

Beyond Texts: Incorporating Co-occurrences into the Review-based Conversation Recommendation Systems

HAOYAO ZHANG, ZHIDA QIN, XUFENG LIANG, JING GUO, SHUANG LI, and TIANYU HUANG, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

JOHN C. S. LUI, Computer Science & Engineering Department, The Chinese University of Hong Kong, Hong Kong

Conversational Recommender Systems (CRSs) interact with users through natural language to provide recommendations and generate responses. Due to limited information in conversation, existing works utilize KGs or reviews to improve CRS. Despite achievements, they overlook co-occurrence relations which have shown effectiveness in collaborative filtering systems. In this work, we first propose a novel framework named **CoCRS**, aiming to incorporate **Co-occurrences into the Review-based Conversation Recommendation Systems**. In CoCRS, we mine co-occurrences from two aspects: (1) *item and entity*, (2) *user and item*. For the first one, we extract entities from redundant review texts by KG and construct a relation-aware item-entity heterogeneous graph. In the second aspect, we analyze review sentiments and construct a sentiment-aware user-item bipartite graph. We encode two graphs to obtain user and entity embeddings. Since users in CRS are anonymous, we generate a virtual similar user representation to match reviews with users. Besides, we capture time-aware preference representation from two-time dimensions. Finally, we generate word-level user representation with word-oriented KG and model user preference by integrating the above representations. Extensive experiments demonstrate that CoCRS outperforms baselines and the cold-start experiment highlights its robustness. The **Large Language Model (LLM)** experiment illustrates the significant role of co-occurrence relationships in LLM-based CRS. Our code are available at <https://github.com/Qin-lab-code/CoCRS>.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Recommender systems, Conversational recommendation

This work is supported by the National Nature Science Foundation of China under Grant Number U2441242.

Authors' Contact Information: Haoyao Zhang, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: zhanghaoyao@bit.edu.cn; Zhida Qin (corresponding author), School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: zanderqin@bit.edu.cn; Xufeng Liang, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: xfliang@bit.edu.cn; Jing Guo, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: guojing@bit.edu.cn; Shuang Li, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: shuangli@bit.edu.cn; Tianyu Huang, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; e-mail: huangtianyu@bit.edu.cn; John C. S. Lui, Computer Science & Engineering Department, The Chinese University of Hong Kong, Hong Kong; e-mail: cslui@cse.cuhk.edu.hk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org/permissions).

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1558-2868/2025/12-ART26

<https://doi.org/10.1145/3771276>

ACM Reference format:

Haoyao Zhang, Zhida Qin, Xufeng Liang, Jing Guo, Shuang Li, Tianyu Huang, and John C. S. Lui. 2025. Beyond Texts: Incorporating Co-occurrences into the Review-based Conversation Recommendation Systems. *ACM Trans. Inf. Syst.* 44, 1, Article 26 (December 2025), 32 pages.
<https://doi.org/10.1145/3771276>

1 Introduction

In recent years, **Conversational Recommender Systems (CRS)** have emerged as a popular research topic in the domain of computer science. The goal of CRS is to engage in natural language conversations with users and provide high-quality recommendations. Compared to traditional recommendation systems, CRS dynamically interacts with users, obtains real-time feedback, and models user preferences based on this feedback. The system additively detects changes in user interests, thereby improving the performance of recommendations. Generally, CRS can be divided into two main modules: the recommendation module and the dialogue module. The recommendation module models user preferences based on the dialogue content and aims to provide accurate recommendations. The generation module focuses on generating fluent natural language to interact with users.

Many works have been devoted to the CRS field and have yielded notable achievements. Initial works [36, 63] utilize HRED [55] or LSTM [59] to understand the conversation and complete the recommendation task. Since a sole conversation often contains a limited number of items, and it is difficult to obtain comprehensive item representations by relying only on content, some works proposed to incorporate external information to effectively model items and generate informative representations. A series of works focuses on the KG. Chen et al. [8] and Zhou et al. [93] delve deeper into the connections among items and entities by incorporating KGs, thereby enriching the representation of each item. Moreover, Ma et al. [48] employ the methods of random walks on KG to reveal the complex causal relationships between items and entities in the conversation. To extract the structured information within them, Li et al. [38] apply tree-structured reasoning to the conversation to achieve more precise representations of the items and entities. Note that, these approaches only consider conversation content, which contains rather limited information. To address such need, Lu et al. [47] introduce texts in reviews and supplements to conversation to alleviate the problem of insufficient information. However, they directly segment the review text and simply select a fixed number of words to concatenate behind the conversation. This approach overlooks the deeper structural and semantic information within reviews, specifically the co-occurrence relationships between entities (e.g., which entities are frequently mentioned together) and the sentiment information between the reviewer and the item (e.g., the reviewer's positive attitude towards one item but negative attitude towards another). The neglect of such rich information prevents their model from achieving a comprehensive understanding of user opinions, which ultimately limits the potential for further performance improvements.

Despite their achievement, they mainly focus on descriptions of entities or items, overlooking the interactional information between users and items. However, solely relying on this information is not sufficient to accurately capture the user's preference representation. Within the realm of traditional **Collaborative Filtering (CF)** challenges, user interaction information (co-occurrence relations) has wielded substantial influence and achieved notable success. Through the exploration of co-occurrence relations, CF systems are able to uncover similarities between users and items, thereby enabling a more precise capture of user preferences. Nevertheless, in CRS, owing to the restricted interaction information within the singular conversations, depending on the user's

conversation to establish co-occurrence relationships becomes quite challenging. Xie et al. [74] utilized co-occurrence information and integrated **Retrieval-Augmented Generation (RAG)** to construct a neighborhood-based CRS to enhance the capabilities of CRS. Inspired by the work of Lu et al. [47], we find that reviews contain abundant and useful co-occurrence relations. Therefore, we propose to mine the co-occurring relationships in reviews to enhance the performance of CRS. While Xie et al. [74] also considered co-occurrence relationships in their CRS, our approach differs from theirs in that we not only incorporate co-occurrence relations from raw conversations but also explicitly mine such relationships from user reviews.

Although co-occurrence relationships already play an important role in CF, mining co-occurring relationships from reviews and introducing them into CRS has two challenges: (1) *Co-occurrence relationship mining from redundant reviews*. Traditional CF systems datasets explicitly contain a significant number of co-occurrence relationships. Reviews primarily contain explicit textual information, inevitably including abundant redundant information (e.g., definite articles and prepositions). Lu et al. [47] simply move the text from reviews to the conversation. This approach can only utilize a little descriptive information from reviews and fails to mine co-occurrence information beneficial for recommendations. Moreover, because of the abundant redundant information in reviews, simply moving words in the reviews to the end of the conversation inevitably introduces some noise. Consequently, effectively denoising and mining co-occurrence relationships from reviews presents a significant challenge. (2) *Accurately user preference modeling*. Firstly, unlike traditional recommendation systems, users in CRS are often anonymous and do not include information such as reviews. When co-occurrence relationships of users are extracted from external reviews, precise matching with users in CRS becomes infeasible. Additionally, conversations usually involve recommendations for multiple items. Each completion of an item recommendation can be named an episode. The most accurate user preferences are usually manifested in the most recent episodes. Previous approaches indiscriminately input the entire conversation into the CRS and model user preferences using all conversation information. For some long conversations, these methods are easily misdirected by past preferences (earlier episodes) and neglect recent episode impact, resulting in a bias in the captured user preferences. Thus, it is challenging to accurately model user preferences.

For the first challenge, we propose to mine the co-occurrences from two aspects. First, the co-occurrence relationships between items and entities are often obscured within amounts of redundant text. To extract this information, we utilize a KG to extract entities and items mentioned so to filter redundant information in the reviews. To obtain more informative and deeper co-occurrence relationships, we further construct a relation-aware item-entity heterogeneous graph, encompassing co-occurrence between items and entities in reviews, namely describing relationships, as well as co-occurrence relationships between items by finding items that multiple identical users review. Besides, reviews not only contain explicit interaction information but also encompass users' subjective sentiments towards items, namely sentiment polarity. To extract this information, we design a sentiment discriminator to analyze the sentiment polarity of each review. Subsequently, we utilize this sentiment polarity to construct a sentiment-aware user-item bipartite graph, comprising two types of edges: like or dislike. We employ **Relational Graph Convolutional Neural Network (R-GCN)** to encode the two graphs and obtain embeddings of users and items. This method uncovers co-occurrence relationships between items and entities in reviews, alleviating the performance issues of KG in specialized domains. The embeddings of users and items serve as a foundation for generating user preference representation.

For the second challenge, we propose to create a virtual similar user to match the co-occurrence information in the review to the users. Specifically, we search for reviews with similar sentiment polarity based on items mentioned in the conversation. Then, we consider the users who generated

these reviews as similar users. Based on the user embeddings learned from the sentiment-aware user-item bipartite graph mentioned above, we fuse the embeddings of similar users through a self-attention mechanism to obtain a more informative virtual similar user representation. Through the above approach, we address the mismatch between users and review in CRS and aggregate co-occurrence relationships from multiple users, serving as a foundation for generating richer user representations. Furthermore, to capture the most authentic user preferences, we use the recent episode from the entire conversation. Hence, we use self-attention to aggregate the entities' representations in the recent episode and the entire conversation to capture the user preferences from two dimensions. Then, we employ a gating mechanism to fuse them to gain a time-aware user preference representation. This method enables the system to perceive variations in user preferences across different temporal dimensions and capture the most genuine user preferences. Finally, we generate user preference representations from the textual dimension and integrate these three representations to obtain the most accurate user preference representation.

Overall, we propose a novel framework named **CoCRS**, aiming to incorporate **Co-occurrences into the Review-based Conversation Recommendation Systems**. This is the first work which uses mining and incorporates co-occurrences in reviews into the CRS. Specifically, we construct the sentiment-aware user-item bipartite graph and the relation-aware item-entity heterogeneous graph to mine the co-occurrence information from reviews. Then, we create the virtual similar user and generate the time-aware user representation to accurately capture the user preference. For response generation, to generate more suitable responses for the CRS, we introduce a pre-trained language model to assist in the generation of dialog responses. To validate the effectiveness of our approach, we conduct extensive experiments on two different-size CRS datasets. Experimental results show that our CoCRS outperforms competitive baselines in both the recommendation and conversation tasks. Besides, we conduct two experiments: directly applying **Large Language Models (LLMs)** to CRS and incorporating co-occurrence relationships into the LLM. The experimental results demonstrate a significant enhancement in the effectiveness of the LLM with the incorporation of co-occurrence relationships. Our main contributions are as follows:

- We introduce CoCRS, a novel review-based method for conversation recommendation systems. To the best of our knowledge, this is the first work which proposes to mine co-occurrence information from reviews and incorporate them into the CRS to enhance the performance of the system.
- We propose two specific methods for mining co-occurrence relationships from redundant reviews: relation-aware item-entity heterogeneous graph construction and sentiment-aware user-item bipartite graph construction. They respectively extract entities from review to obtain co-occurrence between item and entity, and utilize sentiment to augment co-occurrence between user and item.
- We propose an effective approach to construct a virtual similar user based on sentimental information mined from conversation context and reviews, which we use to retrieve valuable co-occurrence relationships. In addition, we propose a time-aware method to utilize the recent episode and the entire conversation, which demonstrates effectiveness for modeling user preference.
- Extensive experiments on two CRS datasets demonstrate CoCRS's superiority over existing methods. Additionally, the cold-start experiment highlights the strong robustness of our framework. The experiment with LLM exhibits incorporating co-occurrence relationships is equally effective in LLM-based CRS.

2 Related Work

2.1 Conversational Recommendation Systems

With the rapid advancement of intelligent agents on e-commerce platforms, CRS has become an emerging research topic, garnering significant attention [12, 36]. CRS mainly explores user preferences through conversational interaction and provides high-quality recommendations. Prior work can be broadly divided into two categories [94].

