Improving Conversational Recommendation with Contextual Adaptation of External Recommenders and LLM-based Reranking

Chuang Li[1⋆[0009−0006−8112−3505]], Weida Liang[1], Hengchang Hu[1], See-Kiong Ng[1], Min-Yen Kan[1], Haizhou Li[1,3], and Yang Deng[1,2]

[1] National University of Singapore, Singapore
[2] Singapore Management University, Singapore
[3] Chinese University of Hong Kong, Shenzhen
{lichuang, weida_liang, hengchanghu}@u.nus.edu
{seekiong, kanmy, haizhou.li, ydeng}@nus.edu.sg

Abstract. We tackle the challenge of integrating large language models (LLMs) with external recommender systems to enhance domain expertise in conversational recommendation (CRS). Current LLM-based CRS approaches primarily rely on zero/few-shot methods for generating item recommendations based on user queries, but this method faces two significant challenges: (1) without domain-specific adaptation, LLMs frequently recommend items not in the target item space, resulting in low recommendation accuracy; and (2) LLMs largely rely on dialogue context for content-based recommendations, neglecting the collaborative relationships among item sequences. To address these limitations, we introduce the *CARE (Contextual Adaptation of Recommenders)* framework. CARE (a) integrates external recommender systems as domain experts, producing candidate items through entity-level insights, and (b) customizes LLMs as rerankers to enhance the accuracy by leveraging contextual information. Our results demonstrate that incorporating CARE framework significantly enhances recommendation accuracy of LLMs by an average of 54% and 25% for ReDial and INSPIRED datasets. The most effective CARE strategy involves LLMs selecting and reranking candidate items that external recommenders provide based on contextual insights.

Keywords: Conversational Recommendation, Large Language Models

1 Introduction

A conversational recommender system (CRS) helps users achieve recommendation-related goals through multiple rounds of conversation [18,23,34]. Compared with traditional recommendation systems that mainly use entity-level information (*e.g., item name, ID or attributes*) as input, CRS adopts conversational data (*e.g., dialogue history*) as input, fully aligning to real-world scenarios [23,14]. Aside

⋆ Work was done during a remote internship at SMU.

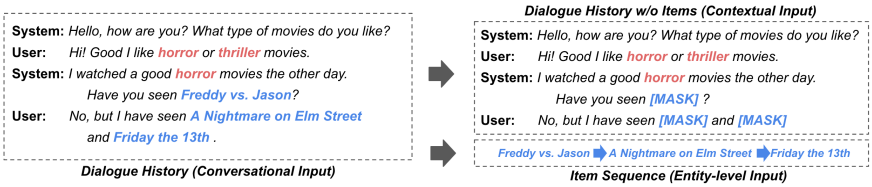

Fig. 1: CRS input example (ReDial Dataset) with conversational, contextual and entity-level inputs. (Blue: item entities; Red: attribute entities)

from the input difference, CRS aims to accomplish two tasks: 1) recommending items based on both the user's query or interests, and 2) generating high-quality conversational responses [23,26].

A key challenge for CRS is effectively leveraging dialogue history to generate accurate recommendations [14,21,24,53]. As illustrated in Figure 1, CRS utilizes two main types of information. First, entity-level information extracted from the dialogue history, including attribute entities (e.g., "user enjoys horror movies") or item entities (e.g., "historical movies watched by the user"). Second, contextual information expressed in natural language (e.g., "user's opinion for a specific movie"). These two types of information are integrated throughout the conversation to guide the CRS in generating relevant recommendations.

With the advent of large language models (LLMs), studies have explored their effectiveness as conversational recommenders [5,14,31]. Leveraging their superior language capabilities, LLMs show impressive results in zero-shot CRS applications. However, two key challenges remain due to the limitations of LLM-based CRS: ***1) item space discrepancy*** and ***2) item information negligence*** (§ 3.1). The issue of *item space discrepancy* arises when LLMs are pre-trained on data that differs from the domain-specific item space, causing LLMs to generate out-of-domain items, which fall outside the relevant target domain [14,47]. The problem of *item information negligence* stems from the zero-/few-shot learning setting, where LLMs lack access to item-specific training data [48,10]. Furthermore, they lack comprehensive user histories for items and attributes, leading them to provide recommendations in cold-start scenarios predominantly. Consequently, these models cannot leverage collaborative patterns among users and items. Instead, they focus on content-based queries from the dialogue history and rely heavily on contextual information in the dialogue history to make recommendations [14,38,15]. These two limitations have been acknowledged in prior studies (e.g., [22,17,14]), but remain unaddressed in subsequent CRS research.

Inspired by advances in traditional recommender systems that integrate LLMs with external collaborative filtering techniques [51,20], we propose the *Contextual Adaptation of RecommEnders (CARE)* framework. CARE bridges LLMs and external entity-level recommenders via contextual adaptation, using item entities as intermediaries. Specifically, we first train a lightweight domain-specific recommender based on entity sequences in dialogue history [57,58]. This recommender acts as a entity-level domain expert, providing insights in target domain (challenge 1). However, operating solely at the entity level limits its

ability to capture user intentions. For example, although both statements—*"I like horror movies for tonight"* and *"I like thrillers but I don't want to watch tonight!"*—express a taste for scary films, they reflect opposite intentions in viewing options. Therefore, we then adapt the recommender's insight but enable LLMs as rerankers to engage contextually by selecting and reranking the entity-based candidates using dialogue history (challenge 2). Specially, we study the rules and scales of the LLM-based rerankers as different strategies defined in § 4.2. Our main contributions are summarised:

- We target two identified but unsolved challenges in LLM-based CRS: *item space discrepancy* and *item information negligence*.
- We introduce the CARE framework, using LLM-based rerankers with external recommenders to leverage both entity-level and contextual information for conversational recommendation, demonstrating different strategies to enhance the quality and control the freedom of LLM-based rerankers.

