Beyond Single Labels: Improving Conversational Recommendation through LLM-Powered Data Augmentation

Haozhe Xu, Xiaohua Wang, Changze Lv, Xiaoqing Zheng*
School of Computer Science, Fudan University, Shanghai, China
{hzxu22,xiaohuawang22,czlv22}@m.fudan.edu.cn
zhengxq@fudan.edu.cn

Abstract

Conversational recommender systems (CRSs) enhance recommendation quality by engaging users in multi-turn dialogues, capturing nuanced preferences through natural language interactions. However, these systems often face the false negative issue, where items that a user might like are incorrectly labeled as negative during training, leading to suboptimal recommendations. Expanding the label set through data augmentation presents an intuitive solution but faces the challenge of balancing two key aspects: ensuring semantic relevance and preserving the collaborative information inherent in CRS datasets. To address these issues, we propose a novel data augmentation framework that first leverages an LLM-based semantic retriever to identify diverse and semantically relevant items, which are then filtered by a relevance scorer to remove noisy candidates. Building on this, we introduce a two-stage training strategy balancing semantic relevance and collaborative information. Extensive experiments on two benchmark datasets and user simulators demonstrate significant and consistent performance improvements across various recommenders, highlighting the effectiveness of our approach in advancing CRS performance. [1]

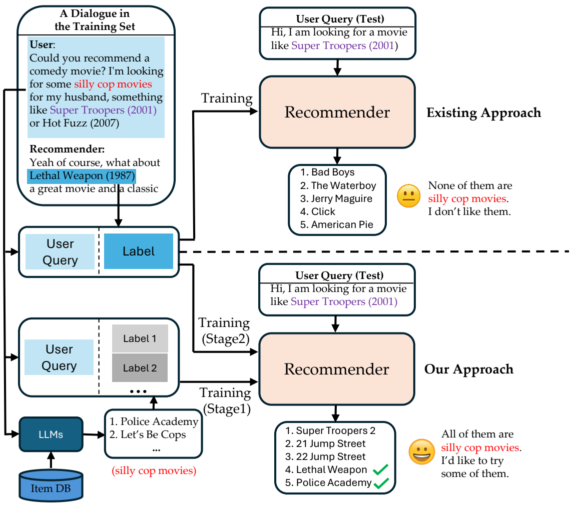

Figure 1: Comparison of existing and our proposed training processes. In the existing process, only one "silly cop movie" is labeled as positive, leading to unsatisfactory recommendations. Our approach considers multiple "silly cop movies" in the first training stage and uses the original label in the second training stage, which helps the recommender recommend "silly cop movies" while preserving collaborative information, such as the co-occurrence pattern of "Super Troopers" and "Lethal Weapon". The user finally accepted "Lethal Weapon" and "Police Academy".

1 Introduction

Conversational recommender systems (CRSs) provide recommendations by engaging users in multi-turn dialogues, capturing preferences through natural language interactions. This allows for explicit feedback and a nuanced understanding of user preferences (Gao et al., 2021; Jannach et al., 2021). As a result, CRSs are considered the next-generation recommender systems (Lin and Zhang, 2023; Chen et al., 2024).

A CRS typically consists of a recommender responsible for producing recommendations and a dialogue generator tasked with generating the conversation. While recent advances in large language models (LLMs) have alleviated the challenges associated with generating the conversation (Qin et al., 2024), significant challenges remain in optimizing the recommender. Figure 1 illustrates a typical example of training a recommender, as commonly used in previous works. In this example, when a user requests "silly cop movies", the recommender provides only one movie as a recommendation. This user query and the recommended movie are treated as an input-label pair to train the recommender. However, this approach does not account for other movies that the user might also be interested in, particularly other "silly cop movies".

*Corresponding author
[1]Code is available in https://github.com/xu1110/FNSCRS

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15573–15590
July 27 - August 1, 2025

Consequently, the recommender fails to suggest suitable movies, reducing the overall user experience. This issue is known as the "false negative issue" in traditional recommender systems (Ding et al., 2020; Liu et al., 2023; Wei et al., 2024), where items that a user might like are incorrectly labeled as negative during training. However, to the best of our knowledge, this issue has received limited attention in the context of CRS.