The first category is *attribute-based CRS* [12, 35, 63, 86], which usually interacts with users through some predefined interaction operations, such as asking for attribute information about items or recommending items and responding to users with predefined language templates. In general, the system first asks questions, and the user provides simple feedback. Subsequently, the system reduces the recommendation item space based on this feedback. Through multiple rounds of iteration, the system eventually determines the recommended items. Early work [12] introduces a preference elicitation strategy and leverages latent structures in recommendation spaces via a probabilistic latent factor model to identify which questions to ask a new user. Zhang et al. [87] developed a unified conversational search and recommendation framework, which for the first time in CRS incorporated both review data and item descriptions to enrich the system's knowledge and improve the accuracy of search and recommendation. To ascertain the timing of recommendations (deciding whether to inquire about attributes or recommend items), Lei et al. [35] determine user preference for both items and item attributes through three stages: estimation, action, and reflection. Kim et al. [27] employ a high-level policy to determine the timing of recommendations and utilize a low-level policy to decide action type through a hierarchical policy learning approach. To provide high-quality recommendations quickly, Lin et al. [39] feature a greedy **Normalized Discounted Cumulative Gain (NDCG)** attribute selector for prioritizing attributes that best represent user preferences and a user representation refiner to fuse collected preferences effectively. In recent years, with the rapid development of deep learning, Chu et al. [13] proposes a meta-exploration policy for rapid user preference identification, customizing the item recommendation module to enhance recommendation quality based on conversation states, and utilizing a transformer-based state encoder to model intricate feedback relations. Zhao et al. [88] utilize a hierarchical hypergraph neural network to capture user preferences from three views: like, dislike, and social, and employ a cross-view **Contrastive Learning (CL)** module to preserve the intrinsic features and correlations of user preferences. Ni et al. [50] leverage deep transformer-based multi-channel meta-reinforcement learning to facilitate connections between different rounds of conversations in determining whether to recommend items or pose questions. Besides, Zhang et al. [82] evaluate state-of-the-art multi-turn CRSs by examining the effectiveness of designed components in different scenarios and their transferability to other models and provide key insights and guidelines for the future. These works mainly focus on completing the recommendation task in as few rounds as possible and ignore the features of natural language. Users only communicate through pre-defined interactions, which may hurt user experiences.

The other category is *topic-based CRS*, which generally interacts with users through natural language, aiming to understand the user preferences so provide accurate recommended items, and generate utterances as responses. Due to limited information in the conversation, early works begin incorporating external knowledge to enrich the conversation context. Chen et al. [8] close the gap between recommender systems and dialogue systems by integrating them in an end-to-end framework and leveraging knowledge graphs to enhance user representations. Zhou et al. [93] use entity-oriented knowledge graph and word-oriented knowledge graph at the entity granularity and word granularity respectively to represent user preferences more comprehensively, and use MIM to narrow the gap between the two knowledge graphs. Yang et al. [79] introduce descriptions of items to

further assist user representation. Subsequently, to fully mine and utilize the information of external knowledge and conversation, Zhou et al. [97] use CL methods to bridge the semantic gap between external knowledge bases. Li et al. [37] and Lin et al. [41] further enhance the user representation by mining the relationships among users, dialogues, and entities in the dialogue database. Li et al. [38] mine structural information in user dialogue entities to learn clearer information about user preferences. Zhang et al. [84] incorporate the dialogue with CRSs and perform dynamic knowledge reasoning with the dialogue context. Shang et al. [57] uses historical dialogue data with hypergraph structure to enrich the contexts. Besides, Deng et al. [16] proposed a unified model, based on PLM, that integrates various tasks, including goal planning, topic prediction, item recommendation, and response generation, into a single sequence-to-sequence paradigm. With the rapid development of LLM, Yang et al. [80] propose a novel evaluation metric, namely Behavior Alignment, to measure the consistency of recommendation strategies between LLM-based CRS and human recommenders. Xie et al. [74] utilized co-occurrence information and integrated RAG to construct a neighborhood-based CRS to enhance the capabilities of CRS. Similarly, Zhu et al. [98] also employed RAG and CF to construct a collaborative retrieval mechanism for enhancing LLMs. Different from their work, our approach incorporates co-occurrence relationships derived not only from conversations but also from reviews. Several studies incorporate review information to enhance the performance of CRSs. Lu et al. [47] employed words from reviews to supplement the user representation. Zhang et al. [83] constructed an empathetic CRS and leveraged a review dataset to enhance its training. Zheng et al. [91] modeled user preferences from multiple perspectives, including items, entities, words, reviews, and knowledge. Zou et al. [99] utilized a Mixture-of-Experts model with a ChairBot coordinator to integrate specialized experts handling conversation history, knowledge graphs, and reviews for recommendation tasks. In contrast to their work, our method eliminates redundant information from reviews and extracts valuable co-occurrence relationships to enhance CRS for recommendation. Recently, some works propose new datasets that better reflect real-world scenarios. Liu et al. [45] present the User Needs-centric E-commerce conversational recommendation dataset, which comprises fine-grained annotated pre-sales dialogues, user behaviors, and product knowledge tuples. Guo et al. [21] create the Explainable Recommendation Dialogues dataset with 756 dialogues and over 2,000 rewritten explanations. Qu et al. [51] first explore **Voice-based CRSs (VCRs)** and create two VCRs benchmark datasets in the e-commerce and movie domains. To depict the effectiveness of CRS, Siro et al. [61] uncover the factor influencing user satisfaction in CRS, the system's ability to make relevant recommendations, based on six aspects: relevance, interestingness, understanding, task-completion, interest-arousal, and efficiency. In this article, we focus on topic-based CRS.

2.2 CF

As one of the most commonly used strategies in RS, CF concentrates on previous user behavior and preferences. Using explicit feedback like star rating and implicit information mined from users' interactions with items like reviews and browsing history, CF systems can produce recommendations automatically. Typically, CF techniques can be summarized as two categories [34]. (1) Memory-based algorithms, also known as the neighborhood approaches, try to estimate similarity between users or items, and thus establish recommendations based on users with comparable preferences or items similar to what the target user likes. (2) Model-based algorithms, also known as latent factor models, integrate CF with machine learning methods, exploiting a model to detect latent patterns in the dataset and parameterizing entities as embeddings. Despite different approaches to implementation, the core idea of mining the co-occurrence relationship between entities is consistent. **Matrix Factorization (MF)** is key method, evolving from SVD [3] to advanced methods like PMF, SVD++, and FM for sparse data [31–33, 49, 53]. In recent years, with the rapid

development of deep learning, models become more and more expressive. Wide and Deep [9], DeepFM [20], NCF [19], and LRML [64] are representative frameworks combining the advantages of Deep Neural Networks with CF-related techniques. Then, Cheng et al. [10] design a general feature-level attention neural network and apply it to CF models. Inspired by widely used **Graph Convolutional Neural Network (GCN)** [30] and GraphSage [22], which propose methods to aggregate information from neighbor nodes, and an augmentation mechanism presented by Xu et al. [77], NGCF [68] utilizes GCN in CF. LightGCN [24] points out feature transformation and nonlinear activation play little role in NGCN and thus simplify the framework. In response to the challenge of data sparsity in GNN-based recommendation systems, Liu et al. [44] integrate entities from the multimodal data into the user-item interaction graph and incorporate them into the CF model. Xu et al. [78] propose to study the vulnerability of Graph Learning-based CF and introduce a defense mechanism to enhance its robustness. Recently FourierKAN-GCF [76] reconsiders the effect of feature transformation eliminated by LightGCN [24] and proposes a novel model. Another line of work focuses on the application of CL. Liu et al. [46], SGL [71] and selfCF [95] integrate CL into CF via graph perturbation and dropout method, while HCCF [73] exploits hypergraph technique to capture global relations and NCL [42] sufficiently considers information from neighborhood. Later on, lightGCL [6] utilizes SVD for view embeddings augmentation. Simultaneously, since most CF methods use ID embeddings, Zhang et al.'s work [85] tries to leverage modality information, redefining co-occurrence relationships. In consideration of fairness issues in CF, Shao et al. [58] introduce the Rubin–Neyman framework to estimate the average causal effects of sensitive attributes and incorporate them as a regularization term to ensure fairness. Furthermore, Zhang et al. [81] quantifies and reduces interaction-wise popularity shortcuts to mitigate popularity bias in CF models. Since reviews can offer valuable insights into user preferences and mitigate data sparsity issues, researchers have investigated the integration of reviews with CF. The DeepCoNN model [90] directly utilizes review-based features for user and item representation. In contrast, CARL [72] and NARRE [7] refine embedding optimization in the interaction view and review view collaboratively. The RGCL model [60] introduces a novel technique that combines ID representation with review features and employs CL to enhance performance. Recently, MAGCL [67] proposes a multi-aspect CL framework to leverage semantic relationships within reviews, while Zhou et al. [96] introduce an explainable mechanism designed to extract valuable information from reviews more effectively. Inspired by the ideas mentioned above, we adequately consider the application of co-occurrence information and propose a practical method.

3 Method

In this section, we will elaborate on the proposed CoCRS, which is shown in Figure 1. First, in order to mine the co-occurrence relationships from reviews, we conduct a sentiment-aware user-item bipartite graph and a relation-aware item-entity heterogeneous graph. Furthermore, we encode the two graphs to generate embeddings of entities and users in reviews, which serves as a foundation for incorporating co-occurrences into CRS. Besides, we encode the DBpedia which includes descriptive information, and use its output to augment the entity embedding by an add operation to generate entity representations. After that, we distinguish similar users with the same sentiment polarity and generate the virtual similar user representation to match relevant information in reviews and dialogue. Later, to better capture the most authentic user preferences, we extract the entities from the conversation and the review with the same sentiment polarity, and respectively aggregate their representations in the recent episodes and the entire conversation based on the self-attention technique to get time-aware user representation. Moreover, we encode the ConceptNet to obtain word embeddings and aggregate the embeddings of conversation words to generate word-level

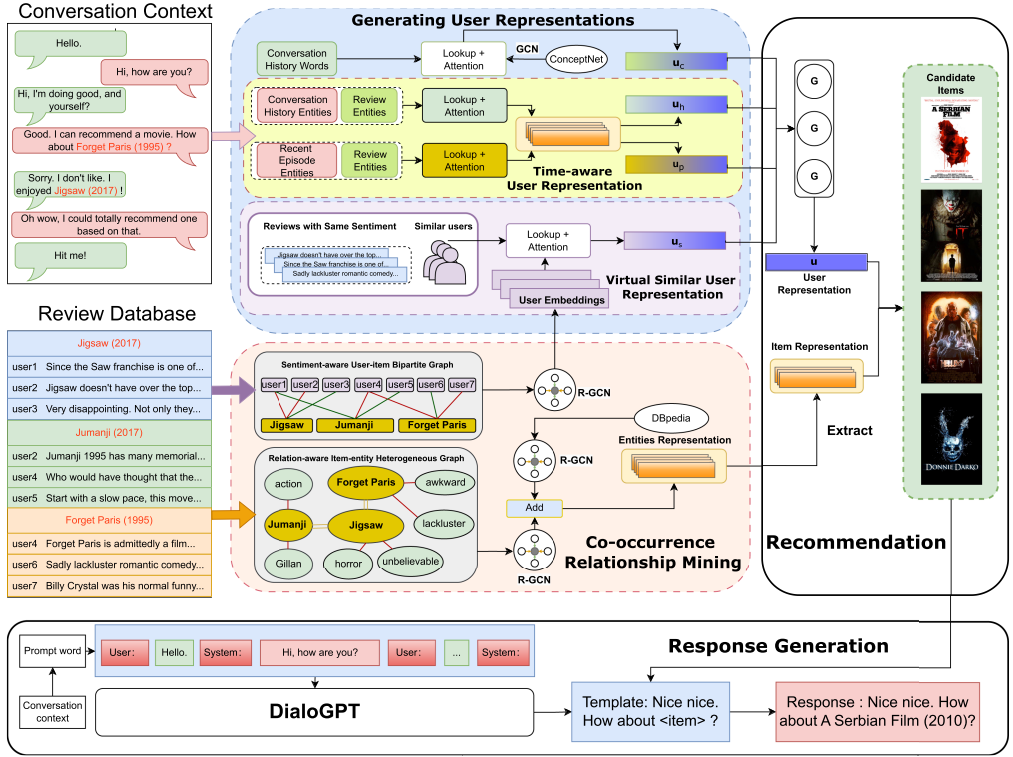


Fig. 1. The overview of our proposed CoCRS. The red line and the green line denote like and dislike in the sentiment-aware user-item bipartite graph. The single line and the double line denote co-occurrence and descriptive in the relation-aware item-entity heterogeneous graph.

user representation. Finally, we utilize the gating mechanism to fuse the above user representations and complete the recommendation task.

For the response generation, we utilize DialogGPT as a base model to complete response generation. Then, we use the prompt word and fine-tune DialogGPT to generate the response that adapts the CRS. Besides, in order to prove that the co-occurrence relationship is still beneficial in LLM, we discuss the limitations of LLM-based CRS and design an experiment to incorporate co-occurrence relationships into LLM.