2 Related Work

Conversational Recommendation System (CRS) aims to model user preferences and generate recommendations through multi-round of conversations [18,9,23]. The current research focus has shifted from the *attribute-based approaches*, where the system and users exchange items or attribute entities using a template [53,21], into *conversational approaches*, where the system interacts with users through natural language [25,6,39]. The majority of CRS in conversational approaches use language models (LMs; e.g., *DialoGPT*) for learning-based preference modelling [24,13,30]. Particularly, LMs are trained to generate recommendations or system responses using word or entity embeddings encoded from the conversations or external knowledge graphs [24,45,41,54].

Large Language Models for CRS. LLMs have shown promising performance in CRS as zero-/few-shot conversational recommenders using item-based [31,5] or conversational inputs [14,35,44] to generate recommendation results. However, the performance of LLMs largely depends on their internal knowledge and their capability will notably diminish in domains with scarce knowledge [22,44]. Thus, LLMs are integrated with external agents [8,29,17] to either directly provide the necessary knowledge resources or retrieve the essential knowledge resources from external knowledge base to improve their performance in domain-specific CRS tasks [22,47]. However, there is limited research in integrating traditional recommender systems to improve the item-specific capability for conversational recommendation tasks [15,58].

Sequential Modelling in CRS. Sequential modelling is well-suited for CRS applications for two reasons: (1) entities in CRS are mentioned in a sequential flow that leads to recommendations, and (2) cold-start scenarios in CRS often limit the collaborative filtering methods [42,58,24,47,58]. Compared to RNN models, transformers have shown superior performance in sequential recommendation [28,37]. Motivated by the work in sequential recommendation [37], the transformer-based sequential modelling framework is also applied to CRS by encompassing the

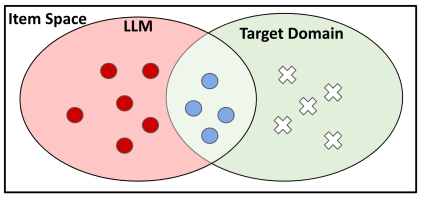

Fig. 2: Item space discrepancy between LLM and target domain. (Blue Dot: good recommendation results; Red Dot: recommendations outside target domain; White Cross: items out of LLM's internal knowledge)

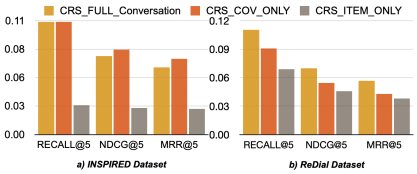

Fig. 3: Ablation study for LLM-based CRS with different levels (conversational or entity) inputs. (Yellow: original conversational inputs in datasets; Orange: Contextual inputs w/o items; Gray: Entity-level inputs w/o conversations)

entities mentioned in CRS conversations as a sequence using positional embedding [58] or knowledge-aware positional encoding [33]. However, the existing sequential modelling methods only use entity-level information while we introduce the framework to incorporate contextual information in modelling user preferences.

3 Preliminaries

Task Formulation. We consider the CRS scenario where a system *sys* interacts with a user u. Each dialogue contains conversation turns denoted as Cov. The sequence of entities mentioned by u or *sys* is extracted as a historical entity sequence $Seq = [e_1, e_2, ..., e_n]$. As this paper limits its scope to the *recommendation task* in CRS, the target function for CRS is expressed as the generation of a recommendation list of items $i_1, i_2, ..., i_n$, given the dialogue history, Cov and sequence of historical entities Seq.

$$i_1, i_2, ..., i_n = CRS(Cov,\ Seq) \tag{1}$$

3.1 Main Problems and Key Challenges

Item Space Discrepancy. We illustrate the issue of item space discrepancy in Figure 2. LLMs generate zero-shot recommendations based solely on their internal knowledge (left circle in red), which can significantly differ from the target item space (right circle in green) in CRS [14,22]. The overlap between these spaces represents where LLMs can make accurate recommendations (blue dots). The red dot indicates recommendations outside the target item space, while the white cross highlights target space items that the LLM cannot recommend due to lack of exposure or knowledge. We show the reduction of out-of-domain items from LLMs using our framework in Figure 8, which directly contributes to the performance improvement. While recommending out-of-domain items may occasionally produce creative outputs, like AI-generated content (AIGC for recommendation) [43,50], in real-world services such as e-commerce, these

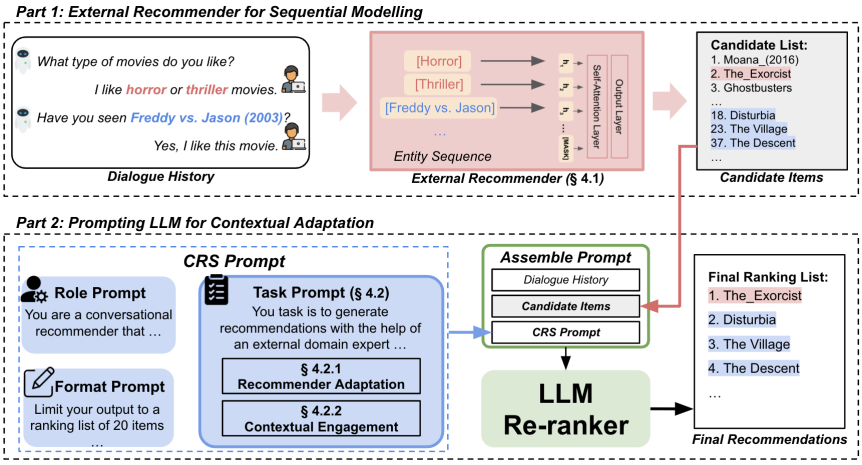

Fig. 4: **CARE Framework**: Conversational inputs are firstly passed to an external recommender for entity-level sequential modelling (§ 4.1). The candidate items output and original dialogue history are jointly used to prompt LLMs for the contextual adaptation and generate conversational recommendations (§ 4.2).

recommendations often misalign with the platform's offerings, which can lead to user disappointment [14,58,45]. Based on this need, we apply policy in different strategies to control the freedom of LLMs in reranking (§ 5.3).