Expanding the label set is an intuitive solution to mitigate the false negative issue (Wei et al., 2024; Liu et al., 2023). This approach incorporates additional items into the training process, enriching the label set. However, CRS datasets pose two unique challenges for data augmentation: ensuring semantic relevance and preserving collaborative information.

Firstly, CRS datasets are rich in semantic information (He et al., 2023). Augmented labels should be diverse and semantically relevant to the dialogue context. For example, if a user requests a "silly cop movie", movies featuring a silly cop theme should be considered semantically relevant. LLMs excel at understanding and generating semantically relevant content, making them well-suited for this task. By analyzing the dialogue context and item descriptions, LLMs can identify multiple semantically relevant items, effectively broadening the label set. However, relying solely on semantic relevance risks overlooking collaborative information, such as commonalities and trends among different users, which play an important role in effective recommendations (Zhu et al., 2024; Zhang et al., 2023; He et al., 2024).

Secondly, collaborative information inherent in the original dataset should be preserved. While LLMs are adept at capturing semantic nuances, they often struggle to capture collaborative information effectively (He et al., 2023). Over-reliance on LLM-suggested labels can lead to recommendations that prioritize semantic relevance at the expense of collaborative information, ultimately reducing user satisfaction. Moreover, while collaborative information is important, it can sometimes introduce biases, such as popularity bias (Zhu et al., 2021; Zhao et al., 2022). Effectively utilizing collaborative information remains a challenging task.

To this end, we introduce a data augmentation approach to tackle the above challenges. Firstly, we use an LLM-based semantic retriever to retrieve multiple and diverse potentially relevant items based on semantic similarity between the dialogue context and the item's descriptive text. This step does not primarily consider collaborative information, thereby avoiding potential biases associated with it. Then, we employ an LLM-based relevance scorer to assign fine-grained relevance scores to the retrieved items, filtering out those irrelevant ones. Items with high relevance scores are retained as augmented labels, forming the augmented dataset. After that, we pre-train the recommender using the augmented dataset. This step enhances the recommender's ability to provide recommendations based on the semantic relevance between user preferences and items. Then, we fine-tune the recommender on the original dataset to integrate the collaborative information. A label smoothing term can be added to control the integration of collaborative information.

We have conducted extensive experiments on two real-world CRS benchmarks, namely ReDial (Li et al., 2018) and INSPIRED (Hayati et al., 2020), as well as a recently proposed user simulator iEvaLM (Wang et al., 2023b). Experimental results show that our approach can significantly and consistently improve the recommendation performance of different recommenders on different datasets. These results demonstrated the effectiveness of our approach.

2 Related Work

2.1 Conversational Recommender Systems

Existing CRS can be categorized into two main types: attribute-based CRS and free-form CRS. An attribute-based CRS explores user preferences by posing clarification questions about item attributes and then generating utterances using predefined templates (Sun and Zhang, 2018; Zhang et al., 2018; Lei et al., 2020). A free-form CRS explores user preferences through various conversational modes, including chit-chat, question-answering, and recommendation attempts (Chen et al., 2019; Zhou et al., 2020; Yang et al., 2022; Zhou et al., 2022; Wang et al., 2022b; Zou et al., 2022; Ren et al., 2022; Wang et al., 2023b; He et al., 2023). In this paper, we focus on the free-form CRS due to its flexibility in engaging users in more natural and diverse interactions.

Developing an effective recommender is fundamental in constructing a CRS. Previous approaches include knowledge-graph-based (Chen et al., 2019; Zhou et al., 2020; Wang et al., 2022b), review-based (Lu et al., 2021), and text-based models

(Yang et al., 2022; Ravaut et al., 2024). Typically, these models use a limited number of positive items as training labels and treat others as negative, which can introduce noise. We propose a method to generate multiple labels for each dialogue.

2.2 Synthetic Data in Conversational Recommendation

To address the challenge of data scarcity, researchers have explored the use of synthetic data to train CRS. These efforts have primarily focused on generating **synthetic dialogues** through template-based dialogue creation (Zhao et al., 2023), counterfactual dialogue simulation (Wang et al., 2023c), data-to-text dialogue generation (Lu et al., 2023), and LLM-based dialogue generation (Liang et al., 2024; Wang et al., 2023a; Kim et al., 2024). In this paper, we investigate the generation of **synthetic labels** from existing dialogues—an underexplored yet complementary approach to existing works.