3.1 External KG Encoding

The KG enables the system to discover items not mentioned in the dialogue by utilizing the internal connections within the KG, thereby facilitating the provision of diversified recommendations. Following previous works [38, 41, 48, 93], we utilize an item-oriented KG (DBpedia [2]) for entity linking, and then use an R-GCN [54] to encode the relational semantics contained in the KG to obtain the entity representations as follows:

$$\mathbf{n}_e^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}_{KG}} \sum_{e' \in \mathcal{E}_e^r} \frac{1}{Z_{e,r}} \mathbf{W}_{KG,r}^{(l)} \mathbf{n}_{e'}^{(l)} + \mathbf{W}_{KG}^{(l)} \mathbf{n}_e^{(l)} \right), \quad (1)$$

where $\mathbf{W}_{KG,r}^{(l)}$, $\mathbf{W}_{KG}^{(l)}$ are learnable weights of layer l for relational semantics, $Z_{e,r}$ is a normalization factor, $\mathcal{E}_e^r$ is a set of neighbor nodes for e under the relation r and $\mathcal{R}_{KG}$ is a relation set. After

aggregation by Equation (1), we get representations of the entity which include relational semantic information in DBpedia, denoted as $E_{KG} = \{\mathbf{n}_{e,1}, \dots, \mathbf{n}_{e,m}\}$, where m denotes the number of entities.

The word in conversations can express specific user preferences. By aggregating word representation in conversations, specific preferences can be recognized. Therefore, word information can offer more personalized recommendations. In order to obtain the semantic information of word tokens, we construct a word-oriented KG via ConceptNet [62] and adopt a GCN [30] to encode:

$$\mathbf{V}_c^{(l+1)} = \text{ReLU} \left(\mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{V}_c^{(l)} \mathbf{W}_c^{(l+1)} \right), \quad (2)$$

where $\mathbf{V}_c^{(l+1)} \in \mathbb{R}^{n_w \times d_w}$ are the embeddings of word nodes, n_w and d_w respectively express the number of words and embedding dimension.

3.2 Co-occurrence Relationship Mining

Incorporating co-occurrence relationships from reviews into CRS to enhance the performance of the system involves a critical step: effectively extracting co-occurrence relationships from reviews. Typically, reviews contain substantial redundant text, with the elements that truly contribute to recommendations often being the users, items, entities, and their co-occurrence relationships. Therefore, we develop a method that explores co-occurrence relationships in reviews from two levels. By constructing sentiment-aware user-item bipartite graph and relation-aware item-entity heterogeneous graph, we aim to mine co-occurrence relationships from reviews at the user and item levels.

3.2.1 Extracting Entities in the Review. In order to filter out redundant information from reviews, we extract entities mentioned in the reviews. Specifically, we pre-process all reviews using DBpedia Spotlight [15] to perform entity extraction. DBpedia Spotlight is a tool for automatically annotating mentions of DBpedia resources in text, providing a solution for extracting entities through DBpedia. Given a review $r = \{w_i\}_{i=1}^m$, we can extract k entities $\mathcal{E}_k^r = \{e_1, e_2, \dots, e_k\}$ via DBpedia, where w represents a word and e denotes an entity.

3.2.2 Sentiment-aware User-item Bipartite Graph Construction. To thoroughly explore the co-occurrence relationships between users and items in comments, the utilization of emotional information to construct a sentiment-aware user-item bipartite graph is contemplated. This approach enables the incorporation of emotional information while exploring the co-occurrence interactions between users and items in comments, thereby mining a fine-grained co-occurrence relationship between users and items. First, we build a sentiment polarity judgment module, and the sentiment polarity of each review can be predicted by the transformer-based sentiment discriminator:

$$\text{score} = \text{Sentiment}(r), r \in \mathcal{R}_{rev}, \quad (3)$$

where $\mathcal{R}_{rev}$ represents all the reviews, $\text{Sentiment}(r)$ represents the sentiment discriminator, processing a review r as input, and score represents the sentiment polarity of the review, which is the user's emotional information about the movie. The closer a score is to 1, the more the user likes the movie; the closer a score is to 0, the more the user dislikes the movie. From this, we can construct the sentiment-aware user-item bipartite graph $\mathcal{G}_S$, given by:

$$\mathcal{G}_S = \langle u, r, i \mid i \in \mathcal{I}, u \in \mathcal{U}, r \in \mathcal{R}_{\mathcal{G}_S} \rangle, \quad (4)$$

where $\mathcal{I}$ represents the set of items, $\mathcal{U}$ represents the set of users in reviews, $\mathcal{R}_{\mathcal{G}_S} = \{\text{"positive"}, \text{"negative"}\}$.

The sentiment and interaction between the reviewer and the item directly reflect the preference of the reviewer and the co-occurrence relationship. To better capture this preference information and

the co-occurrence relationship, we use R-GCN to encode the sentiment-aware user-item bipartite graph $\mathcal{G}_S$ like Formula (1). After aggregation, we obtain the user embeddings in the review including the sentiment information and the co-occurrence relationship, denoted by embedding matrices $U_{\mathcal{R}_S} \in \mathbb{R}^{n \times d}$, where n denotes the number of reviewers and d denotes the embedding dimension.

3.2.3 Relation-aware Item-entity Heterogeneous Graph Construction. From the entire dataset of reviews, there is aggregated co-occurrence information between items reviewed by multiple users. On a sole review, it often includes relevant items and entities, showcasing personalized co-occurrence information from the users. To mine co-occurrence relationships at the item level, we construct the relation-aware item-entity heterogeneous graph. First, we utilize the extracted entities $\mathcal{E}_k^r$ from reviews in Section 3.2.1 to serve as the information that users use to describe the item. Consequently, we establish an edge labeled “descriptive” between the item and the entities found in the reviews. In addition, when multiple users write reviews for two items, there is a co-occurrence relationship between the two items. Therefore, we establish an edge labeled “co-occurrence” between these two items, given by:

$$\mathcal{G}_{\mathcal{R}} = \langle e, r, e \rangle e \in \mathcal{E}, r \in \mathcal{R}_{\mathcal{G}_{\mathcal{R}}}, \quad (5)$$

where $\mathcal{R}_{\mathcal{G}_{\mathcal{R}}} = \{\text{“co-occurrence”}, \text{“descriptive”}\}$, $\mathcal{E}$ denotes entities. Similarly, we use another R-GCN to encode the relation-aware item-entity heterogeneous graph $\mathcal{G}_{\mathcal{R}}$ like Equation (1). Thus, we get the entity representations including descriptive information and item representations containing co-occurrence relationships, denoted by embedding matrices $E_{\mathcal{G}_{\mathcal{R}}} \in \mathbb{R}^{m \times d}$, where m denotes the number of entities and items and d denotes the embedding dimension.

The descriptive information contained in the review is personalized and inaccurate. Structured knowledge graphs are constructed manually and contain accurate descriptive information. In order to generate entity and item representations that contain co-occurrence and descriptive information, we add the entity representations $E_{\mathcal{K}_G}$ obtained from the KG and the entity representations $E_{\mathcal{G}_{\mathcal{R}}}$ of the relation-aware item-entity heterogeneous graph to obtain the fused entity and item representations:

$$E = E_{\mathcal{K}_G} + E_{\mathcal{G}_{\mathcal{R}}}. \quad (6)$$

3.3 Generating User Representations

Although the aforementioned steps enable a thorough exploration of co-occurrence relationships in reviews, they are not utilized in recommendation tasks. When integrating co-occurrence relationships into CRS and generating accurate user preferences, two key issues arise: (1) how to precisely match review users with dialogues, and (2) how to capture precise user preferences from the entire dialogue. To address these issues, in this section, we construct a virtual similar user to match co-occurrence relationships in reviews and dialogues. Subsequently, we design a time-aware mechanism to comprehensively capture user preferences from two distinct temporal dimensions.

3.3.1 Virtual Similar User Representation. We construct virtual similar users for each of the conversations to incorporate co-occurrences from reviews. First, we use Algorithm 1 to find similar users from reviews. For each item mentioned in the conversation, the interlocutor expresses an emotional polarity, positive or negative. Therefore, we find users in the reviews who have the same sentiment polarity for the item and add them to the set $\mathcal{U}_S$. The process is shown in the Algorithm 1. When find all the users with the same sentiment polarity, we get the embeddings U_S of similar users from $U_{\mathcal{R}_S}$ through a lookup operation. Then, we obtain the virtual similar user representation u_s by aggregating U_S using self-attention:

$$u_s = \text{Att}(U_S) = U_S \cdot \alpha_s, \alpha_s = \text{Softmax}(b_s^T \cdot \tanh(W_s \cdot U_S)), \quad (7)$$

Algorithm 1: Finding Similar Users

Input: Items Sequence $\mathcal{I}_S$ in conversation history; user set $\mathcal{U}$ in reviews ;

Output: similar user set $\mathcal{U}_S$;

```

1:  $\mathcal{U}_S \leftarrow \{\}$ 
2: for  $i$  in  $\mathcal{I}_S$  do
3:    $s_i \leftarrow \text{Sentiment}(i)$ 
4:   //The sentiment of the current user towards item  $i$ .
5:    $\mathcal{U}_c \leftarrow \{\}$ 
6:   for  $u$  in  $\mathcal{U}$  do
7:     if  $\text{IsSame}(s_i, \text{Sentiment}(u, i))^1$  then
8:       //The reviewer and the current user have the same sentiment for item  $i$ .
9:        $\mathcal{U}_c \leftarrow \{u\}$ 
10:    end if
11:  end for
12:   $\mathcal{U}_S \leftarrow \mathcal{U}_S \cup \mathcal{U}_c$ 
13: end for
14: return  $\mathcal{U}_S$ 

```

where $\mathbf{W}_s$, $\mathbf{b}_s$ denote learnable parameters.

3.3.2 Time-aware User Representation. To accurately model users, we capture user preferences from two different time scales: conversation history and recent conversations. Conversation history includes dialog interactions from the beginning to the present. Recent conversations include interactions from the last time the system made a recommendation to the present, which contains the discussion of the recommendation episodes. Due to the limited number of entities present in the dialogue, it is challenging to model the conversation across two different time scales. Therefore, we supplement the conversation with entities and items mentioned in reviews with similar sentiment polarity. Then, we obtain the entity sequences in the conversation history and recent conversation. Through a lookup operation, we get the entity embeddings representations $\mathbf{E}_h$ in the conversation history and the entity embeddings representations $\mathbf{E}_p$ in recent conversation.

Given the representations $\mathbf{E}_h$ and $\mathbf{E}_p$, we respectively apply self-attention operation to capture user preferences from two different time scales, which is given by:

$$\alpha_h = \text{Softmax}(\mathbf{b}_h^T \cdot \tanh(\mathbf{W}_h \cdot \mathbf{E}_h)), \mathbf{u}_h = \mathbf{E}_h \cdot \alpha_h, \quad (8)$$

$$\alpha_p = \text{Softmax}(\mathbf{b}_p^T \cdot \tanh(\mathbf{W}_p \cdot \mathbf{E}_p)), \mathbf{u}_p = \mathbf{E}_p \cdot \alpha_p, \quad (9)$$

where $\mathbf{W}_p$ and $\mathbf{W}_h$ are parameter matrix, $\mathbf{b}_p$ and $\mathbf{b}_h$ are learnable vector. Then, we use a gating mechanism [59] to fuse the user representations $\mathbf{u}_t$ from two different time scales:

$$\gamma = \sigma(\mathbf{W}_f[\mathbf{u}_h; \mathbf{u}_p]), \mathbf{u}_t = \gamma \cdot \mathbf{u}_h + (1 - \gamma) \mathbf{u}_p, \quad (10)$$

where $\mathbf{u}_t$ is the fused user representation, $\mathbf{W}_f$ denotes learnable parameters, $[\cdot]$ represents the concatenation operation.

In CRS, user preferences can vary across different times. Therefore, inspired by the work [38], in order to distinguish items between the conversation history and the recent conversation, we use

¹The function $\text{isSame}(\cdot)$ checks whether the user and a reviewer share the same polarity for a given item, returning True if matched and False otherwise.

isolation loss [38] to separate the item representations, which is calculated as follows:

$$\mathcal{L}_{iso} = \sum_{i \neq j} \frac{\mathbf{e}_i \mathbf{e}_j}{|\mathbf{e}_i| \cdot |\mathbf{e}_j|}, \quad (11)$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ respectively denote the item representations in the conversation history and the item representations in the recent conversation.

3.3.3 User Representation Fusion. According to Equation (2), we obtain the matrix $\mathbf{C}_s$ consisting of the embeddings of all words in the dialogue. Then, we use the self-attention mechanism to aggregate the representations of words in the dialogue to obtain the word-level representation $\mathbf{u}_c$ of the user:

$$\beta = \text{Softmax}(\mathbf{b}_c^T \cdot \tanh(\mathbf{W}_c \cdot \mathbf{C}_s)), \mathbf{u}_c = \beta \cdot \mathbf{C}_s, \quad (12)$$

where $\mathbf{W}_c$, $\mathbf{b}_c$ denote learnable parameters. In the preceding sections, we describe how to obtain the different user representations, including the word-level user representation $\mathbf{u}_c$, the virtual similar user embedding $\mathbf{u}_s$, and the user representation $\mathbf{u}_t$ fused at two different time scales. These three representations complement each other, enabling CoCRS to perform diversified preference modeling and effectively incorporate co-occurrences into review-based CRS. We merge these three representations using two gating mechanisms and employ an optimal fusion sequence:

$$\mathbf{u} = \text{gate}(\mathbf{u}_g, \mathbf{u}_c), \mathbf{u}_g = \text{gate}(\mathbf{u}_t, \mathbf{u}_s), \quad (13)$$

where gate represents a gating operation similar to Equation (10). The determination of the fusion sequence and related experiments are presented and analyzed in Section 4.9.