Item-specific Information Negligence. LLMs excel at mapping user descriptions or queries to target items [14,38]. However, a key limitation is their lack of understanding of the item-specific information from the item entities mentioned by the users or system [14,22]. To investigate this, we follow the approach in [14] and evaluate LLM performance by separating different input types with or without item entities (Figure 1): (1) the full conversation ($CRS\_FULL\_Conversation$), which includes the complete dialogue history; (2) the entity-level item sequence (CRS_ITEM_ONLY), consisting solely of item entities (Seq); and (3) a context-only input (CRS_COV_ONLY), where item entities were excluded from the dialogue history ($Cov - Seq$). The results in Figure 3 demonstrate that LLMs primarily base their recommendations on contextual information, neglecting the item-specific information in the item entity sequence. While traditional recommenders are experts in entity-based information. To fill the gap, we propose CARE-CRS to integrate both contextual and item-specific information for conversational recommendation via LLM-based reranking.

4 CARE-CRS

The CARE-CRS framework (Figure 4) consists of two parts: 1) external recommenders processes entities in dialogue history for sequential modelling and 2) LLMs as rerankers for conversational recommendations.

4.1 Transformer-based Recommender for Sequential Modelling

The external recommender system takes entity sequences as input. To generate this sequence, we first apply entity extraction scripts to align the entities in the dialogue with the ones in knowledge graphs (e.g., movies, actors, attributes), as shown in Figure 1 [1,58,45]. The script includes 3 main steps: 1) the full dialogue is tokenized into short entity-level terms, 2) the similarity of each entity in dialogue and knowledge graphs are compared, and the final entities are selected based on the similarity score and 3) the selected entities are converted as a sequential list for each dialogue. For example, in dialogue "User: I want to watch a movie with Tom Cruise; System: How about watching Mission Impossible?", the corresponding entity sequences is ['Tom Cruise', 'Mission Impossible']. These step converts the dialogues into entity-level inputs for the sequential modelling.

While the external recommender can be implemented using any system with entity-level inputs and outputs, we adopt a transformer architecture for its superior performance in sequential modelling, as shown in prior works [33,58,28,19], Notably, we deliberately exclude LLMs from the recommender to preserve a clear modular separation from the LLM-based re-rankers. The model has three components: an embedding layer, a self-attention layer, and an output layer [7,37]. The embedding layer computes each entity's embedding combing graph embedding h_K and positional embedding h_P for final embedding h_i^0 as $h_K + h_P$, as in [33,58]. The self-attention layer uses a Multi-head Self-attention mechanism to learn the final sequential representation H, formulated as:

$$H = \text{Self-Attention}(h_1^0, h_2^0, ..., h_n^0, \textit{[MASK]}) \tag{2}$$

The output layer projects the final hidden states of sequential representation H to the item space Out_k using two layers of neural networks, where W_1, W_2, b_1, and b_2 are the weights and biases of the feed-forward networks. The final outputs are converted into text format for LLMs.

$$\text{Out}_k = [\text{out}_1 \ldots, \text{out}_k];\ \text{out}_i = \text{GELU}(HW_1^T + b_1)W_2^T + b_2 \tag{3}$$

4.2 Prompting LLMs for Contextual Adaptation

Without external recommenders, the existing approach uses instructional prompts to guide LLMs in generating recommendation outputs from dialogue history [14]. As shown in Figure 4 (Part 2), a prompt consists of three components: 1) **role prompt (P_R)** that defines the system's role, 2) **format prompt (P_F)** that constrains the output format for future post-processing, and 3) **task prompt (P_T)** that specifies the task and function of the system. The final prompt, $P_{CRS} = P_R + P_T + P_F$, integrates these three components. The overall process is formulated as follows:

$$i_1, i_2, ..., i_n = LLM(\ P_{CRS}, Cov) = LLM(\ (P_R + P_F + P_T), Cov) \tag{4}$$

In our setup, to better integrate external recommenders, we update the task prompt P_T from two key perspectives: 1) Recommender Adaptation and 2)

Contextual Engagement. These updates enable the LLMs to effectively leverage contextual information, facilitating their ability to generate, select, or rerank the entity-level recommendations derived from an external recommender system.

Adaptation of Recommender as Domain Expert. LLMs demonstrate significant potential when integrated with external sub-systems to form agent-based [49,46] or expert-guided systems [11]. Adaptation refers to the challenge of effectively introducing and describing these external systems so that LLMs can better understand and collaborate with them as domain experts [40] (Recommender Adaptation in Figure 4 Part 2). We design three methods to adapt the recommender systems as domain experts: 1) Direct Prompting, 2) Recommender Description, and 3) Self-Reflection (colored in orange). Each approach modifies the task prompt P_T in the LLM to facilitate this adaptation.

– **Direct Prompting:** Direct introduce the recommender as a domain expert without any description of its function or input/output format.

Task Prompt ($P_T^{m=1}$): *"To help you with the recommendation, we introduce a domain expert who provides recommendations based on the training data and you can use the domain expert's recommendations as examples for your output"*

– **Description of Recommender:** Briefly describe how the recommender works to model the sequential inputs and generate the candidate items.

Task Prompt ($P_T^{m=2}$): *"To help you with the recommendation, we introduce a domain-expert which is a recommender for sequential modelling that uses the entities mentioned in the dialogues to generate a ranking list of items."*

– **Self Reflection:** Step by step lead the LLM to examine all the resources of the recommender (e.g., code, paper or data samples). After each resource, we ask LLM to self-reflect, correcting or generating a new prompt to describe the recommender [56]. We repeat these steps until the prompts generated from the LLMs has minor or no difference with the previous one and LLMs confirm its confidence about its output. We then fix it as the final task prompt $P_T^{m=3}$ for self-reflection.

