189	0.2058	0.4088	0.0850	0.0946	0.1132	0.1583
UCCR w/o En	0.0167	0.0506	0.0071	0.0085	0.0092	0.0165	0.1976	0.3885	0.0812	0.0908	0.1084	0.1502
UCCR w/o Wo	0.0207	0.0592	0.0080	0.0095	0.0114	0.0196	0.2106	0.4196	0.0865	0.0959	0.1168	0.1613
UCCR w/o It	0.0211	0.0626	0.0082	0.0098	0.0116	0.0201	0.2146	0.4193	0.0865	0.0966	0.1173	0.1619
UCCR	**0.0232***	**0.0664***	**0.0088***	**0.0107***	**0.0122***	**0.0214***	**0.2161***	**0.4258***	**0.0883***	**0.0981***	**0.1182***	**0.1642***

1,482 users related to 33,834 movies. Since we highlight the historical dialogue sessions in CRS, two datasets are split according to the **chronological order**. We randomly choose several users, using their last several dialogue sessions as our validation and test sets. The remaining sessions are the train set. Here we choose the last two sessions for TG-ReDial and the last four sessions for ReDial to guarantee that the whole train/validation/test sample ratio is about 8:1:1. In ReDial, some users have less than four dialogue sessions (i.e., they have no historical session information in the train set). These users are also used for evaluating UCCR in cold-start scenarios.

4.1.2 Baselines Following [41], we evaluate the superiority of our UCCR by considering the following nine representative baselines: (1) ***SASRec*** [7] only leverages user historical items for recommendation. (2) ***Text CNN*** [9] encodes utterances in the current session to learn user preferences by CNN-based model. (3) ***BERT*** [8] is a pre-training [34] model that encodes current utterances for recommendation. (4) ***ReDial*** [14] is a CRS method which adopts an auto-encoder framework. (5) ***KBRD*** [2] adopts the external knowledge graph DBpedia for user mentioned entities in current dialogue session to enhance the user representations. (6) ***TG-ReDial*** [41] presents the task of topic-guided conversational recommendation, which incorporates topic threads to control the dialogue state transitions. (7) ***KGSF*** [39] incorporates both the semantic and KG information for modeling user preferences. They use mutual information maximization to align representations of words and entities from current dialogue session. (8) ***KECRS*** [35] proposes bag-of-entity with a high-quality KG to better capture user preferences. (9) ***RevCore*** [18] incorporates the user reviews on movies to enhance CRS models.

Among baselines, *SASRec*, *Text CNN* and *Bert* are classical recommendation methods, and *ReDial*, *KBRD*, *TG-ReDial*, *KGSF*, *KECRS* and *RevCore* are CRS methods. All these methods only consider the current dialogue session. Besides, we do not compare [18] since it needs external user review information. For fair comparisons, we implement all the baselines and UCCR by the open-source toolkit CRSLab [38]. The hyper-parameter settings of baselines follow the default settings on CRSLab, which reaches the best performances. Note that the results of CRS methods on TG-ReDial are slightly lower than the public results[1], as we split the two datasets by **chronological** order, while previous work simply randomly split the samples, which omits the user's historical information.

4.1.3 Evaluation Metrics The recommender module and dialogue generation module are evaluated separately. For the recommender part, we want to know whether UCCR models the user preferences and provides high-quality recommendations accurately. Thus, we adopt HR@k, MRR@k and NDCG@k for evaluation ($k = 10, 50$)[2]. For the dialogue module, we consider both automatic and human evaluations. In the automatic evaluations, we adopt BLEU-2,3 [21] and perplexity (PPL) for testing the accuracy and fluency of generations, and Distinct n-gram [13, 41] ($n = 2, 3, 4$) for the diversity. In the human evaluations, three annotators are invited to score the *Fluency* and *Informativeness* of the generated responses. The range of scores is from 0 to 2, and the scores of three annotators are averaged.