Since entity embeddings come from entity-oriented networks and word embeddings come from a word-oriented network, there is a semantic gap between them. Inspired by previous works [38, 93], we use alignment loss to close the gap, which is calculated as follows:

$$\mathcal{L}_{ali} = \mu_c \frac{\mathbf{u}_g \mathbf{u}_c}{|\mathbf{u}_g| \cdot |\mathbf{u}_c|} + (1 - \mu_c) \sum_{i \neq j} \frac{\mathbf{u}_{g,i} \mathbf{u}_{c,j}}{|\mathbf{u}_{g,i}| \cdot |\mathbf{u}_{c,j}|}, \quad (14)$$

where μ_c is a hyperparameter, i and j denote different users.

3.4 Recommendation and Optimization

Given the user representation $\mathbf{u}$ and the representation $\mathbf{e}_i$ of the recommended item i , we can calculate the probability of the recommended item i as follows:

$$P_{rec}(i) = \text{Softmax}(\mathbf{u}^T \cdot \mathbf{e}_i). \quad (15)$$

In order to expedite the process of reducing the semantic gap between word and entity and emphasizing the representation of items in recent conversations, we pre-train the recommendation module using alignment loss and isolation loss. The joint loss is expressed as follows:

$$\mathcal{L}_{pre} = \lambda \mathcal{L}_{ali} + \mu \mathcal{L}_{iso}, \quad (16)$$

where $\lambda \in [0, 1]$ and $\mu \in [0, 1]$ are hyperparameters.

Then, we train the recommendation module with cross-entropy loss, which is calculated as follows:

$$\mathcal{L}_{rec} = - \sum_u \sum_{i=1}^{N_r} \log P_{rec}(i), \quad (17)$$

where N_r is the number of recommended items.

3.5 Response Generation

In this section, we will give an introduction to the response generation model. We adopt the PLM DialoGPT [87] to build our response generation module. DialoGPT has been pre-trained on the dialogue dataset and is adept at generation in dialogue. However, since the large number of dialogues containing items in CRS, the direct use of DialoGPT cannot accommodate the task of response generation. To address this, we add a special token [ITEM] into the vocabulary to replace the items within the response, enabling DialoGPT to better adapt to the generation of responses in CRS. Once the response generation is complete, the [item] token is replaced with the items recommended by the recommendation module to complete the response generation task. To make the model more aware of its role as a recommender, we design the prompt words guidance model. Specifically, we add special tokens [User] and [System] to the front of the sentences for the roles of user and system. Then, we concatenate all sentences into a long text $[t_U, x_1, \dots, x_m, t_S, x_{m+1}, \dots, x_{N_g}]$, where dialogue history is $D = [x_1, \dots, x_m]$, generated response is $R = [x_{m+1}, \dots, x_{N_g}]$, and t_U, t_S are special tokens [User] and [System]. The probability of the generated response is calculated as follows:

$$P_{gen}(R | D) = \prod_{n=m+1}^{N_g} P_{gen}(x_n | x_1, \dots, x_{n-1}). \quad (18)$$

The loss function is set to be:

$$\mathcal{L}_{gen} = -\frac{1}{N_g} \sum_{n=1}^{N_g} \log P_{gen}(x_n | x_{<n}), \quad (19)$$

where N_g is the length of the generated response.

3.6 LLM with Co-occurrence

Recently, LLMs have exhibited unparalleled capabilities in reasoning and generation [1, 4, 5, 89]. Consequently, LLM-based CRS [18, 70] has attracted significant attention. BiasCRS [69] leverages predefined cue words with LLM to generate new dialogues, thereby enriching the dataset. RARec [14] employs a prompt-driven approach to capture user intentions and enhances CRS performance by retrieving reviews based on these intentions.

However, these methods fundamentally rely solely on the powerful text comprehension and generation capabilities of LLM. Since LLM has not been trained to learn co-occurrence information beneficial for recommendations, it is difficult to accurately capture user preferences. Consequently, their works still fail to capture user preferences relevant to recommendation tasks without co-occurrence information. Therefore, we attempt to integrate co-occurrence information into LLM-enhanced CRS. Considering the experimental equipment, we chose T5 and BERT as the LLM for our experiments.

Consider the dialogue history $D = \{x_1, \dots, x_n\}$. We adapt both BERT and T5 for the recommendation task, with BERT serving as the primary example (the implementation for T5 follows the same process). Specifically, we first encode the dialogue history D using BERT:

$$\mathbf{u}_{Bert} = \text{Pool}(\text{Bert}([x_1, \dots, x_n])). \quad (20)$$

Subsequently, based on Equations (7) and (13), we derive the aggregated representation of the virtual similar users u_s and the time-aware user representation u_t . To enrich BERT's encoding with co-occurrence information, we fuse u_s, u_t and encode representation with BERT $\mathbf{u}_{Bert}$ as follows:

$$\mathbf{u}_f = \text{gate}(\mathbf{u}_{Bert}, \text{gate}(\mathbf{u}_t, \mathbf{u}_s)). \quad (21)$$

Finally, we compute the recommendation probability for item i using the fused user representation $\mathbf{u}_f$:

$$P_{rec}(i) = \text{Softmax}(\mathbf{u}_f \cdot \mathbf{e}_i), \quad (22)$$

where $\mathbf{e}_i$ is derived from Equation (6), which incorporates co-occurrence information.

In contrast, the baseline approach using only BERT for recommendation computes the probability as follows:

$$\mathbf{y}_p = \mathbf{u}_{Bert} \cdot \mathbf{W}_{item}^T, \quad (23)$$

where $\mathbf{W}_{item}$ is a learnable parameter matrix, with each row representing a learnable item embedding. Each element i of the output vector $\mathbf{y}_p$ corresponds to the predicted score for item i . The implementation for T5 follows an analogous procedure.

During training, we optimize LLM and other modules by cross-entropy loss. For comprehensive details regarding the experimental outcomes, please refer to Section 4.10.

4 Experiment and Analysis

In this section, we evaluate our CoCRS. First, we introduce our experimental setup, including datasets, baselines, performance metrics, and implementation details. Then, we show and analyze the experimental results. Through extensive experiments, we aim to answer the following research questions:

- RQ1*: How does CoCRS perform on the recommendation and response generation tasks compared with baselines?
- RQ2*: What are the impacts of different components of CoCRS?
- RQ3*: What are the benefits of CoCRS in cold-start scenarios?
- RQ4*: How do different hyperparameter settings impact CoCRS performance?
- RQ5*: How do different fusion sequences of the three user representations affect the performance of CoCRS?
- RQ6*: How do co-occurrences affect the performance of LLM-CRS?
- RQ7*: How does CoCRS work when interacting with users?

4.1 Dataset

We verify our CoCRS on three widely used baseline datasets on CRS: *REDIAL* [36], *INSPIRED* [23], and *Reddit-Moive* [25]. The REDIAL dataset is a CRS English dataset on movie recommendations, built by the general package workers on Amazon Mechanical Turk, with 10,006 conversations, and 182,150 utterances. Similar to the REDIAL dataset, the INSPIRED dataset is a CRS English dataset on movie recommendations, but on a smaller scale, with 1,001 conversations, and 35,811 utterances. Unlike the first two crowdsourced datasets, the Reddit-Movie dataset is derived from movie-related posts on the Reddit platform, representing the largest conversational movie recommendation dataset to date. This subset comprises 171,773 dialogues with 180,562 utterances, yielding an average dialogue length of 2.1023 utterances per conversation. Detailed statistical information is presented in the Table 1.

4.2 Baselines

Respectively, we assess the performance of our CoCRS on both the recommendation and the generation task, and compare the results against relevant baseline methods from prior baselines.

Table 1. Statistics of the Datasets

Dataset	Redial	Inspired	Reddit-Movie
#Conversations	10,006	1,001	171,773
#Utterances	182,150	35,811	361,124
#Movies	6,281	1,472	26,172
#Avg. utterances	18.2040	35.7752	2.1023

(1) Baselines for Recommendation:

- *SASRec* [26], a prevalent baseline, integrates self-attention mechanism into sequential recommender systems, enabling model to leverage historical information and adaptively fit the density of datasets.
- *TextCNN* [28] utilizes CNN for semantic analysis, which can also be used to extract user preferences by encoding dialogue.
- *BERT* [17] is a widely used PLM for processing natural language in various fields. Due to the powerful ability to infer, is applied to establish recommendations based on context.
- *Baize* [75] employs a novel technique called Self-Distill with Feedback and is fine-tuned based on LLaMA-13B.
- *Vicuna* [11] is trained by fine-tuning LLaMA-13B on user-shared conversations collected from ShareGPT.
- *DeepSeek-7B-Chat* [43] is an advanced 7B-parameter language model trained on an extensive dataset comprising 2 trillion tokens of English and Chinese text.

(2) Baselines for Generation:

- *GPT-2* [52] is an auto-regressive model based on the Transformer structure. GPT-2 breaks the potential of thinking that model can only perform specific tasks without supervised learning and becomes one of the most commonly used baselines.
- *DialoGPT* [87] is also an auto-regressive model pre-trained on dialogue corpus. Based on GPT-2 architecture, DialoGPT uses fewer parameters but is capable of generating more near-human responses.

(3) Baselines for CRS:

- *REDIAL* [36] is a model proposed in the REDIAL dataset. It adopts an auto-encoder for recommendation and HRED [56] for response generation.
- *KBRD* [8] uses Transformer [65] to build response generation modules and firstly uses external KGs to enhance entity semantics referred to in conversation history to enhance user representation.
- *KGSF* [93] uses two different levels of external KGs to enhance the performance, and uses the MIM [66] for the semantic space of its two KGs.
- *RevCore* [47] proposes a framework for review enhancement to improve recommendation and response generation.
- *UCCR* [37] uses multiple aspects of information such as similar users, current conversations, and historical conversations to comprehensively model users.
- *TREA* [38] considers causal relationships of entities and proposes a tree-structure reasoning schema to model user representations.

4.3 Performance Metrics

Based on existing research, we evaluate the performance of the CRS model in both recommendation and response generation. Regarding the *recommendation task*, our objective is to determine whether the model effectively captures user preferences and provides high-quality recommendations. To

assess this, we employ the following evaluation metrics: $Recall@k$ ($k = 10, 50$), $Mean Reciprocal Rank@k$ ($MRR@k$, $k = 10, 50$), and $NDCG@k$ ($k = 10, 50$). These metrics evaluate the extent to which the top k items recommended by the model align with the actual items provided in real recommendations.

For the *response generation task*, we employ both automatic and human evaluation. In the case of automatic evaluation, we utilize the following metrics: $BLEU-2$, $BLEU-3$, and $Distinct-n$ ($n = 2, 3, 4$). $BLEU-2$ and $BLEU-3$ assess the degree of word overlap between the generated conversations and the real conversations. The $Distinct-n$ exhibits a measure of the diversity in the generated responses. In the human evaluation, three annotators are invited to rate a score in the range $[0, 1, 2]$ from three aspects *Fluency*, *Relevance*, and *Informativeness* on a random selection of 100 turns of dialogue which are from the test set. Finally, the average scores of the three annotators are used to represent the final result.

4.4 Implementation Details

We train the framework and baselines on 1 GPU NVIDIA 3090 24G card. We use pre-trained BERT [17] as the backbone model of the sentiment discriminator. Then, we use a linear layer for sentiment classification. We train our sentiment discriminator on the IMDB movie review dataset [40]. It is a dataset for binary sentiment classification and includes a set of 25,000 highly polar movie reviews for training and 25,000 for testing. After the sentiment has demonstrated an accuracy rate greater than 80% on test datasets, we propose to leverage it to infer the sentiment polarity of our review data. The dimension of the entity representation is set to 128, the Adam optimizer [29] is used to optimize the model, and the number of R-GCN network layers is set to 1. We use the DialoGPT-small model as the base model for generation, which is composed of 12 Transformer layers, and the embedding dimension is 768. The context and response generation maximum length are set to 256 and 50, respectively. During recommendation and generation training, we both set the learning rate to $1e-3$ and $1e-5$. Alignment loss μ_c was set to 0.8, and hyperparameter λ was set to 0.3. Gradient clipping limits gradient to $[0, 0.3]$. For every model, we use the same data split. For UCCR, since the method did not public data processing code for REDIAL and can not be used on the INSPIRED dataset, we processed the data according to their paper in the REDIAL dataset. For GPT-2 and DialoGPT, we used the small model. For other baselines, we reproduce their results by the CRSLab [92].