Task Prompt ($P_T^{m=3}$): *"To help you with the recommendation, you can access an advanced recommendation system that specializes in enhancing conversational recommendations by leveraging both the sequence of entities mentioned in a conversation and external knowledge embedded in knowledge graphs. This system ..."*

Contextual Engagement for Conversational Recommendations. After introducing the recommender as a domain expert, we propose contextual engagement strategies that enable LLMs to learn item space information from the expert and incorporate contextual data to refine its recommendations [14,16]. This allows

Dataset	#Dialogues	#Turns	#Users	#Items	Avg. #Entities/Dialogue
ReDial	11,348	139,557	764	6,281	7.24
INSPIRED	999	35,686	999	1,967	12.88

Table 1: Statistics of datasets.

LLMs to generate target domain-specific recommendations and correct external recommender errors, such as *1) context-agnostic rankings, where the recommender ignores the explicit preference over an item in the context* or *2) popularity bias, where the statistically popular item is over-ranked*, both are common issues in learning-based approaches [58,27]. In addition, as practiced in major industrial reranking systems, we design policy to control the freedom of LLM generation (colored in blue) [2,4,32]. We introduce three contextual engagement strategies: 1) reranking and 2) selection-then-reranking, modifying prompt P_T and formulated in the format of [Adaptation + Engagement + Policy].

– **Reranking:** Provide the same number of recommendations as the desired output and ask the model to rerank the candidate items using dialogue history.

Task Prompt ($P_T^{s=1}$): *To help ... You need to **rerank** the recommendations, placing the domain expert's suggestions in the appropriate order based on your understanding of the dialogue history. You **cannot** generate items beyond ...*

– **Selection-then-Reranking:** Show a larger set of recommendations and ask the model to select from them, rerank the selected items, to form a ranked list.

Task Prompt ($P_T^{s=2}$): *To help ... You need to **select** the most appropriate items from the domain expert's recommendations and **rerank** them in a ranked order based on dialogue history. You **cannot** generate items beyond ...*

Given the dialogue history Cov and the top-K candidate items from the external recommender Out_k as inputs, LLM is prompted to generate n outcomes with adaptation methods m, contextual engagement strategies s, using the final prompt $P_{CARE} = (P_R + (P_T^m + P_T^s) + P_F)$ as formulated in:

$$\begin{aligned} i_1, i_2, ..., i_n &= LLM(\ P_{CARE}, Cov, Out_k) \\ &= LLM(\ (P_R + (P_T^m + P_T^s) + P_F), Cov, Out_k) \end{aligned} \tag{5}$$

5 Experiments

We address the following research questions (RQs) in this section:

– **RQ1**: How is the performance of CARE-CRS in recommendation tasks?
– **RQ2**: How are contextual adaptation strategies optimised in CARE-CRS?
– **RQ3**: How efficient is CARE-CRS in solving identified challenges and biases?

5.1 Setup

Datasets & Evaluation Metrics. The experiment is conducted on two public CRS datasets: ReDial [24] and INSPIRED [13]. The example of the dataset is shown in Figure 1 with statistics shown in Table 1 and we follow the original data split (8:1:1) [55]. We evaluate our model and baselines mostly on their recommendation abilities using metrics HIT@K, MRR@K and NDCG@K with K in $\{5, 10\}$, and response evaluation is not considered in this work [6,14,24,41,45].
Baselines. To compare our model with the existing CRS works, we adopt the baselines including 3 types of models. a) Benchmarks(BMK) released with datasets: **ReDial** [24] use auto-encoder and **INSPIRED** [13] fine-tunes transformer-based LMs for recommendation; b) Learning-based approaches: **KBRD** [3] adopts external knowledge graph from DBpedia [1], **KGSF** [54] use semantic fusion to align the representation of entities and dialogue history, **SASRec** [19] use self-attention structure and **UniCRS** [6] use prompt learning as a unified approach to jointly improve recommendation and response genenration; c) LLM-based methods: **ZSCRS** [14] studies the zero-shot conversational recommendation capability of LLM (ChatGPT), **Llama 3** [12] implements ZSCRS using open-sourced LLM (Llama3-8B-Instruct) and **MemoCRS** [47] retrieve memory-based and collaborative information from the user's profile for preference modelling and generation (GPT4).
Implementation Details. We adopt the data pre-processed from the open-source CRS toolkit CRSLab [55] and use closed-sourced ChatGPT and open-sourced Llama 3 [12][4].The temperature of ChatGPT is set as 0 to ensure the same output with fixed input tokens. For the sequential modelling recommender, we follow the existing settings [33,58] using 2 layers and 2 attention heads with an SGD optimizer and learning rate of 5e-3, dropout rates of 0.2. For our baselines, we replicate UniCRS [45] using their source code and other models using open-sourced CRSLab [55]. For MemoCRS [47], as their memory resources are not published, we report the results from the paper.

5.2 Experimental Results (RQ1)

Recommendation Performance. Table 2 presents the recommendation task results. We report the result of CARE-CRS using GPT3.5 as the base model and the optimized prompt with recommender description and selection-then-reranking as the best strategy (§ 5.3). All LLM-based approaches significantly outperform learning-based and benchmark baselines, which highlights the strong ability of LLMs to handle conversational queries for CRS applications [14,47]. Our proposed CARE-CRS framework, which integrates contextual recommender adaptation, surpasses all state-of-the-art LLM-based methods with statistically significant improvements for top-5 and -10 accuracy across all metrics on both datasets. The performance gain for the top-5 metrics is generally higher than that for the top-10, with an average increase of 65% and 31% over the best LLM baseline. In Figure 5, we show the recommendation results of CARE-CRS using