2.3 Large Language Model-enhanced Conversational Recommendation

With the development of large language models (LLMs), there are some early attempts to apply large language models for conversational recommendation. These methods focus on dialogue management (Friedman et al., 2023; Fang et al., 2024; Feng et al., 2023), evaluation (Wang et al., 2023b; Huang et al., 2024), dialogue generation (Qin et al., 2024) or recommendation (He et al., 2023, 2024; Yang and Chen, 2024; Xi et al., 2024). Although showing promising results, most of these methods suffer from high inference latency. In this paper, we use LLMs to create synthetic data to support the training of downstream recommenders, which can avoid extensive inference latency.

2.4 False Negative Issues in Traditional Recommender Systems

The false negative issue occurs when a user is not exposed to certain items that they may be interested in, causing the recommender system to incorrectly assume that the user is not interested in them. These items are false negative samples (FNS).‘

The current methods in traditional recommender systems can mainly be divided into two categories. The first category treats FNS as noise and designs strategies to mitigate their negative impact. For instance, Ding et al. (2020) reduces the likelihood of FNS being selected during negative sampling, while Zhang et al. (2024a) decreases the effect of noisy gradients from FNS through adversarial training. The second category of methods involves augmenting the training dataset with potential FNS (Wei et al., 2024; Liu et al., 2023).

Close to our work, Wei et al. (2024) propose using a base recommender trained on the target datasets to retrieve potential FNS, and using an LLM to filter irrelevant ones in graph-based recommendations. However, the key distinction in our approach is that we prioritize identifying semantically relevant items without initially considering collaborative information from the original dataset. In contrast, Wei et al. (2024) utilize both collaborative and semantic information to construct their augmented dataset. Our strategy offers two main advantages. Firstly, by initially ignoring collaborative information, we can retrieve a wider variety of items, reducing potential biases like popularity bias brought by collaborative information. This enhances the model's capability to learn the semantic connections between dialogue context and items. Secondly, our approach allows us to incorporate collaborative information at a later stage in a controlled manner, enabling a balanced integration of semantic relevance and collaborative information to suit various application scenarios.

3 Method

Figure 2 shows the overview of our approach. The main idea of our approach is to identify multiple semantically relevant items pre-dialogue (i.e. user query) to form the augmented dataset. Then, a two-stage training scheme is employed to help the recommender focus on the semantic relationship between user preferences and items in the first training stage and integrate collaborative information in the second training stage.

3.1 Data Synthesis Stage

In practice, since the candidate pool of the item database can be very large, it is impractical for the LLM to score each candidate individually. Therefore, we first use an LLM-based semantic retriever to retrieve a subset of potentially relevant items, thereby narrowing down the candidate set. Next, an LLM-based relevance scorer is employed to assign fine-grained relevance scores to these items.

3.1.1 Relevant Items Retrieval

Given item descriptive texts $\mathcal{I} = \{i_t\}_{t=1}^{n}$ and dialogue contexts $\mathcal{C} = \{c_t\}_{t=1}^{m}$, we first encode both of them into dense vector representations using