4.5 Result and Analysis (RQ1)

4.5.1 Recommendation. The experimental results of the recommendation mask on the two datasets are shown in Table 2. We observe that REDIAL model outperforms the classical recommendation model SASRec, which uses mentioned items in the conversation to accomplish recommendations, on the REDIAL dataset. However, on the INSPIRED dataset, REDIAL model performs worse than SASRec. It means that REDIAL model achieves worse performance on the INSPIRED dataset than the REDIAL dataset. This could be attributed to the REDIAL model heavily relying on items mentioned in conversations, whereas the items mentioned in the Inspired dataset are more sparse. TextCNN and BERT outperform REDIAL on both datasets, which might be because they utilize text information to model user preferences and the amount of text is more than items. Since BERT has been pre-trained on a large amount of text, BERT is better at language comprehension and achieves better performance than TextCNN on both datasets. KBRD introduces external KG and entities to achieve more accurate recommendations than REDIAL, BERT, SASRec, and TextCNN. KGSF fuses two different levels of KGs to achieve significant improvements. RevCore enhances user representations by using reviews to improve performance. TREA proposes a

Table 2. Evaluation Results of Recommendation (t -Test with p -Value < 0.05)

Dataset	Model	Recall@10	Recall@50	NDCG@10	NDCG@50	MRR@10	MRR@50
REDIAL	SASRec	0.045	0.119	0.022	0.038	0.015	0.019
	TextCNN	0.069	0.193	0.037	0.064	0.027	0.033
	BERT	0.138	0.308	0.074	0.112	0.055	0.063
	REDIAL	0.049	0.144	0.024	0.044	0.016	0.021
	KBRD	0.176	0.343	0.094	0.131	0.070	0.078
	KGSF	0.199	0.376	0.094	0.137	0.070	0.079
	RevCore	0.210	0.394	0.116	0.157	0.088	0.097
	UCCR	0.084	0.202	0.049	0.074	0.038	0.043
	TREA	0.212	0.416	0.117	0.162	0.088	0.097
	Baize	0.170	0.204	0.091	0.099	0.067	0.069
	Vicuna	0.184	0.240	0.103	0.116	0.078	0.081
	DeepSeek	0.108	0.165	0.057	0.071	0.0423	0.045
	CoCRS	0.234	0.424	0.129	0.171	0.097	0.106
INSPIRED	SASRec	0.131	0.262	0.083	0.112	0.069	0.075
	TextCNN	0.124	0.217	0.077	0.098	0.063	0.067
	BERT	0.169	0.320	0.095	0.127	0.080	0.087
	REDIAL	0.045	0.217	0.016	0.052	0.008	0.014
	KBRD	0.145	0.229	0.104	0.121	0.091	0.094
	KGSF	0.161	0.258	0.107	0.125	0.095	0.099
	RevCore	0.188	0.297	0.133	0.156	0.117	0.121
	UCCR	-	-	-	-	-	-
	TREA	0.203	0.316	0.140	0.165	0.121	0.126
	Baize	0.116	0.136	0.067	0.071	0.051	0.052
	Vicuna	0.171	0.202	0.107	0.115	0.088	0.090
	DeepSeek	0.120	0.141	0.074	0.080	0.060	0.061
	CoCRS	0.222	0.359	0.151	0.168	0.134	0.137
Reddit-Movie	SASRec	0.0198	0.0588	0.01082	0.01906	0.0081	0.0097
	TextCNN	0.0396	0.1174	0.0200	0.0367	0.0141	0.0175
	BERT	0.0510	0.1488	0.0184	0.0226	0.0260	0.0469
	REDIAL	0.0001	0.0012	0.00007	0.0003	0.00004	0.00008
	KBRD	0.0404	0.1139	0.0210	0.0368	0.0151	0.0184
	KGSF	0.0488	0.1325	0.0258	0.0438	0.0189	0.0226
	RevCore	0.0498	0.1318	0.0269	0.0446	0.0200	0.0236
	UCCR	-	-	-	-	-	-
	TREA	0.0394	0.1246	0.0194	0.0376	0.0135	0.0170
	Baize	0.0544	0.0683	0.0290	0.0325	0.0213	0.0222
	Vicuna	0.0562	0.0813	0.0320	0.0378	0.0246	0.0260
	DeepSeek	0.0581	0.0842	0.0314	0.0377	0.0232	0.0248
	CoCRS	0.0717	0.1925	0.0377	0.0635	0.0275	0.0328

The bold indicates the best performance.

Tree-structure reasoning approach to clarify the causal relationships between entities and improve the recommendation results.

As shown in Table 2, our CoCRS significantly outperforms all baselines on two datasets. The proposed approach demonstrates a remarkable improvement in performance compared to the strong baseline method TREA in all metrics. The superiority of the CoCRS approach can be attributed to

two factors: (1) it models the co-occurrence relationship between items in the review to capture meaningful associations and similarities, while also effectively incorporating the rich descriptions of the various entities referenced in relation to the items to enhance the semantic understanding of the item representations. (2) The proposed virtual similar user and time-aware modeling approach effectively leverages the co-occurrence relationships between users within the review and is able to more accurately model user preferences. Moreover, compared to RevCore [47], which is another review-based CRS, our CoCRS consistently achieves better results on all metrics. A potential explanation could be that RevCore simply carries some of the text from the reviews into the conversation ignoring the co-occurrence information in the reviews, which limits its ability to fully mine the provided information for the reviews. Although UCCR constructs a similar user selector, it only retrieves similar users at the word level. Our model finds similar users by the same sentiment polarity and constructs a virtual similar user to incorporate the co-occurrences into the CRS. Therefore, the performance of our model is better than UCCR. For TREA, our model introduces the latent information of review to enrich entity representations and user representations at multiple levels, so the effect of accurate recommendation is improved more.

In the Redial dataset, LLMs including Baize, Vicuna, and DeepSeek demonstrate superior performance over SASRec, TextCNN, and REDIAL across Recall@10, NDCG@10, and MRR@10 metrics, benefiting from their powerful language understanding and generation capabilities. However, none of them can outperform specialized CRS-designed models (e.g., RevCore). Notably, DeepSeek exhibits relatively weaker performance compared to Baize and Vicuna, likely due to its smaller model parameter. On the Inspired dataset, Baize demonstrates suboptimal performance due to the lengthy dialogue structure, whereas Vicuna and DeepSeek maintain robust comprehension capabilities for extended conversations, benefiting from their fine-tuning on conversational datasets. On the Inspired dataset, Baize exhibits relatively lower performance in handling lengthy conversations, while Vicuna and DeepSeek, which have been fine-tuned on conversational datasets, maintain strong comprehension capabilities even with long conversations.

As shown in Table 2, in the Reddit-Movie dataset, despite its large scale, the relatively short average dialogue rounds lead to suboptimal performance of CRS models, such as TREA, which relies on tree-structured reasoning of dialogue entities for user modeling. The limited conversation length prevents TREA from accurately capturing entity tree structures, resulting in performance degradation. In contrast, LLMs demonstrate robust language understanding capabilities that enable effective user preference modeling even with brief conversations. However, all these models underperform compared to CoCRS. CoCRS leverages review data to construct a Relation-aware Item-entity Heterogeneous Graph, enabling effective modeling of item co-occurrence relationships even in short conversation scenarios. Furthermore, the framework incorporates virtual similar users to augment user preference modeling. These design features demonstrate CoCRS's robust applicability across larger-scale datasets.

4.5.2 Generation.

Automatic Evaluation. We also evaluate the response generation task as a part of the automatic evaluation. The results of the automatic evaluation are shown in the Table 3. As evident from the evaluation results of the automatic metrics, our model exhibits the most favorable performance. Compared to the models that are not pre-trained, GPT-2 is significantly higher than the other models in Dist score because it has been pre-trained on a large amount of corpus. DialoGPT has a higher score in Dist score than GPT-2 due to its pre-training on dialogue data. The BLEU score of these PLMs is significantly lower than the other models on the REDIAL dataset. This is because the pre-trained models are not specifically designed for the generation task of CRS, and although they can generate more expressive responses, they do not necessarily match the ground truth. In the INSPIRED

Table 3. Evaluation Results of Response Generation (t -Test with p -Value < 0.05)

Dataset	Model	Dist-2	Dist-3	Dist-4	BLEU-2	BLEU-3
REDIAL	REDIAL	0.0548	0.1905	0.4243	0.0226	0.0069
	KBRD	0.0687	0.2562	0.4148	0.0237	0.0087
	KGSF	0.0756	0.3020	0.5190	0.0240	0.0090
	RevCore	0.2085	0.3348	0.5117	0.0250	0.0105
	UCCR	0.0818	0.3289	0.5635	0.0257	0.0106
	TREA	0.2197	0.5560	0.8378	0.0280	0.0137
	GPT-2	0.7029	1.1230	1.4299	0.0106	0.0052
	DialoGPT	0.7206	1.1878	1.5412	0.0121	0.0056
	CoCRS	0.8679	1.3667	1.6791	0.0310	0.0171
INSPIRED	REDIAL	0.4060	1.2260	2.2050	0.0013	0.0001
	KBRD	0.5670	2.0170	3.6210	0.0205	0.0064
	KGSF	0.6080	2.5190	4.9290	0.0196	0.0049
	RevCore	3.0850	4.6020	5.3410	0.0202	0.0112
	UCCR	-	-	-	-	-
	TREA	1.6420	3.8480	6.2530	0.0206	0.0079
	GPT-2	3.8822	5.6312	6.4166	0.0211	0.0104
	DialoGPT	3.9062	5.7562	6.7025	0.0216	0.0107
	CoCRS	4.6667	6.4334	6.8885	0.0263	0.0144
Reddit-Movie	REDIAL	0.0083	0.0105	0.0115	0.0475	0.0225
	KBRD	0.0407	0.1833	0.3636	0.0585	0.0173
	KGSF	0.0253	0.0260	0.0249	0.1004	0.0514
	RevCore	0.0713	0.0785	0.0788	0.0988	0.0515
	UCCR	-	-	-	-	-
	TREA	0.0192	0.0246	0.0310	0.0109	0.0018
	GPT-2	0.4421	0.6880	0.8654	0.0477	0.0386
	DialoGPT	0.6149	0.8947	1.0674	0.2268	0.2084
	CoCRS	0.8547	1.2512	1.4503	0.2450	0.2315

The bold indicates the best performance.

dataset, the PLMs have a good BLEU score but still lower than CoCRS, because the INSPIRED is a small-size dataset and has less noise, which makes it easier for PLMs to learn and generate more accurate responses. This indicates that the pre-trained model cannot be directly used for the response generation task of CRS. On the Reddit-Movie dataset, TREA demonstrates notably poor performance in generation tasks due to the short conversation length. Other models exhibit trends consistent with their performance observed in both the Redial and Inspired datasets. Instead, our model adds specific prompt words to the dialogue and fine-tunes the model to make it more suitable for the CRS response generation task. Thus, it can generate more diverse and matching responses.

Human Evaluation. The results of the human evaluation for response generation are reported in Table 4. Compared to other baselines, CoCRS exhibits the most favorable performance. This indicates that the prompt words and fine-tuning help guide the PLM to adapt to the response generation task in CRS. The prompt words enable the PLM to clarify its role in CRS and the fine-tuning enables the PLM to adapt to the response-generation task of CRS to generate more fluent, relevant, and informative responses.

4.6 Ablation Study (RQ2)

To evaluate the effectiveness of various components in our model, we conducted an ablation study. We focused on the following parts and performed ablation experiments: (1) *w/o RG*: Removal

Table 4. Human Evaluation Results on the REDIAL Dataset

Model	Relevance	Fluency	Informativeness
RevCore	1.153	1.880	1.076
UCCR	1.264	1.623	1.068
TREA	1.222	1.931	1.059
GPT2	1.572	1.709	1.239
DialoGTP	1.538	1.743	1.290
CoCRS	1.726	1.940	1.367

The bold indicates the best performance.