[4] ChatGPT: gpt-3.5-turbo-0125; Llama 3: meta-llama/Meta-Llama-3-8B-Instruct

	Models	ReDial			INSPIRED		
		H@5/10	M@5/10	N@5/10	H@5/10	M@5/10	N@5/10
BMK	**ReDial**	.029/.041	.017/.019	.020/.024	.003/.003	.001/.001	.002/.002
	INSPIRED	.099/.168	.050/.059	.062/.084	.058/.097	.041/.046	.046/.058
Learning	**KBRD**	.082/.140	.034/.042	.046/.065	.019/.032	.007/.009	.010/.014
	KGSF	.091/.138	.039/.045	.051/.066	.016/.029	.006/.007	.008/.012
	SASRec	.036/.056	.020/.023	.024/.030	.088/.149	.054/.062	.062/.082
	UniCRS	.101/.161	.039/.047	.054/.073	.091/.106	.062/.064	.070/.074
LLM	**ZSCRS**	.111/.165	.057/.064	.070/.087	.109/.142	.065/.069	.076/.086
	MemoCRS	.136/.215	.072/.082	.088/.113	NA	NA	NA
	CARECRS	.194*/.248*	.133*/.140*	.148*/.166*	.144*/.169*	.090*/.093*	.103*/.111*
	Improvement	42.7/15.4%	84.7/70.7%	68.2/46.9%	22.0/11.9%	38.5/29.2%	32.1/16.8%

Table 2: Results of different models on recommendation task in H/M/N (HIT/MRR/NDCG). Best results from our model and baselines are coloured; Symbol * indicates statistical significance with $p < 0.05$.

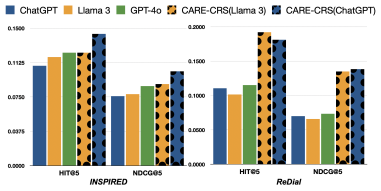

Fig. 5: Comparison between open-sourced LLMs and ChatGPT in recommendation accuracy.

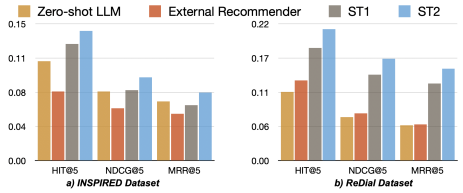

Fig. 6: Analysis of different CARE strategies in CARE-CRS and ST1/ST2 stands for Reranking/Selection-then-Reranking.

open-sourced Llama 3 with 8B parameters. We adopt the GPT-4o [5] models as a strong baseline to compare the results before and after applying CARE framework. For both datasets, the original recommendation performance of LLMs is lower than GPT-4o. However, after applying the CARE framework, they all outperform the GPT-4o in the recommendation performance, showing the efficiency and robustness of our framework for both the open- and closed-sourced LLMs.

Compared to those leading LLM-based approaches, our method significantly enhances zero-shot performance by integrating a lightweight entity-level recommender for contextual adaptation, which benefit from a) sufficient candidate items, reducing the ratio of out-of-domain recommendations (discussed in § 5.4) and b) entity-level sequential modelling from the recommender, incorporating both item-based and contextual information for final recommendations.

Training Efficiency and System Complexity. In contrast to learning-based baselines like UniCRS, which fully fine-tune a language model such as DialoGPT [52] with 762M parameters, our approach leverages a zero-shot LLM inference through API without fine-tuning. The only training involved is for the external

[5] gpt-4o-08-06

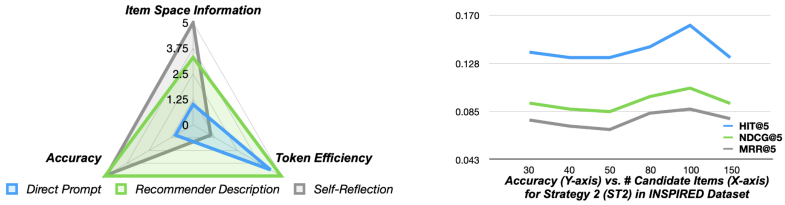

Fig. 7: Ablation study on adaptation methods(left) and candidate numbers(right).

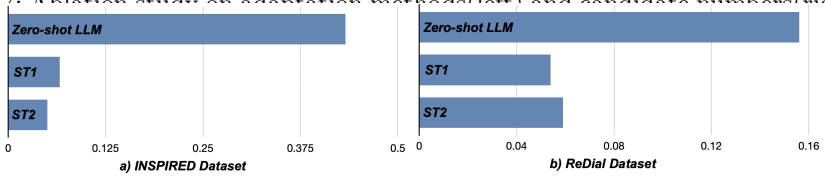

Fig. 8: Ratio of recommendation items out of target domain for different strategies.

recommender, which has 2 layers and 2 attention heads using a transformer structure, totalling 2.7 M parameters—less than 1% of the size of the baseline models. As our overall framework is modular, both the external recommender and the LLMs can be replaced with newer models. Future work can further enhance training efficiency by using alternative pre-trained recommendation systems in the target domain, eliminating the need for additional training.

5.3 Discussion on CARE-CRS Methods and Strategies (RQ2)

We propose several methods for contextual adaptation between LLMs and recommenders in § 4. This section presents three ablation studies to evaluate and optimize the approaches for recommender adaptation, contextual engagement, and managing the number of candidate items.

Contextual Engagement. Figure 6 illustrates the recommendation accuracy of the zero-shot LLM, external recommender, and the CARE framework using contextual engagement strategies (*ST1*, *ST2* corresponding to *reranking*, and *selection-then-reranking*). The performance of the zero-shot LLM and external recommender varies by dataset: the LLM excels in the INSPIRED dataset, while the external recommender performs better in ReDial. This discrepancy arises from the entity composition of each dataset (Table 1). Because INSPIRED dataset contains more attribute entities, which provides rich context information for LLMs. This conclusion is further substantiated by Table 2, which shows that CARE-CRS, primarily introducing item-based information, achieves a significantly greater performance gain on ReDial than on INSPIRED. Despite dataset differences, *ST1* and *ST2* consistently outperform both the zero-shot LLM and external recommender by leveraging reranking or selection based on context. The consistency among the two strategies shows that the CARE framework is robust across different types of datasets, whether item- or attribute-rich.