4.1.4 Implementation Details The dimensions of embeddings are set to 128 and 300 for recommendation and dialogue respectively. The number of layers is set to 1 for both R-GCN and GCN considering effectiveness and efficiency. The hyper-parameters of both historical entities learner λ_e (Eq. (3)) and historical items learner λ_i are set to 0.1, and λ_a (Eq. (6)) in multi-view preference mapper equals 0.1. For the historical aspect of user-centric modeling, both τ_w (Eq. (9)) for words and τ_e for entities are set to 6. For the look-alike aspect, both δ_w (Eq. (10)) and δ_e (Eq. (8)) are set to 0.85. Finally, the weight of multi-view preferences mapper loss (in Eq. (14)) is 0.025. All of them are selected by grid search on the validation set. For training, we adopt the Adam optimizer [10] with a learning rate of 0.001, where the batch size is set as 128. The epochs of preference mapper training are 3, and we train the model 25 epochs for both recommendation and dialogue tasks. For baselines, hyper-parameter settings follow their own implementations, which reaches the best performances and guarantees fair comparisons.

[1] https://github.com/RUCAIBox/CRSLab
[2] Following https://github.com/RUCAIBox/CRSLab.

Table 2: Results on dialogue generation. Flu. and Inf. stand for Fluency and Informativeness, respectively. The marker * indicates that the improvement is statistically significant compared with the best baseline (t-test with p-value< 0.05).

Dataset	TG-ReDial								ReDial							
Method	Bleu-2	Bleu-3	Dist-2	Dist-3	Dist-4	PPL	Flu.	Inf.	Bleu-2	Bleu-3	Dist-2	Dist-3	Dist-4	PPL	Flu.	Inf.
ReDial	0.0409	0.0102	0.2672	0.5288	0.8012	55.71	0.71	0.75	0.0217	0.0078	0.0689	0.2697	0.4638	56.21	0.73	0.91
KBRD	0.0423	0.0119	0.3482	0.6911	0.9972	53.08	0.83	0.88	0.0238	0.0088	0.0712	0.2883	0.4893	54.89	0.82	1.00
KGSF	0.0461	0.0135	0.4447	1.0450	1.5792	51.27	1.01	1.09	0.0249	0.0091	0.0756	0.3024	0.5177	54.75	0.95	1.14
KECRS	0.0332	0.0078	0.1893	0.3799	0.6531	58.97	0.63	0.64	0.0133	0.0051	0.0473	0.1532	0.3188	59.35	0.59	0.71
RevCore	0.0467	0.0136	0.4513	1.0932	1.6631	51.03	1.06	1.11	0.0252	0.0098	0.0769	0.3065	0.5283	54.43	0.98	1.15
UCCR w/o En	0.0465	0.0138	0.4349	1.0289	1.5543	51.33	1.02	1.08	0.0245	0.0089	0.0729	0.3001	0.5082	54.95	0.96	1.12
UCCR w/o Wo	0.0478	0.0141	0.5093	1.2239	1.8583	50.68	1.07	1.14	0.0253	0.0097	0.0801	0.3195	0.5493	54.01	1.00	1.18
UCCR w/o It	0.0481	0.0142	0.5217	1.2589	1.9122	50.34	1.08	1.16	0.0255	0.0103	0.0815	0.3255	0.5561	53.56	1.03	1.18
UCCR	**0.0494***	**0.0145***	**0.5365***	**1.2783***	**1.9376***	**50.21***	**1.13***	**1.18***	**0.0257***	**0.0106***	**0.0818***	**0.3289***	**0.5635***	**53.24***	**1.06***	**1.22***

4.2 Overall Performance (RQ1)

4.2.1 Recommendation The recommendation module's results on two datasets are shown in Table 1. Based on the results, we can see that our UCCR significantly outperforms all the baselines by a large margin on both two datasets, which verifies that UCCR could successfully capture multi-aspect multi-view user preferences and achieve the SOTA performances under the user-centric manner. We analyze the effectiveness of our UCCR compared with different baselines as follows:

First, UCCR outperforms the six CRS methods, this shows the effectiveness of the user-centric modeling, which can understand users from multiple aspects. Although the current session is very important in CRS, the historical session and look-alike user aspects are also essential supplements to model users' diverse preferences accurately, especially when there is little information in the current session. Moreover, our multi-view preference mapper also provides additional training for extracting intrinsic correlations between different views, which helps to build better user representations.