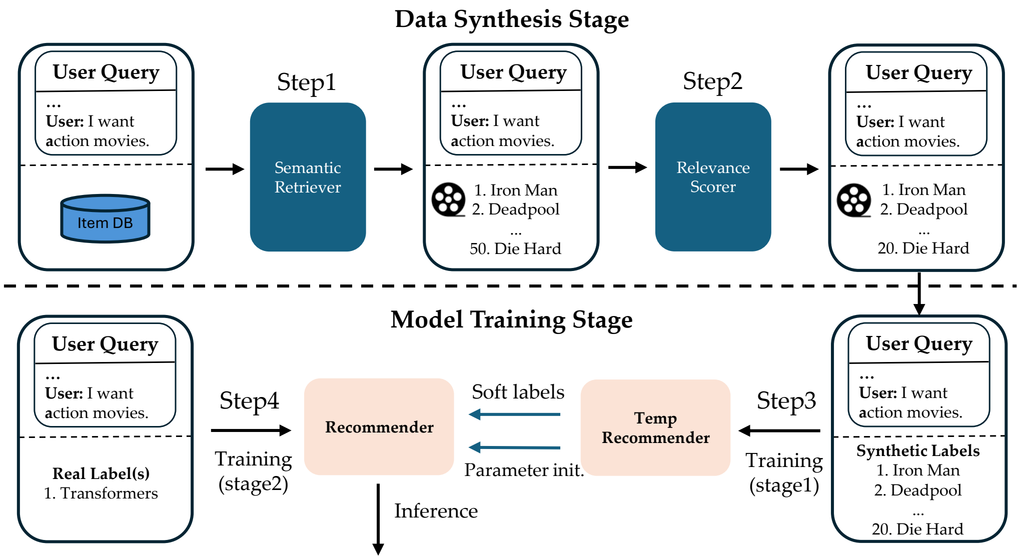

Figure 2: Overview of our approach. We first identify semantically relevant items as synthetic labels with LLMs-based semantic retriever and LLM-based relevance scorer. Then, we employ a two-stage training approach to pre-train the recommender with synthetic labels and fine-tune it with real labels.

an off-the-shelf LLM-based text encoder. Then, for each c_t in $\mathcal{C}$, we retrieve the top-k most similar items i_t through the max inner product search. These items are treated as potential candidates for further filtering in the next step.

This retrieval process relies solely on semantic information, without considering collaborative information, thereby avoiding potential biases such as popularity bias introduced by collaborative information in the dataset. This helps us cover a wide range of items instead of overly focusing on popular ones.

3.1.2 Relevance Estimation

After identifying potential relevant items, we use an LLM to perform detailed relevance estimations and filter out irrelevant samples. To enhance efficiency and scalability, we employ a state-of-the-art LLM (e.g. GPT-4) to generate context-item-score triples using a chain-of-thought prompting approach (Wei et al., 2022). These triples are then used to train a Gemma2-9b (Rivière et al., 2024), which assigns a relevance score ranging from 0 to 4 for each sample in the initial retrieval step. We retain context-item pairs with a relevance score exceeding a threshold value to construct the final synthetic training dataset $\mathcal{D}_{syn} = \{c_i, l_i\}_{i=1}^{n}$. We choose 3.5 as the threshold value for its superior performance in our experiments. See Appendix A for more details.

3.2 Model Training Stage

After we obtain the synthetic training dataset $\mathcal{D}_{syn}$, we can train the recommender with the synthetic training dataset $\mathcal{D}_{syn}$ and the original real-world training dataset $\mathcal{D}_{real}$. A two-stage training scheme is designed to fully utilize the semantic information in the synthetic dataset while leveraging the collaborative information in the real-world dataset in a controllable manner.

3.2.1 Pre-training the Recommender on the Synthetic Dataset

In the first stage, the recommender is pre-trained using the synthetic dataset. This step aims to expose the model to a diverse range of semantically relevant items, helping it develop a foundational understanding of the semantic relationships between user preferences and the items. This is particularly important in conversational recommendations. By using synthetic data, this approach avoids introducing biases found in real-world data, such as specific popularity distributions. A standard cross-entropy loss is applied for optimization:

$$L_{pre} = -\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} \cdot \log P(i,j) \tag{1}$$

Where N, M denotes the number of dialogue contexts and items in the training dataset. $y_{i,j}$ denotes a binary label that is equal to 1 when a user-item pair is labeled as positive in the synthetic dataset. $P(i,j)$ denotes a probability of the user-item pair being a positive pair calculated by the recommender. The method of calculation varies among different recommenders.

3.2.2 Fine-tuning the Recommender on the Real-world Dataset

After training with the synthetic dataset, the CRS can effectively provide recommendations based on semantic relevance between user preference and items. However, relying solely on semantic relevance may not be sufficient to make effective recommendations. In practice, collaborative information, referring to commonalities and trends in user behavior on target datasets/platforms, also plays an important role in making appropriate recommendations (He et al., 2024; Zhang et al., 2023; Zhu et al., 2024). This information is dataset/platform-dependent and can evolve due to semantic-independent factors (e.g. popularity trend), making it difficult for LLM to capture through semantic consistency (He et al., 2023). As a result, recommenders trained with LLM-suggested labels also struggle to consider collaborative information, leading to less appropriate and personalized recommendations.