Adaptation Methods. In § 4, we propose 3 different adaptation methods to introduce the role of the external recommender to LLM-based CRS and Figure 7

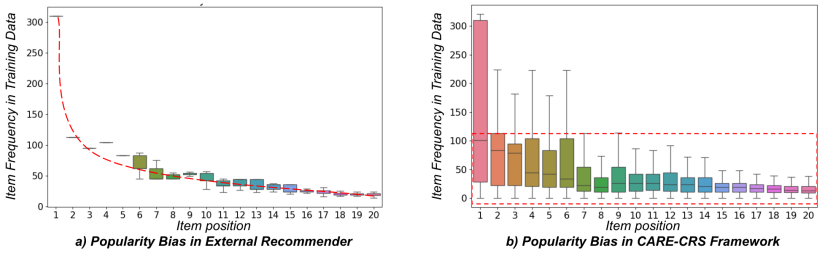

Fig. 9: Popularity bias of external recommender (left) and CARE-CRS (right).

(left) presents a comparison of various adaptation methods across three key dimensions: *a) item space information*, *b) recommendation accuracy*, and *c) token efficiency*. Item space information assesses how effectively each method can prevent the LLMs from generating out-of-domain items, while token efficiency reflects the token count per prompt. Recommendation accuracy compares the influence of task prompts on the final recommendation results. All metrics are normalized on a scale from 0 to 5, where higher scores indicate better outcomes. Our results demonstrate that both *description of recommender* and *self-reflection*—which incorporate detailed descriptions of external systems and their functionalities—significantly enhance recommendation accuracy and item space information compared to *direct prompting*. However, the *self-reflection* method incurs a higher token count, resulting in lower token efficiency compared to *recommender description*. In summary, as illustrated in the radar chart (Figure 7), *description of recommender* delivers the best overall performance in the CARE framework.

Candidate Numbers. In Figure 7, we analyze how candidate number k impacts recommendation accuracy in *ST2*, omitting *ST1* due to its fixed parameter in k. For *ST2* (default $k > 20$), although the candidate number moderately affects performance (optimized $k = 100$), it is relatively stable compared to the strategy selection. However, increasing the number of candidates also leads to longer input tokens, necessitating flexible parameter to optimize performance and efficiency.

5.4 Discussion of CARE-CRS on CRS Challenges and Bias (RQ3)

We further analyze the performance of the CARE framework in tackling key challenges and biases, such as item space discrepancy and popularity bias.

Item Space Discrepancy from LLMs. In Figure 8, we report the ratio of recommended items falling outside the target domain for each method. Because LLM-based recommendations are generated in free text, we map each recommended item to a target-domain item via Levenshtein distance, which calculates the semantic similarity between two terms; any item whose distance exceeds a threshold $\vartheta = 2$ is classified as out-of-domain. Zero-shot LLMs often generate items that do not exist in the target domain (*item space discrepancy*, §3.1), whereas external recommenders, which are trained to rank items solely within the target domain, exhibit no such discrepancy. Our proposed strategies mitigate

this issue by applying policy to a given candidate set. As shown in Figure 8, CARE reduces out-of-domain outputs and consequently improve recommendation accuracy (Figure 6).

Popularity Bias from Recommender. As discussed in § 4.2, popularity bias remains a significant challenge in learning-based recommendation systems [27,36]. Entity-level recommenders often reinforce this bias by consistently ranking the same popular items at the top. Figure 9 presents a box plot of item popularity to their ranking position, showing that external recommenders (with high mean and low variance) consistently rank the same popular items at the top positions. In contrast, with contextual engagement in the CARE framework, the model explicitly understands the user's diverse intentions and preferences (with low mean and high variance). As a result, they can either select lower-ranked items and rerank them to higher positions to form the final ranking list. As a real example shown in Figure 4, even though the user prefers horror movies, the external recommender still ranks *Monna* (a fantasy movie) first due to its popularity in training data, which is irrelevant to the context. However, with our method, the LLM can remove this popular item by excluding *Monna* from the final ranking list. Consequently, it reranks other relevant movies such as *The Exorcist* (horror) and *Disturbia* (thriller) by elevating them to top positions, which aligns better with the user's stated interests and contextual information.

6 Conclusion

This paper presents a new framework that integrates LLM-based reranking with external recommenders, aiming to address two key limitations of LLM-based CRS. We employ a transformer-based recommender to perform entity-level sequential modelling and generate a candidate set of items. Subsequently, LLMs are guided to select and rerank these candidates based on conversational context through a contextual adaptation process. This design enables seamless and training-efficient collaboration between LLMs and external recommenders. Results and analysis demonstrate the efficacy over both open- and closed-source LLMs, while mitigating the identified limitations and bias.

Acknowledgments. This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant (Proposal ID: 24-SIS-SMU-002) and the Lee Kong Chian Fellowship awarded to DENG Yang by Singapore Management University.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia - a crystallization point for the web of data. Journal of Web