Then, our UCCR significantly outperforms the non-CRS method SASRec. SASRec is a competitive sequential recommendation (SR) method which only uses historical items to learn user preferences, and it performs badly in CRS. The reason is that SASRec ignores the core features in CRS, i.e. current session, and only depends on historical items for user modeling. In contrast, UCCR models the correlations between historical and current sessions properly in historical session learner, which brings in additional user preferences from historical sessions related to the current user intention.

Finally, our UCCR beats the non-CRS methods Text CNN and BERT, for they directly model user preferences from the contextual utterances. We can find that useful information in natural language is sparse and hard to extract. Hence, considering multi-view information from multiple aspects in UCCR is also essential.

4.2.2 Dialogue Generation. We also evaluate UCCR on the dialogue generation task. Here we do not compare with *TG-ReDial* in dialogue evaluation, since it adopts an extra pre-train model GPT-2 for generation, which is not fair comparing with other methods.

The results are shown in Table 2, and we can see that: (1) UCCR generates more fluent, diverse, and informative utterances from both automatic and human evaluations perspectives, compared with the baseline methods. The main reason is that UCCR provides better user representations by considering the historical features and look-alike users, and they serve as vocabulary bias (in Eq. 16) to generate proper tokens. Thus, better user representations also improve the generation quality. (2) Besides, compared with ReDial, we can see that the external knowledge graph of entities and semantic similarity information also contribute to better generations.

4.3 Results on Cold-Start Scenarios (RQ2)

In this section, we further evaluate UCCR in the cold-start scenarios from two perspectives: (1) the current information is limited, and (2) the historical information is limited, which are practical in CRS.

4.3.1 Cold-Start Current Information Cold-start issues are common and critical in real-world CRS. Nearly 55% recommendations occur when the user mentioned entities of the current session are no more than 2 in ReDial. Thus we simulate this scenario by considering the recommendations with few current entities, i.e., users only mention 0,1,2,3 entities in the current session.

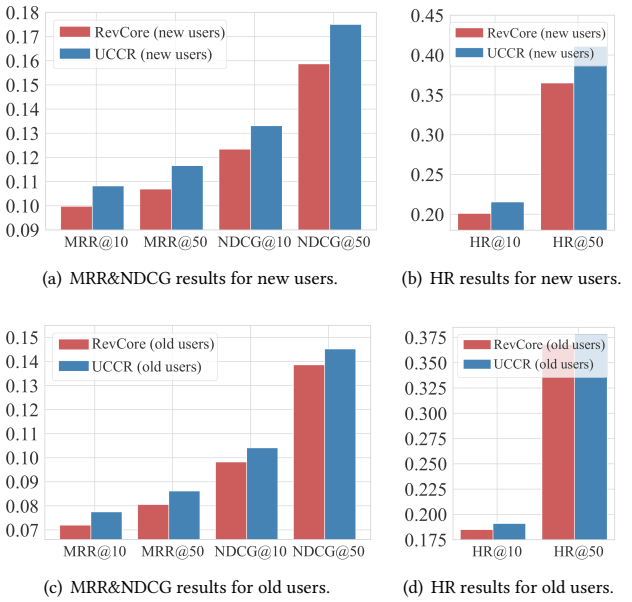
Table 3: Results of cold-start scenarios on ReDial with different number of user's current entities.