To address this challenge, we propose to fine-tune the recommender on the original real-world dataset. This step ensures that the recommender can not only provide semantically relevant recommendations but also integrate collaborative information, enhancing its ability to reflect real-world user behaviors and preferences. Similar to the pre-training stage, a cross-entropy loss is applied:

$$L_{ce} = -\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} \cdot \log P(i,j) \tag{2}$$

Here, $y_{i,j}$ denotes positive labels in the original real-world dataset.

Furthermore, a label smoothing term can be introduced during fine-tuning to balance the recommender's reliance on semantic relevance with collaborative information in making recommendations:

$$L_{soft} = \sum_{i=1}^{N} D_{KL}(P(i), \hat{y_i}) \tag{3}$$

Where $\hat{y_{i,j}}$ is soft labels provided by the recommender before fine-tuning. $D_{KL}()$ denote the Kullback-Leibler Divergence. Finally, we optimize the recommender through:

$$L_{finetune} = L_{ce} + \alpha \cdot L_{soft} \tag{4}$$

Dataset	Train	Train Aug	Valid	Test
ReDial	29810	377313	3318	4036
INSPIRED	1404	15891	164	161

Table 1: Statistics of the datasets after pre-processing. Numbers mean positive context-item pairs in the corresponding split. "Train Aug" denotes our synthetic dataset.

Where a larger α indicates a decreased reliance on collaborative information, and vice versa.

4 Experiment

To verify the effectiveness of our approach, we conduct experiments to figure out whether our approach can enhance the recommendation quality of different recommenders, and whether our approach can outperform other data augmentation technologies and other LLM-powered baselines.

Backbone models. We consider BARCOR (Wang et al., 2022a), UniCRS (Wang et al., 2022b) and Llama2-7b-chat-hf (Touvron et al., 2023) (denoted as Llama2) as backbone models. These models are either the most representative CRS models or the most representative large language model. See Appendix C.1 for details.

Baselines. In addition to these backbone models, we introduce two additional data augmentation baselines.

- A false negative detection approach adapted from traditional recommender systems (Wei et al., 2024), which considers both collaborative information and semantic information in the data synthesis stage by using supervised recommenders in the item retrieval stage. We denote this baseline as **self-distillation (SD)**. See Appendix C.2 for details.
- **CFCRS** (Wang et al., 2023c): It generates synthetic training dialogues through counterfactual data simulation.

Datasets. We conduct our experiments on two widely used datasets, namely ReDial proposed by Li et al. (2018) and INSPIRED proposed by Hayati et al. (2020), which focus on conversational movie recommendation. We choose these two datasets because they are real-world datasets, which can lead to realistic evaluation. The data statics are shown in Table 1.

Previous work has found cases where items already mentioned in dialogue are also treated as

Models	ReDial			INSPIRED		
	R@1	R@10	R@50	R@1	R@10	R@50
Offline Dataset						
BARCOR	3.13	17.34	36.32	2.86	11.06	30.81
BARCOR + SD	3.52	19.95	41.08	2.48	19.38	35.78
BARCOR + CFCRS	3.50	18.98	38.46	**4.72**	20.50	35.16
BARCOR + ours	**4.31**	**21.26**	**43.84**	3.73	**21.12**	**43.11**
UniCRS	3.53	19.60	40.50	3.97	20.00	40.66
UniCRS + SD	3.08	20.27	42.26	4.64	20.53	38.94
UniCRS + CRCRS	**3.76**	19.91	40.60	**7.55**	22.78	**44.24**
UniCRS + ours	**3.76**	**20.93**	**42.74**	5.43	**22.91**	39.47
Llama2	3.93	20.74	41.34	4.46	11.68	34.16
Llama2 + SD	3.97	21.77	43.58	4.72	25.84	40.75
Llama2 + CRCRS	4.27	21.57	42.24	5.22	19.35	37.39
Llama2 + ours	**4.46**	**22.37**	**44.20**	**9.32**	**28.26**	**50.93**
User Simulator						
BARCOR	11.34	23.68	43.03	3.73	13.67	30.43
BARCOR + SD	21.95	36.74	54.71	8.07	27.95	35.40
BARCOR + CFCRS	19.87	34.17	51.23	14.91	24.22	37.27
BARCOR + ours	**27.77**	**44.69**	**63.35**	**24.84**	**40.37**	**61.49**
UniCRS	20.46	36.44	55.72	8.07	23.60	40.37
UniCRS + SD	22.07	39.12	59.78	**15.53**	24.84	40.99
UniCRS + CFCRS	**25.02**	41.50	59.96	14.91	23.60	40.99
UniCRS + ours	24.44	**43.65**	**63.42**	13.66	**29.81**	**45.34**
Llama2	24.33	42.83	61.89	4.97	12.42	34.78
Llama2 + SD	24.20	45.71	65.46	24.22	35.40	42.23
Llama2 + CFCRS	27.95	45.19	63.08	26.09	39.75	51.55
Llama2 + ours	**28.65**	**50.00**	**71.42**	**34.78**	**57.76**	**73.29**