Semantics **7**(3), 154–165 (2009). https://doi.org/https://doi.org/10.1016/j.websem.2009.07.002, https://www.sciencedirect.com/science/article/pii/S1570826809000225, the Web of Data
2. Carraro, D., Bridge, D.: Enhancing recommendation diversity by re-ranking with large language models. ACM Transactions on Recommender Systems (2024)
3. Chen, Q., Lin, J., Zhang, Y., Ding, M., Cen, Y., Yang, H., Tang, J.: 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). pp. 1803–1813. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1189, https://aclanthology.org/D19-1189
4. Chen, S., Wang, Y., Wen, Z., Li, Z., Zhang, C., Zhang, X., Lin, Q., Zhu, C., Xu, J.: Controllable multi-objective re-ranking with policy hypernetworks. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 3855–3864. KDD '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3580305.3599796, https://doi.org/10.1145/3580305.3599796
5. Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., Sun, Z., Zhang, X., Xu, J.: Uncovering chatgpt's capabilities in recommender systems. In: Proceedings of the 17th ACM Conference on Recommender Systems. p. 1126–1132. RecSys '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3604915.3610646, https://doi.org/10.1145/3604915.3610646
6. Deng, Y., Zhang, W., Xu, W., Lei, W., Chua, T.S., Lam, W.: A unified multi-task learning framework for multi-goal conversational recommender systems. ACM Trans. Inf. Syst. **41**(3) (feb 2023). https://doi.org/10.1145/3570640
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
8. Feng, Y., Liu, S., Xue, Z., Cai, Q., Hu, L., Jiang, P., Gai, K., Sun, F.: A large language model enhanced conversational recommender system (2023)
9. Gao, C., Lei, W., He, X., de Rijke, M., Chua, T.S.: Advances and challenges in conversational recommender systems: A survey. AI Open **2**, 100–126 (2021). https://doi.org/https://doi.org/10.1016/j.aiopen.2021.06.002, https://www.sciencedirect.com/science/article/pii/S2666651021000164
10. Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., Zhang, J.: Chat-rec: Towards interactive and explainable llms-augmented recommender system. arXiv preprint arXiv:2303.14524 (2023)
11. Ge, Y., Hua, W., Mei, K., Tan, J., Xu, S., Li, Z., Zhang, Y., et al.: Openagi: When llm meets domain experts. Advances in Neural Information Processing Systems **36** (2024)
12. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., et al.: The llama 3 herd of models (2024), https://arxiv.org/abs/2407.21783
13. Hayati, S.A., Kang, D., Zhu, Q., Shi, W., Yu, Z.: Inspired: Toward sociable recommendation dialog systems. In: Proceedings of the 2020 Conference on Empirical

Methods in Natural Language Processing (EMNLP). pp. 8142–8152. Association for Computational Linguistics, Online (Nov 2020), https://www.aclweb.org/anthology/2020.emnlp-main.654
14. He, Z., Xie, Z., Jha, R., Steck, H., Liang, D., Feng, Y., Majumder, B.P., Kallus, N., McAuley, J.: Large language models as zero-shot conversational recommenders. arXiv preprint arXiv:2308.10053 (2023)
15. He, Z., Xie, Z., Steck, H., Liang, D., Jha, R., Kallus, N., McAuley, J.: Reindex-then-adapt: Improving large language models for conversational recommendation. arXiv preprint arXiv:2405.12119 (2024)
16. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems (2024), https://arxiv.org/abs/2305.08845
17. Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D., Xie, X.: Recommender ai agent: Integrating large language models for interactive recommendations. arXiv preprint arXiv:2308.16505 (2023)
18. Jannach, D., Manzoor, A., Cai, W., Chen, L.: A survey on conversational recommender systems. ACM Comput. Surv. **54**(5) (may 2021). https://doi.org/10.1145/3453154
19. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation (2018), http://arxiv.org/abs/1808.09781
20. Kim, S., Kang, H., Choi, S., Kim, D., Yang, M., Park, C.: Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 1395–1406. KDD '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3637528.3671931, https://doi-org.libproxy1.nus.edu.sg/10.1145/3637528.3671931
21. Lei, W., He, X., Miao, Y., Wu, Q., Hong, R., Kan, M.Y., Chua, T.S.: Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In: Proceedings of the 13th International Conference on Web Search and Data Mining. p. 304–312. WSDM '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3336191.3371769
22. Li, C., Deng, Y., Hu, H., Kan, M.Y., Li, H.: Incorporating external knowledge and goal guidance for llm-based conversational recommender systems (2024), https://arxiv.org/abs/2405.01868
23. Li, C., Hu, H., Zhang, Y., Kan, M.Y., Li, H.: A conversation is worth a thousand recommendations: A survey of holistic conversational recommender systems. In: KaRS Workshop at ACM RecSys '23. Singapore (2023), https://arxiv.org/abs/2309.07682v1
24. Li, R., Kahou, S., Schulz, H., Michalski, V., Charlin, L., Pal, C.: Towards deep conversational recommendations. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 9748–9758. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
25. Li, R., Kahou, S.E., Schulz, H., Michalski, V., Charlin, L., Pal, C.: Towards deep conversational recommendations. In: Advances in Neural Information Processing Systems 31 (NIPS 2018) (2018)
26. Li, S., Xie, R., Zhu, Y., Ao, X., Zhuang, F., He, Q.: 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. pp. 223–233 (2022). https://doi.org/10.1145/3477495.3532074, http://arxiv.org/abs/2204.09263