#Entity	Method	H@10	H@50	M@10	M@50	N@10	N@50
0	RevCore	10.23	26.31	0.0317	0.0409	0.0483	0.0799
	UCCR	**11.61**	**28.36**	**0.0384**	**0.0471**	**0.0574**	**0.0906**
1	RevCore	23.88	41.76	0.1094	0.1186	0.1377	0.1764
	UCCR	**24.69**	**43.93**	**0.1153**	**0.1231**	**0.1409**	**0.1830**
2	RevCore	22.65	41.92	0.0939	0.1045	0.1271	0.1693
	UCCR	**23.44**	**42.12**	**0.0996**	**0.1084**	**0.1313**	**0.1725**
3	RevCore	23.15	44.69	0.0859	0.0967	0.1202	0.1684
	UCCR	**23.41**	**44.95**	**0.0886**	**0.0987**	**0.1214**	**0.1703**
≥ 6	RevCore	18.63	40.77	0.0789	0.0898	0.1048	0.1562
	UCCR	**19.28**	**41.64**	**0.0829**	**0.0942**	**0.1116**	**0.1617**

The results are shown in Table 3. We can see that: (1) UCCR outperforms RevCore for all numbers of current entities. Especially, when there is no current entity, the performance of RevCore is poor as RevCore only leverages current features for user modeling. While

UCCR outperforms RevCore significantly as the user-centric modeling with multi-aspect information. It reconfirms that the historical features and look-alike users are powerful supplements to the user preferences modeling, and our UCCR learns multi-aspects user features appropriately. (2) As the number of current entities increases, the gap between RevCore and UCCR gradually narrowed, while UCCR consistently beats RevCore. (3) Moreover, we also show that our historical and look-alike features are not only useful in the beginning stage of the dialogue (lack of current entities), but also in the latter stage (plenty of current entities). In the last row of Table 3, we show the situation that current entities are no less than 6 (about 10% of the whole test set), where UCCR also outperforms RevCore.

4.3.2 Cold-Start Historical Information. In UCCR, we consider user historical features for user modeling, while some users have no historical dialogue sessions in practice. Hence, to evaluate UCCR with little historical information, we split users into new and old users according to whether they have historical sessions. Fig. 3 presents the results on ReDial. We can find that: UCCR performs well for both old and new users, especially for new users. The improvement ratios of new and old users are about 10.3% V.S. 4.8% on NDCG@50 and 12.6% V.S. 2.8% on HR@50. Although there are no historical sessions for new users, UCCR still learns better user preferences by multi-aspects user modeling (especially from the look-alike user aspect). Thus, our UCCR is applicable to all users.



(a) MRR&NDCG results for new users. (b) HR results for new users.

(c) MRR&NDCG results for old users. (d) HR results for old users.

Figure 3: The results for cold-start historical sessions scenario.

4.4 Ablation Study (RQ3)

To evaluate the effectiveness of each part of UCCR, we conduct the ablation study for both multiple views and aspects.

First, we show the impacts of each view in Table 1 and Table 2. Specifically, we report the results with different views turned off. Here En, Wo, and It refer to the entity, word, and item views. From the results, we can observe that all the three views contribute to the main model, as the performance decreases with any of the views removed. Another observation is that the entity view is of vital importance and the result w/o En drops largely, even worse than KGSF in Table 1. A possible reason is that entities contain structural knowledge, and are more correlated with recommended movies.

Table 4: Ablation study for three aspects.

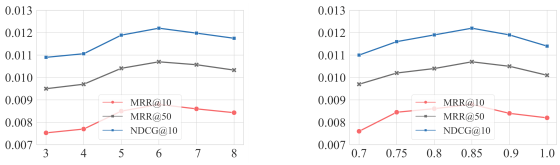
	H@10	H@50	M@10	M@50	N@10	N@50
UCCR w/o En	1.67	5.06	0.0071	0.0085	0.0092	0.0165
+ Current	1.95	6.12	0.0076	0.0093	0.0103	0.0192
+ Historical	2.14	6.33	0.0082	0.0101	0.0114	0.0204
+ Look-alike	2.32	6.64	0.0088	0.0107	0.0122	0.0214