Table 5. Ablation Results of Recommendation (t -Test with p -Value < 0.05)

Dataset	Model	Recall@10	Recall@50	NDCG@10	NDCG@50	MRR@10	MRR@50
REDIAL	CoCRS	0.2340	0.4241	0.1291	0.1714	0.0971	0.1063
	<i>w/o RG</i>	0.2195	0.4119	0.1200	0.1616	0.0884	0.0973
	<i>w/o SU</i>	0.2212	0.4202	0.1223	0.1666	0.0924	0.1020
	<i>w/o TA</i>	0.2167	0.4071	0.1172	0.1595	0.0870	0.0962
INSPIRED	CoCRS	0.2223	0.3554	0.1537	0.1844	0.1346	0.1409
	<i>w/o RG</i>	0.2187	0.3200	0.1525	0.1743	0.1317	0.1361
	<i>w/o SU</i>	0.2188	0.3515	0.1460	0.1749	0.1235	0.1294
	<i>w/o TA</i>	0.2187	0.3125	0.1408	0.1612	0.1164	0.1206
Reddit-Movie	CoCRS	0.0717	0.1925	0.0377	0.0635	0.0275	0.0328
	<i>w/o RG</i>	0.0670	0.1761	0.0347	0.0582	0.0247	0.0295
	<i>w/o SU</i>	0.0690	0.1801	0.0351	0.0591	0.0249	0.0298
	<i>w/o TA</i>	0.0699	0.1846	0.0365	0.0593	0.0260	0.0300

The bold indicates the best performance.

of relation-aware item-entity heterogeneous graph. (2) *w/o SU*: Removal of virtual similar user. (3) *w/o TA*: Removal of time-aware preferences modeling and retaining the conversation history modeling. The experimental results are shown in the Table 5. It can be observed that removing these modules has an impact on the performance of the model, indicating that our proposed approach is effective in improving the model's performance. In particular, *w/o TA* shows a sharp decline in performance for all metrics. This observation highlights the importance of modeling users from two different time scales. By incorporating the different time-scales user representation, CoCRS is able to effectively model the users' real-time preferences and interests, which tend to evolve over time. In contrast, a system that solely relies on the conversation history may struggle to fully capture the nuanced and time-varying user preferences. The performance of CoCRS *w/o SU* has declined. This indicates that the construction of virtual similar users can help address the mismatch between review users and dialogue users. By incorporating the user preference information including the co-occurrence relationships in reviews into the current dialogue context, the proposed approach is able to enhance the performance of the CRS. Besides, there is also a decrease in the performance of *w/o RG*, which demonstrates that indeed effective in extracting the co-occurrence relationships between items from the review, and utilizing this information to supplement the item representations.

In the Reddit-Movie dataset, the relatively short conversation turns result in a significant overlap between the conversation history and current conversation, leading to a smaller performance drop in the *w/o TA*. The *w/o SU* shows the next most moderate decline, as the brief conversations mean some conversation users lack similar users, thereby limiting this module's effectiveness. Notably, the *w/o RG* exhibits the most substantial performance degradation, which further validates the effectiveness of our approach in mining co-occurrence information from reviews.

4.7 Results on Cold-start Scenarios (RQ3)

The cold-start problem is a critical issue that is widely prevalent in the context of Recommender Systems. This challenge arises when the system is required to provide recommendations to new users or users with limited interaction history. In a typical CRS scenario, many users will engage the system to request item recommendations within the first few rounds of the conversation. As such, it is essential to evaluate the system's performance in cold-start situations. To effectively simulate and study the cold-start problem, we consider the situation where the number of dialogue rounds between the user and the CRS is restricted. Importantly, it is recognized that the number of available dialogue rounds may vary across different cold-start users, depending on their individual cases and behaviors. Therefore, we simulate the cold-start scenario by considering the dialogue history ranging from the currently recommended item up to the previous k rounds ($k = 0, 1, 2, 3, 4, 5$). In the context of conversational interactions, a single round is defined as one question from the user followed by one response from the system. When $k = 0$, it is a cold-start user with no historical dialogue information. By exploring this range of dialogue history lengths, the research is able to systematically investigate the CRS performance under varying degrees of cold-start users, which is a key characteristic of the cold-start problem.

The results of the cold-start scenario evaluation are presented in Figure 2. The findings demonstrate that the proposed CoCRS approach outperforms the TREA baseline across the range of dialogue history lengths examined. Notably, when the number of dialogue rounds is limited, the TREA method exhibits relatively poor performance. This can be attributed to the fact that TREA constructs a tree-structured representation to model the user's preferences by reasoning about the relationships between the entities mentioned in the conversation. However, when the available dialogue history is scarce, TREA struggles to capture valid and meaningful structures from the limited entity sequence, which ultimately impairs its ability to provide accurate recommendations in the cold-start setting. The proposed CoCRS approach leverages the co-occurrence relationships between items mentioned in reviews to enrich the item representations, thereby incorporating valuable prior information into the item embeddings. As a result, even in the cold-start scenario with $k = 0$, where there is no contextual information available, CoCRS is still able to provide relatively accurate recommendations. As the dialogue history increases ($k > 0$), the interactive information contained in the conversation also grows. CoCRS is able to leverage the sentiment polarity of the items mentioned in the dialogue and user co-occurrences in reviews to construct a virtual user representation, supplementing the user's preferences. Consequently, the performance of CoCRS improves as the dialogue history becomes more extensive. When the dialogue history is longer (such as $k > 4$), the time-aware mechanism employed by CoCRS models the user's preferences from two different time scales, allowing the system to maintain its strong performance even with a longer dialogue history.

4.8 Hyperparameter Experiments (RQ4)

In this section, we conduct an investigation into the influence of hyperparameters λ and μ associated with two loss functions on CoCRS.

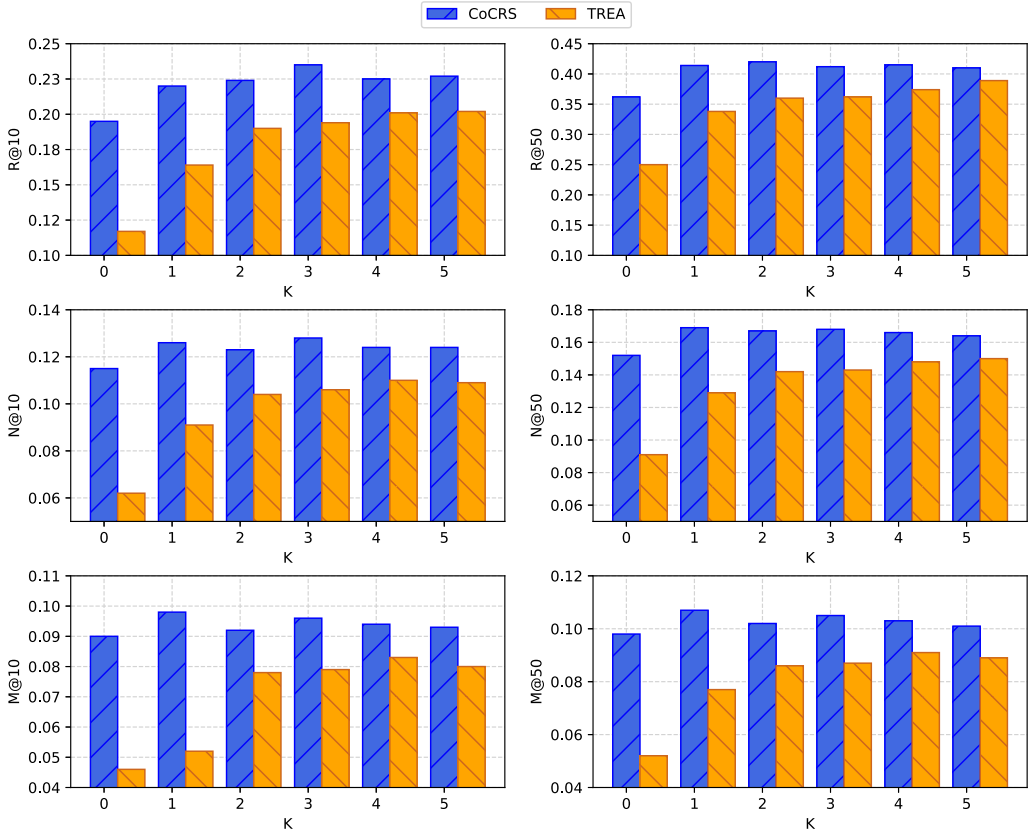


Fig. 2. Results of cold-start scenarios on the ReDial dataset.

The Weight λ of Alignment Loss. Based on the observations derived from the first row of the Figure 3, we can find that when λ is set to 1, all the evaluation metrics attain their optimal values. Conversely, when λ is set to 0, indicating that the alignment loss is not used to guide the model, the performance deteriorates significantly. Additionally, the model's performance demonstrates an increasing trend as λ increases. We speculate that a substantial semantic gap exists between words and entities, and the utilization of alignment loss effectively aligns their respective semantic spaces, thereby mitigating this disparity. Consequently, as λ grows larger, the alignment loss assumes a more critical role in enhancing the model's performance.

The Weight μ of Isolation Loss. As illustrated in the second row of the Figure 3, we can observe that a low value of μ , i.e., 0.3, is sufficient for CoCRS to achieve satisfactory performance. In contrast, increasing the value of μ leads to a decline in performance. This phenomenon can be attributed to isolation loss, which facilitates the model's more distinct perception of recent conversations. When μ is set to a larger value, the model excessively prioritizes recent conversations, disregarding the history conversation. It hampers the effective modeling of user preferences from two-time scales. Besides, setting μ to 0, which removes the isolation loss, results in CoCRS being unable to differentiate between the conversation history and recent conversation, leading to a decline in performance. Accordingly, we empirically set μ to 0.3.

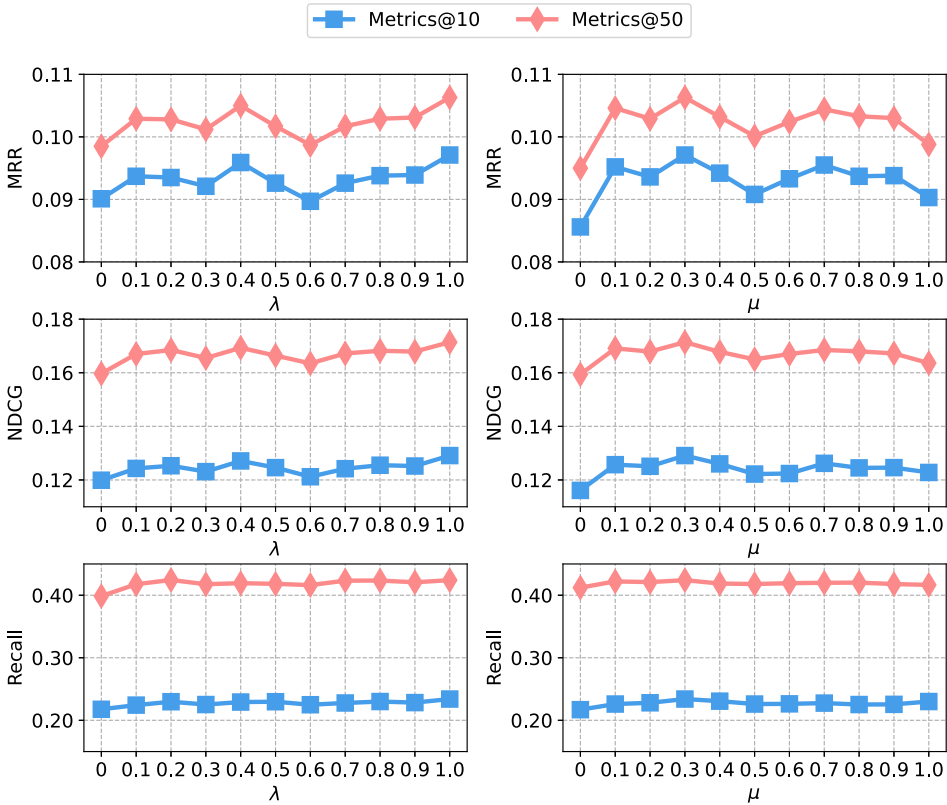


Fig. 3. Hyperparameter experiments.

4.9 Fusion Sequence Effect (RQ5)

In this section, we conduct experiments to investigate the impact of the fusion order among the various user representations on the performance of the CoCRS model. The user representations considered include the word-level user representation u_c , the virtual similar user representation u_s , the user representation u_h based on the conversation history, and the user representation u_p based on the recent conversation. We use the notations W , C , S , and R to represent u_c , u_h , u_s , and u_p , respectively. The order of appearance indicates the fusion order. For instance, *RCSW* means that we first fuse u_p and u_h using a gate, then fuse the result with u_s , and finally fuse with u_h . Due to the constraints of the time-aware mechanism employed by CoCRS, u_h and u_p need to be fixed. Consequently, there are a total of 3 fusion orders to explore. Additionally, to investigate the impact of disrupting the time-aware mechanism, we also conducted an experiment with the *SCWR* order. The experimental results for these fusion order configurations are presented in the Table 6.