27. Lin, A., Wang, J., Zhu, Z., Caverlee, J.: Quantifying and mitigating popularity bias in conversational recommender systems. In: Proceedings of the 31st ACM international conference on information & knowledge management. pp. 1238–1247 (2022)
28. Liu, Q., Wu, S., Wang, D., Li, Z., Wang, L.: Context-aware sequential recommendation. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). pp. 1053–1058. IEEE (2016)
29. Liu, Y., Zhang, W., Chen, Y., Zhang, Y., Bai, H., Feng, F., Cui, H., Li, Y., Che, W.: Conversational recommender system and large language model are made for each other in E-commerce pre-sales dialogue. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 9587–9605. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.findings-emnlp.643, https://aclanthology.org/2023.findings-emnlp.643
30. Liu, Z., Wang, H., Niu, Z.Y., Wu, H., Che, W.: DuRecDial 2.0: A bilingual parallel corpus for conversational recommendation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 4335–4347. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.356, https://aclanthology.org/2021.emnlp-main.356
31. Palma, D.D., Biancofiore, G.M., Anelli, V.W., Narducci, F., Noia, T.D., Sciascio, E.D.: Evaluating chatgpt as a recommender system: A rigorous approach (2023)
32. Pei, C., Zhang, Y., Zhang, Y., Sun, F., Lin, X., Sun, H., Wu, J., Jiang, P., Ge, J., Ou, W., et al.: Personalized re-ranking for recommendation. In: Proceedings of the 13th ACM conference on recommender systems. pp. 3–11 (2019)
33. Petruzzelli, A., Martina, A.F.M., Spillo, G., Musto, C., De Gemmis, M., Lops, P., Semeraro, G.: Improving transformer-based sequential conversational recommendations through knowledge graph embeddings. In: Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization. p. 172–182. UMAP '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3627043.3659565, https://doi-org.libproxy1.nus.edu.sg/10.1145/3627043.3659565
34. Pramod, D., Bafna, P.: Conversational recommender systems techniques, tools, acceptance, and adoption: A state of the art review. Expert Systems with Applications **203**, 117539 (2022)
35. Sanner, S., Balog, K., Radlinski, F., Wedin, B., Dixon, L.: Large language models are competitive near cold-start recommenders for language- and item-based preferences. In: Proceedings of the 17th ACM Conference on Recommender Systems. p. 890–896. RecSys '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3604915.3608845, https://doi.org/10.1145/3604915.3608845
36. Shen, T., Li, J., Bouadjenek, M.R., Mai, Z., Sanner, S.: Towards understanding and mitigating unintended biases in language model-driven conversational recommendation. Information Processing & Management **60**(1), 103139 (2023)
37. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. CoRR **abs/1904.06690** (2019), http://arxiv.org/abs/1904.06690
38. Tay, Y., Tran, V.Q., Dehghani, M., Ni, J., Bahri, D., Mehta, H., Qin, Z., Hui, K., Zhao, Z., Gupta, J., Schuster, T., Cohen, W.W., Metzler, D.: Transformer memory as a differentiable search index (2022), https://arxiv.org/abs/2202.06991

39. Wang, J., Lin, D., Li, W.: A target-driven planning approach for goal-directed dialog systems. IEEE Transactions on Neural Networks and Learning Systems (2023)
40. Wang, K., Lu, Y., Santacroce, M., Gong, Y., Zhang, C., Shen, Y.: Adapting llm agents through communication. arXiv preprint arXiv:2310.01444 (2023)
41. Wang, L., Hu, H., Sha, L., Xu, C., Wong, K., Jiang, D.: Finetuning large-scale pre-trained language models for conversational recommendation with knowledge graph. CoRR **abs/2110.07477** (2021), https://arxiv.org/abs/2110.07477
42. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q.Z., Orgun, M.: Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830 (2019)
43. Wang, W., Lin, X., Feng, F., He, X., Chua, T.S.: Generative recommendation: Towards next-generation recommender paradigm. arXiv preprint arXiv:2304.03516 (2023)
44. Wang, X., Tang, X., Zhao, X., Wang, J., Wen, J.R.: Rethinking the evaluation for conversational recommendation in the era of large language models. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 10052–10065. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-main.621, https://aclanthology.org/2023.emnlp-main.621
45. Wang, X., Zhou, K., Wen, J.R., Zhao, W.X.: Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 1929–1937. KDD '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3534678.3539382
46. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X., Wang, C.: Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155 (2023)
47. Xi, Y., Liu, W., Lin, J., Chen, B., Tang, R., Zhang, W., Yu, Y.: Memocrs: Memory-enhanced sequential conversational recommender systems with large language models (2024), https://arxiv.org/abs/2407.04960
48. Yang, T., Chen, L.: Unleashing the retrieval potential of large language models in conversational recommender systems. In: Proceedings of the 18th ACM Conference on Recommender Systems. pp. 43–52 (2024)
49. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models (2023)
50. Yoon, S.e., He, Z., Echterhoff, J.M., McAuley, J.: Evaluating large language models as generative user simulators for conversational recommendation. arXiv preprint arXiv:2403.09738 (2024)
51. Zhang, J., Hou, Y., Xie, R., Sun, W., McAuley, J., Zhao, W.X., Lin, L., Wen, J.R.: Agentcf: Collaborative learning with autonomous language agents for recommender systems. In: Proceedings of the ACM Web Conference 2024. p. 3679–3689. WWW '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3589334.3645537, https://doi-org.libproxy1.nus.edu.sg/10.1145/3589334.3645537
52. Zhang, Y., Sun, S., Galley, M., Chen, Y.C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, B.: Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv preprint arXiv:1911.00536 (2019)
53. Zhang, Y., Chen, X., Ai, Q., Yang, L., Croft, W.B.: Towards conversational search and recommendation: System ask, user respond. In: Proceedings of the 27th acm

international conference on information and knowledge management. pp. 177–186 (2018)
54. Zhou, K., Zhao, W.X., Bian, S., Zhou, Y., Wen, J.R., Yu, J.: Improving conversational recommender systems via knowledge graph based semantic fusion. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1006–1014 (2020)
55. Zhou, K., Zhou, X.W.Y., Shang, C., Cheng, Y., Zhao, W.X., Li, Y., Wen, J.R.: Crslab: An open-source toolkit for building conversational recommender system. arXiv preprint arXiv:2101.00939 (2021)
56. Zhou, P., Pujara, J., Ren, X., Chen, X., Cheng, H.T., Le, Q.V., Chi, E.H., Zhou, D., Mishra, S., Zheng, H.S.: Self-discover: Large language models self-compose reasoning structures (2024), https://arxiv.org/abs/2402.03620
57. Zou, J., Kanoulas, E., Ren, P., Ren, Z., Sun, A., Long, C.: Improving conversational recommender systems via transformer-based sequential modelling. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2319–2324. ACM (2022). https://doi.org/10.1145/3477495.3531852, https://dl.acm.org/doi/10.1145/3477495.3531852
58. Zou, J., Kanoulas, E., Ren, P., Ren, Z., Sun, A., Long, C.: Improving conversational recommender systems via transformer-based sequential modelling. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 2319–2324. SIGIR '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3477495.3531852