Then, we show the effectiveness of each aspect of user modeling in Table 4. As entity view impacts the performance most, we take entity view for example, and add each aspect (i.e. current, historical, and look-alike) into UCCR w/o En. We further conduct t-test (p-value<0.05) for each aspect to confirm the statistical significance. With no doubt, when adding the current features, it improves mostly, which indicates that current features are of vital importance in CRS. When adding other two aspects, the results also rise, which shows that they are powerful supplements to user modeling in CRS. Besides, note that the results drop slightly when the historical items missing in Table 1, which further proves that current features are most important in CRS.

4.5 Parameter Sensitive Analysis (RQ4)

In this section, we investigate the sensitivity of hyper-parameters in historical aspect τ_e and look-alike aspect δ_e (in Sec. 3.5.1).

For each hyper-parameter, we search the value in empirical intervals, and six representative values are reported in Fig. 4. In general, the results first rise and then decline with the increase of the hyper-parameters values. And our UCCR performs well in a wide range of τ_e and δ_e values. In a word, both historical and look-alike aspects contribute to the recommendation performances, and we should tune them in fine-grained.



(a) τ_e for historical aspect. (b) δ_e for look-alike aspect.

Figure 4: Hyper-parameters sensitive analysis for historical entities and look-alike users on TG-Redial.

5 Related Work

Conversational Recommender System (CRS) [2, 12, 33] focus on capturing user preferences through dialogues and provide high-quality items. One category is attribute-based conversational recommendation systems [4, 5, 15, 22, 28, 32, 33, 40], which only care about providing high-quality recommendations and do not pay much attention to utterances generation. They converse with users via the pre-defined questions, which have pre-defined slots and patterns.

Recently, several works [2, 14, 17, 39, 41] focus to build end-to-end conversational recommender systems. They aim to provide high-quality recommendations and generate fluent utterances, simultaneously. [14] releases a CRS dataset coined ReDial and proposes an HRED-based [26] model. Then, the following works mainly focus on the current user features, and pay much attention to natural language understanding: incorporating external knowledge [2, 23, 39], controlling dialogue policy [17, 41], using user reviews for movies [18], designing templates for generations [16]. For external knowledge, [2] incorporates the entity knowledge graph for representation learning. Then [39] incorporates both the entity knowledge (DBpedia) and the semantic similarity of words (ConceptNet). For dialogue policy, [17] divides a conversation into several goals with goal planning, and [41] uses topics to guide dialogues. They care about generating proactive and natural human-like utterances. For movie reviews, [18] collects user reviews on movies to better model user preferences. Finally, [16] learns templates for utterances generation and does not consider the recommendation task.

These CRS methods only use the information from the current dialogue session, and omit the historical session and look-alike features. To be noticed, the dialogue/conversation history mentioned in [2, 39, 41] is not the historical sessions, as their "historical" information is actually historical turns of the current dialogue session. Moreover, historical sessions are also different from historical user-item interactions in traditional recommendations, as in CRS, the relation between historical and current sessions should be balanced.

6 Conclusion and Future Work

In this paper, we proposed a novel method UCCR for a comprehensive user modeling in CRS. Different previous methods which only focus on the current session and dialogue understanding, UCCR returns to the central subjects in CRS (i.e. users). UCCR models users by considering multi-aspect multi-view information from the current session, historical sessions, and look-alike users. The relations between different aspects are further explored to properly leverage historical and look-alike features. Extensive experiments verify the effectiveness of UCCR.

In the future, we will use more sophisticated methods to better model all aspects. We will also explore the correlations between different aspects to further improve the user understanding in CRS.

Acknowledgement

The research work is supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002104. This work is also supported by Alibaba Group through Alibaba Innovative Research Program and the National Natural Science Foundation of China under Grant (No.61976204, U1811461, U1836206). Xiang Ao is also supported by the Project of Youth Innovation Promotion Association CAS, Beijing Nova Program Z201100006820062.