Based on the experimental results, we observe that the *RCSW* configuration achieved the best performance. This seems reasonable since the time-aware mechanism employed by CoCRS models the user preferences across multiple temporal scales using the entities and items mentioned in the dialogue. Compared with the word-level representation u_c , the virtual similar user representation u_s and the output of the time-aware mechanism have a smaller semantic gap, as u_s is constructed using the sentiment polarity of the items. Therefore, the fusion of *RCSW* is more effective. In the *RCWS* order, the fusion of u_c and the output of the time-aware mechanism happens first. Since

Table 6. Fusion Sequence on the REDIAL Datasets (t -Test with p -Value < 0.05)

Model	Recall@10	Recall@50	NDCG@10	NDCG@50	MRR@10	MRR@50
<i>RCSW</i>	0.2340	0.4241	0.1291	0.1714	0.0971	0.1063
<i>RCWS</i>	0.2241	0.4159	0.1229	0.1654	0.0921	0.1012
<i>WSRC</i>	0.2195	0.4227	0.1223	0.1672	0.0927	0.1023
<i>SCWR</i>	0.2125	0.4015	0.1157	0.1577	0.0863	0.0954

The bold indicates the best performance.

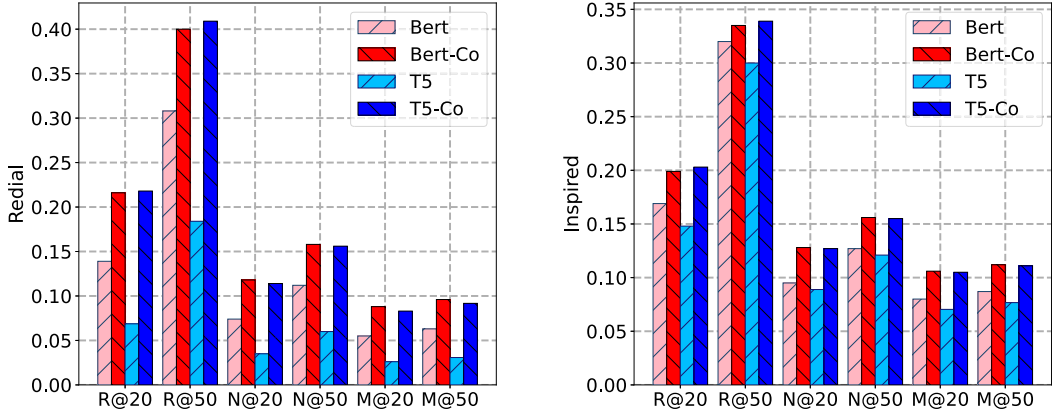


Fig. 4. BERT and T5 with co-occurrences on both datasets.

the semantic gap between the entity-level and word-level representations is larger, it leads to a decline in performance. For the *WSRC* order, the fusion of u_c and u_s happens first. Although there is still a semantic gap, it is smaller compared to the *RCWS* case. This seems reasonable since u_s is constructed based on the sentiment polarity of the items in the reviews and dialogue. So u_s is similar to u_c . Consequently, the performance of *WSRC* is better than *RCWS*. Finally, the *SCWR* order disrupts the time-aware fusion of u_h and u_p . Hence, it exhibits the worst performance among the configurations.

4.10 LLM With Co-occurrences (RQ6)

In this section, we validate BERT and T5 with Co-occurrences on two datasets. Additionally, as a comparison, we directly input the conversation history into BERT and T5 for recommendation. Specifically, we respectively use BERT and T5 as encoders and input the entire conversation into the encoder. Then, we put the output of the encoder into a linear layer to calculate the recommendation probability for each item. Finally, we choose items for recommendation based on probabilities in descending order. For incorporating co-occurrences into BERT and T5, we also use them as encoders. Then, we utilize the output of the encoder to replace the word-level user representation in CoCRS, while the rest follows the flow of CoCRS. The experimental results are illustrated in the Figure 4.

The superior performance of BERT over T5 is evident across both datasets. This discrepancy can be attributed to T5's expertise in converting questions into textual formats. We involve directly employing T5 as an encoder and connecting it to downstream linear layers for recommendation, failing to fully exploit the capabilities of T5. In contrast, BERT which is the encoder-only structure proves more effective in capturing user preferences. Following the integration of co-occurrence

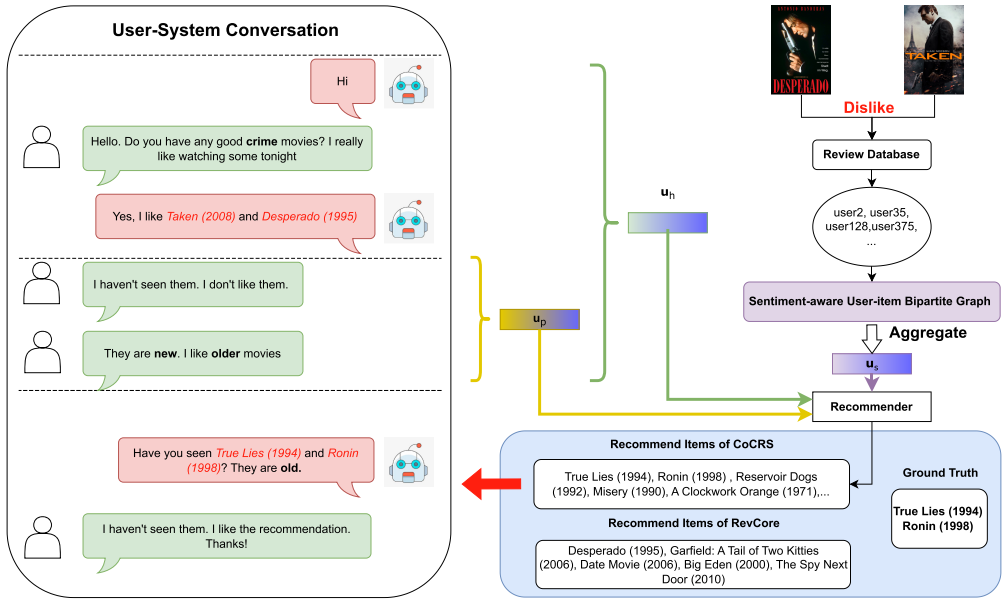


Fig. 5. A case study for the CoCRS process.

information, both LLMs exhibit substantial performance improvements. This observation highlights the beneficial role of co-occurrence information in assisting LLMs in learning the similarity relationships among items and modeling user preferences. Despite LLMs' remarkable language abilities, their performance in recommendation systems remains suboptimal. Nevertheless, by incorporating co-occurrence relationships, we can overcome the limitations of LLMs in this domain and construct a more potent LLM specifically designed for recommendation tasks.

4.11 Case Study (RQ7)

To illustrate the ability of CoCRS to generate high-quality recommendations, we present a case, which includes the workflow of CoCRS and the recommendation results of CoCRS and RevCore, on the REDIAL dataset. As shown in Figure 5, when the user requests service from the system and expresses a preference for “*crime*” movies, the system initially recommends two crime movies, “*Taken*” and “*Desperado*.” However, the user expresses dislike for these two movies. In response to this feedback, the system identifies users who dislike these movies from the review database and retrieves corresponding user embeddings from the sentiment-aware user-item bipartite graph. Subsequently, the system aggregates these embeddings to generate a virtual user representation. Then, the user explains the reasons for dislike and expresses a preference for “*old*” movies. At this point, the system respectively models user preferences based on entities and movies mentioned in the entire conversation history and the current episode. Based on the above processing, the system accurately recommends the movie, i.e., “*True Lies*” and “*Ronin*,” which is consistent with the ground truth. Finally, the system generates the response to reply to the user.

From the recommended results, it is evident that CoCRS suggests exclusively “*crime*” movies, demonstrating the system’s accurate capture of the user’s preference for “*crime*.” Furthermore, CoCRS’s time-aware method precisely identifies that the user prefers older movies. Consequently, the movies recommended by CoCRS have relatively earlier production dates. However, the top movie recommended by RevCore has appeared in the conversation history. This indicates that while

RevCore has learned about the user's preference for crime movies, it has not captured the user's dislikes and preference for older movies. Additionally, RevCore's recommendation list includes movies of other genres, such as comedy. The possible reason is that the method of RevCore, which simply moves words and entities in the reviews to the end of the conversation, introduces some noise about "comedy". Therefore, these results validate the effectiveness of our proposed approach CoCRS in accurately modeling user preferences.

4.12 Complexity Discussion

The computational complexity of our method in primarily consists of R-GCN operations on graph structures, finding similar users, virtual similar user representation, time-aware user representation and GCN operations on ConceptNet. Following prior works [38, 47, 93], we use a single-layer R-GCN, layer $L = 1$, for graph encoding. Therefore, the time complexity of our approach can be simplified as:

$$O((|R_{KG}| \cdot |E_{KG}| + |V_{KG}| + |E_{G_R}| + |V_{G_R}| + |E_{G_S}| + |V_{G_S}| + N_c + n_{U_S} + L_{his} + L_{cur}) \cdot d^2), \quad (24)$$

where R_{KG} denotes the number of distinct relation types in KG, E_{KG} and V_{KG} denote the number of edges and nodes in KG, E_{G_R} and E_{G_S} denote the number of edge in G_R and G_S , V_{G_R} and V_{G_S} denote the number of node in G_R and G_S , d is embedding dimension. where R denotes the number of distinct relation types in KG, n_{U_S} denotes the number of similar users, L_{his} and L_{cur} denote the number of entities in conversation history and in current conversation, N_c and E_c denote the number of nodes and edges in ConceptNet, d is embedding dimension.

Additionally, we have computed the time complexity for both TREA and RevCore. The time complexity of TREA is given as follows:

$$O((|R_{KG}| \cdot |E_{KG}| + |V_{KG}| + N_c + L_{bran}) \cdot d^2 + |E_c| \cdot d), \quad (25)$$

where L_{bran} denotes the construction of the reasoning tree. The time complexity of RevCore is given as follows:

$$O((|R_{KG}| \cdot |E_{KG}| + |V_{KG}| + N_c + L_{his}) \cdot d^2 + |E_c| \cdot d). \quad (26)$$

The computational complexity is primarily dominated by KG, as the dimensionality d is typically constrained to a limited size (e.g., 128), while the scale of relationship R_{KG} and edge E_{KG} products in the KG exceeds that of G_R and G_S , and other components. Furthermore, the G_R and G_S construction process is completed once during the pre-processing stage and does not participate in training, thus it does not contribute to the computational complexity. Therefore, the additional computational complexity introduced by our model is acceptable while simultaneously achieving superior performance.

5 Conclusion

In this article, we propose a framework to mine the Co-occurrence relationship in review and model users from two-time scales to improve CRS. Our framework aims to augment the entity representations with co-occurrence relationship information and model user preference with the virtual similar user and time-aware user representation. In particular, we construct a relation-aware item-entity heterogeneous graph for item representations and a sentiment-aware user-item bipartite graph to generate a virtual similar user. Then we model the user preference by the virtual similar user and the time-aware mechanism which captures user preferences from the two-time dimensions of conversation history and recent conversations. Extensive experimental results show that our method outperforms several baselines.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. arXiv:2303.08774. Retrieved from <https://arxiv.org/abs/2303.08774>
- [2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference*. Springer, 722–735.
- [3] Daniel Billsus and Michael J. Pazzani. 1998. Learning collaborative information filters. In *Proceedings of the 15th International Conference on Machine Learning (ICML '98)*, 46–54.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, Vol. 33, 1877–1901.
- [5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv:2303.12712. Retrieved from <https://arxiv.org/abs/2303.12712>
- [6] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple yet effective graph contrastive learning for recommendation. In *Proceedings of the 11th International Conference on Learning Representations*, 1–15.
- [7] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*, 1583–1592.
- [8] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 1803–1813.
- [9] 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, (DLRS '16)*. ACM, New York, NY, 7–10.
- [10] Zhiyong Cheng, Fan Liu, Shenghan Mei, Yangyang Guo, Lei Zhu, and Liqiang Nie. 2022. Feature-Level attentive ICF for recommendation. *ACM Transactions on Information Systems* 40, 4, Article 75 (Mar. 2022), 24 pages.
- [11] Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. Retrieved April 14, 2023 from <https://vicuna.lmsys.org>.
- [12] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 815–824.
- [13] Zhendong Chu, Hongning Wang, Yun Xiao, Bo Long, and Lingfei Wu. 2023. Meta policy learning for Cold-Start conversational recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, New York, NY, 222–230.
- [14] Justin Cui, Kai Dicarantonio, Sara Kemper, Kathy Lin, Danjie Tang, Anton Korikov, and Scott Sanner. 2024. Retrieval-augmented conversational recommendation with prompt-based semi-structured natural language state tracking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, 2786–2790.
- [15] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, 121–124.
- [16] Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. 2023. A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Transactions on Information Systems* 41, 3, Article 77 (Feb. 2023), 25 pages.
- [17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol 1 (Long and Short Papers), Jill Burstein, Christy Doran, and Tamar Solorio (Eds.), Association for Computational Linguistics, 4171–4186.
- [18] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. 2023. Leveraging large language models in conversational recommender systems. arXiv:2305.07961. Retrieved from <https://arxiv.org/abs/2305.07961>
- [19] Mingsheng Fu, Hong Qu, Zhang Yi, Li Lu, and Yongsheng Liu. 2018. A novel deep learning-based collaborative filtering model for recommendation system. *IEEE Transactions on Cybernetics* 49, 3 (2018), 1084–1096.
- [20] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1725–1731.