References

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *SIGIR*. 378–387.

[2] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. In *EMNLP*. 1803–1813.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[4] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *SIGKDD*. 815–824.

[5] Zuohui Fu, Yikun Xian, Yaxin Zhu, Shuyuan Xu, Zelong Li, Gerard de Melo, and Yongfeng Zhang. 2021. HOOPS: Human-in-the-Loop Graph Reasoning for Conversational Recommendation. In *SIGIR*. 2415–2421.

[6] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A survey on conversational recommender systems. *arXiv preprint arXiv:2004.00646* (2020).

[7] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[8] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[9] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*. 1746–1751.

[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[11] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic web* 6, 2 (2015), 167–195.

[12] 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 *SIGKDD*. 2073–2083.

[13] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL*. 110–119.

[14] Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NeurIPS*. 9748–9758.

[15] 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. *TOIS* 39, 4 (2021), 1–29.

[16] Zujie Liang, Huang Hu, Can Xu, Jian Miao, Yingying He, Yining Chen, Xiubo Geng, Fan Liang, and Daxin Jiang. 2021. Learning Neural Templates for Recommender Dialogue System. In *EMNLP*. 7821–7833.

[17] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards Conversational Recommendation over Multi-Type Dialogs. In *ACL*. 1036–1049.

[18] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-augmented Conversational Recommendation. *arXiv preprint arXiv:2106.00957* (2021).

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[20] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *SIGIR*. 695–704.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*. 311–318.

[22] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to Ask Appropriate Questions in Conversational Recommendation. *SIGIR* (2021).

[23] Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John Philip McCrae. 2020. Suggest me a movie for tonight: Leveraging Knowledge Graphs for Conversational Recommendation. In *COLING*. 4179–4189.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[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] Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, Vol. 31.

[27] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, Vol. 31.

[28] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *SIGIR*. 235–244.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NeurIPS*. 3104–3112.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[31] Jianling Wang, Kaize Ding, and James Caverlee. 2021. Sequential Recommendation for Cold-start Users with Meta Transitional Learning. In *SIGIR*.

[32] Zhihui Xie, Tong Yu, Canzhe Zhao, and Shuai Li. 2021. Comparison-based Conversational Recommender System with Relative Bandit Feedback. In *SIGIR*. 1400–1409.

[33] 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 *WSDM*. 364–372.
[34] Zheni Zeng, Chaojun Xiao, Yuan Yao, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2021. Knowledge transfer via pre-training for recommendation: A review and prospect. *Frontiers in big Data* (2021), 4.
[35] Tong Zhang, Yong Liu, Peixiang Zhong, Chen Zhang, Hao Wang, and Chunyan Miao. 2021. KECRS: Towards Knowledge-Enriched Conversational Recommendation System. *arXiv preprint arXiv:2105.08261* (2021).
[36] Zhi-Dan Zhao and Ming-Sheng Shang. 2010. User-based collaborative-filtering recommendation algorithms on hadoop. In *2010 third international conference on knowledge discovery and data mining*. IEEE, 478–481.
[37] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*, Vol. 33. 5941–5948.
[38] Kun Zhou, Xiaolei Wang, Yuanhang Zhou, Chenzhan Shang, Yuan Cheng, Wayne Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2021. CRSLab: An Open-Source Toolkit for Building Conversational Recommender System. In *ACL*. 185–193.
[39] 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 *SIGKDD*. 1006–1014.
[40] Kun Zhou, Wayne Xin Zhao, Hui Wang, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. Leveraging historical interaction data for improving conversational recommender system. In *CIKM*. 2349–2352.
[41] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *COLING*. 4128–4139.
[42] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-Meta Framework for Cross-domain Recommendation to Cold-Start Users. In *SIGIR*. 1813–1817.
[43] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for cross-domain recommendation. In *WSDM*. 1507–1515.
[44] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *SIGIR*. 1167–1176.