- [21] Shuyu Guo, Shuo Zhang, Weiwei Sun, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2023. Towards explainable conversational recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 2786–2795.
- [22] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, Vol. 30, 1–11.
- [23] Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyang Shi, and Zhou Yu. 2020. INSPIRED: Toward sociable recommendation dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8142–8152.
- [24] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. ACM, New York, NY, 639–648.
- [25] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 720–730.
- [26] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.
- [27] Heeseon Kim, Hyeongjun Yang, and Kyong-Ho Lee. 2023. Confident action decision via hierarchical policy learning for conversational recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, New York, NY, 1386–1395.
- [28] Yoon. Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, 1746–1751.
- [29] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 1–15.
- [30] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*, 1–14.
- [31] Yehuda Koren. 2008. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, 426–434.
- [32] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 447–456.
- [33] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [34] Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. In *Recommender Systems Handbook*. Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.), Springer, 91–142.
- [35] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 304–312.
- [36] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*, Vol. 31 (2018), 1–11.
- [37] Shuokai Li, Ruobing Xie, Yongchun Zhu, Xiang Ao, Fuzhen Zhuang, and 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*, 223–233.
- [38] Wendi Li, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Ye Yuan, Wenfeng Xie, and Dangyang Chen. 2023. TREA: Tree-Structure reasoning schema for conversational recommendation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.), Association for Computational Linguistics, 2970–2982.
- [39] Allen Lin, Ziwei Zhu, Jianling Wang, and James Caverlee. 2023. Enhancing user personalization in conversational recommenders. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, New York, NY, 770–778.
- [40] Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (Eds.). 2011. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Association for Computational Linguistics. Retrieved from <https://aclanthology.org/P11-2000.pdf>.
- [41] Dongding Lin, Jian Wang, and Wenjie Li. 2023. COLA: Improving conversational recommender systems by collaborative augmentation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence and 35th Conference on Innovative Applications of Artificial Intelligence and 13th Symposium on Educational Advances in Artificial Intelligence (AAAI '23/IAAI '23/EAAI '23)*, 4462–4470.

- [42] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*, 2320–2329.
- [43] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. DeepSeek-V3 technical report. arXiv:2412.19437. Retrieved from <https://arxiv.org/abs/2412.19437>
- [44] Kang Liu, Feng Xue, Dan Guo, Le Wu, Shujie Li, and Richang Hong. 2023. MEGCF: Multimodal entity graph collaborative filtering for personalized recommendation. *ACM Trans. Inf. Syst.* 41, 2, Article 30 (Apr. 2023), 27 pages.
- [45] Yuanxing Liu, Weinan Zhang, Baohua Dong, Yan Fan, Hang Wang, Fan Feng, Yifan Chen, Ziyu Zhuang, Hengbin Cui, Yongbin Li, and Wanxiang Che. 2023. U-NEED: A fine-grained dataset for user Needs-Centric E-commerce conversational recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 2723–2732.
- [46] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive learning for recommender system. arXiv:2101.01317. Retrieved from <https://arxiv.org/abs/2101.01317>
- [47] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-Augmented Conversational Recommendation. In *Proceedings of the Findings of the Association for Computational Linguistics (ACL-IJCNLP '21)*. Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.), Association for Computational Linguistics, 1161–1173.
- [48] Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2021. CR-Walker: Tree-Structured graph reasoning and dialog acts for conversational recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.), Association for Computational Linguistics, 1839–1851.
- [49] Andriy Mnih and Russ R. Salakhutdinov. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, Vol. 20, 1–8.
- [50] Yuxin Ni, Yunwen Xia, Hui Fang, Chong Long, Xinyu Kong, Daqian Li, Dong Yang, and Jie Zhang. 2023. Meta-CRS: A dynamic Meta-Learning approach for effective conversational recommender system. *ACM Transactions on Information Systems* 42, 1, Article 28 (Aug. 2023), 27 pages.
- [51] Xinghua Qu, Hongyang Liu, Zhu Sun, Xiang Yin, Yew Soon Ong, Lu Lu, and Zejun Ma. 2023. Towards building voice-based conversational recommender systems: Datasets, potential solutions and prospects. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 2701–2711.
- [52] Alec Radford, Jeffrey Wu, Rewon Luan, David Luan, Dario Amodei, Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
- [53] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, 995–1000.
- [54] 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 *Proceedings of the Semantic Web: 15th International Conference (ESWC '18)*. Springer, 593–607.
- [55] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI '16)*, 3776–3783.
- [56] 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 *Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI '17)*, 3295–3301.
- [57] Chenzhan Shang, Yupeng Hou, Wayne Xin Zhao, Yaliang Li, and Jing Zhang. 2023. Multi-grained hypergraph interest modeling for conversational recommendation. *AI Open* 4 (2023), 154–164.
- [58] Pengyang Shao, Le Wu, Kun Zhang, Defu Lian, Richang Hong, Yong Li, and Meng Wang. 2024. Average User-Side counterfactual fairness for collaborative filtering. *ACM Transactions on Information Systems* 42, 5, Article 140 (May 2024), 26 pages.
- [59] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-Chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, Vol. 28, 1–9.
- [60] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. 2022. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1283–1293.
- [61] Clemencia Siro, Mohammad Aliannejadi, and Maarten De Rijke. 2023. Understanding and predicting user satisfaction with conversational recommender systems. *ACM Transactions on Information Systems* 42, 2, Article 55 (Nov. 2023), 37 pages.

- [62] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI '17)*, 4444–4451.
- [63] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *Proceedings of the The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 235–244.
- [64] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference*, 729–739.
- [65] 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 *Advances in Neural Information Processing Systems*, Vol. 30, 1–11.
- [66] Paul Viola and William M. Wells, III. 1997. Alignment by maximization of mutual information. *International Journal of Computer Vision* 24, 2 (1997), 137–154.
- [67] Ke Wang, Yanmin Zhu, Tianzi Zang, Chunyang Wang, Kuan Liu, and Peibo Ma. 2023. Multi-aspect graph contrastive learning for review-enhanced recommendation. *ACM Transactions on Information Systems* 42, 2 (2023), 1–29.
- [68] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165–174.
- [69] Xi Wang, Hossein Rahmani, Jiqun Liu, and Emine Yilmaz. 2023. Improving conversational recommendation systems via bias analysis and language-model-enhanced data augmentation. In *Proceedings of the Findings of the Association for Computational Linguistics (EMNLP '23)*. Houda Bouamor, Juan Pino, and Kalika Bali (Eds.), Association for Computational Linguistics, 3609–3622.
- [70] Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the evaluation for conversational recommendation in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Houda Bouamor, Juan Pino, and Kalika Bali (Eds.), Association for Computational Linguistics, 10052–10065.
- [71] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 726–735.
- [72] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A Context-Aware User-Item representation learning for item recommendation. *ACM Transactions on Information Systems* 37, 2, Article 22 (Jan. 2019), 29 pages. DOI : <https://doi.org/10.1145/3298988>
- [73] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 70–79.
- [74] Zhouhang Xie, Junda Wu, Hyunsi Jeon, Zhankui He, Harald Steck, Rahul Jha, Dawen Liang, Nathan Kallus, and Julian McAuley. 2024. Neighborhood-Based collaborative filtering for conversational recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys '24)*. ACM, New York, NY, 1045–1050. DOI : <https://doi.org/10.1145/3640457.3688191>
- [75] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. arXiv:2304.01196. Retrieved from <https://arxiv.org/abs/2304.01196>
- [76] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, and Edith C. H. Ngai. 2024. FourierKAN-GCF: Fourier Kolmogorov-Arnold network – An effective and efficient feature transformation for graph collaborative filtering. arXiv:2406.01034. Retrieved from <https://arxiv.org/abs/2406.01034>
- [77] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the 35th International Conference on Machine Learning*. Jennifer Dy and Andreas Krause (Eds.), Vol. 80, PMLR, 5453–5462.
- [78] Senrong Xu, Liangyue Li, Zenan Li, Yuan Yao, Feng Xu, Zulong Chen, Quan Lu, and Hanghang Tong. 2023. On the vulnerability of graph learning-based collaborative filtering. *ACM Transactions on Information Systems* 41, 4, Article 87 (Mar. 2023), 28 pages.
- [79] Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2022. Improving Conversational Recommendation Systems' Quality with Context-Aware Item Meta-Information. In *Proceedings of the Findings of the Association for Computational Linguistics (NAACL '22)*. Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.), Association for Computational Linguistics, 38–48.
- [80] Dayu Yang, Fumian Chen, and Hui Fang. 2024. Behavior alignment: A new perspective of evaluating LLM-based conversational recommendation systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. ACM, New York, NY, 2286–2290.

- [81] An Zhang, Wenchang Ma, Jingnan Zheng, Xiang Wang, and Tat-Seng Chua. 2024. Robust collaborative filtering to popularity distribution shift. *ACM Transactions on Information Systems* 42, 3, Article 80 (Jan. 2024), 25 pages.
- [82] Lu Zhang, Chen Li, Yu Lei, Zhu Sun, and Guanfang Liu. 2024. An empirical analysis on multi-turn conversational recommender systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. ACM, New York, NY, 841–851.
- [83] Xiaoyu Zhang, Ruobing Xie, Yougang Lyu, Xin Xin, Pengjie Ren, Mingfei Liang, Bo Zhang, Zhanhui Kang, Maarten de Rijke, and Zhaochun Ren. 2024. Towards empathetic conversational recommender systems. In *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys '24)*. ACM, New York, NY, 84–93. DOI: <https://doi.org/10.1145/3640457.3688133>
- [84] Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2023. Variational reasoning over incomplete knowledge graphs for conversational recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 231–239.
- [85] Xiaokun Zhang, Bo Xu, Zhaochun Ren, Xiaochen Wang, Hongfei Lin, and Fenglong Ma. 2024. Disentangling ID and modality effects for session-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1883–1892.
- [86] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 177–186.
- [87] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT: Large-Scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Asli Celikyilmaz and Tsung-Hsien Wen (Eds.), Association for Computational Linguistics, 270–278.
- [88] Sen Zhao, Wei Wei, Xian-Ling Mao, Shuai Zhu, Minghui Yang, Zujie Wen, Danyang Chen, and Feida Zhu. 2023. Multi-view hypergraph contrastive policy learning for conversational recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 654–664.
- [89] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. arXiv:2303.18223. Retrieved from <https://arxiv.org/abs/2303.18223>
- [90] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, 425–434.
- [91] Yongsen Zheng, Ruilin Xu, Ziliang Chen, Guohua Wang, Mingjie Qian, Jinghui Qin, and Liang Lin. 2024. HyCoRec: Hypergraph-enhanced multi-preference learning for alleviating Matthew effect in conversational recommendation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.), Association for Computational Linguistics, 2526–2537. DOI: <https://doi.org/10.18653/v1/2024.acl-long.138>
- [92] 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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, 185–193.
- [93] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1006–1014.
- [94] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided conversational recommender system. In *Proceedings of the 28th International Conference on Computational Linguistics*. Donia Scott, Nuria Bel, and Chengqing Zong (Eds.), International Committee on Computational Linguistics, 4128–4139.
- [95] Xin Zhou, Aixin Sun, Yong Liu, Jie Zhang, and Chunyan Miao. 2023. Selfcf: A simple framework for self-supervised collaborative filtering. *ACM Transactions on Recommender Systems* 1, 2 (2023), 1–25.
- [96] Yao Zhou, Haonan Wang, Jingrui He, and Haixun Wang. 2025. Review-Based explainable recommendations: A transparency perspective. *ACM Transactions on Recommender Systems* 3, 3, Article 34 (Mar. 2025), 20 pages. DOI: <https://doi.org/10.1145/3701762>
- [97] Yuanhang Zhou, Kun Zhou, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and He Hu. 2022. C²-CRS: Coarse-to-fine contrastive learning for conversational recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1488–1496.

- [98] Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025. Collaborative retrieval for large language model-based conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, 3323–3334.
- [99] Jie Zou, Cheng Lin, Weikang Guo, Zheng Wang, Jiwei Wei, Yang Yang, and Heng Tao Shen. 2025. Multi-type context-aware conversational recommender systems via mixture-of-experts. *Information Fusion* 126 (2025), 103638.

Received 29 August 2024; revised 16 September 2025; accepted 26 September 2025