Table 2: Main experimental results. The best methods in each group are marked in bold. R@k denotes Recall@K. Standard deviations and significance tests can be found in Table 8.

Models	ReDial			INSPIRED		
	R@1	R@5	R@10	R@1	R@5	R@10
BARCOR + ours	4.31	14.00	21.26	3.73	15.40	21.12
Llama2 + ours	**4.46**	**14.52**	**22.37**	**9.32**	**20.19**	**28.26**
Zero-Shot(GPT3.5)	3.57	11.54	16.33	4.35	11.80	17.39
Zero-Shot(GPT4o)	4.05	11.62	17.20	6.21	16.15	21.74

Table 3: Comparisons with strong proprietary LLMs under zero-shot setting.

labels, leading to the repetition of recommended items (He et al., 2023). To ensure more reliable training and evaluation, we have removed these samples from the dataset and filtered out repeated items from the recommender's prediction list following He et al. (2023).

Evaluation Setups. We conduct two types of evaluations, namely evaluation with the offline dataset and evaluation with the user simulator. The former is the most widely used evaluation protocol in previous work (Wang et al., 2022b,a). The latter is a newly proposed protocol that uses an LLM to interact with the CRS, which mimics real-world human behavior. We use iEvaLM (Wang et al., 2023b) as the user simulation protocol. Following previous work, we apply Recall@k k={1,10,50} as evaluation metrics.

Implementation Details. In the retrieval step, we retrieve the 50 most similar items for each dialogue. We use GritLM (Muennighoff et al.) as the text encoder due to its superior text representation ability. We follow Wang et al. (2023c) to perform the data processing. Since we focus on recommendation task, we implement dialogue generation by filling recommended items into templates like Wang et al. (2022b). We will discuss other variants in Section 6 and Appendix E. We run each experiment with the offline dataset evaluation five times and report the average numbers. Due to the extensive cost, we run each experiment with the user simulator once.

4.1 Main Results

Table 2 shows the results on two benchmark datasets, including both evaluation setups.

Compared with base models, our approach can consistently and significantly improve the performance of all three representative CRS models under both settings. These results demonstrated the effectiveness of our proposed approach. We will conduct further analysis on how these improvements are achieved in Section 5.1 and Section 5.2.

Compared with self-distillation, our approach yields greater improvements, confirming the effectiveness of disregarding collaborative information during data synthesis while integrating it through two-stage training. Further analysis of this will be provided in Section 5.3.

Compared to CFCRS, our approach demonstrates superior performance in most cases. This improvement can be attributed to our method's utilization of the LLM's semantic understanding ability to augment training labels based on existing dialogues, thereby ensuring semantic consistency between dialogues and labels. In contrast, CFCRS generates new dialogues through entity-based counterfactual data simulation, which lacks deep semantic understanding and may result in lower-quality dialogues. Additionally, our approach remains compatible with CFCRS, as the latter's generated dialogues still suffer from the false negative issue. Exploring the potential synergy between generating new dialogues (as in CFCRS) and augmenting labels (as in our method) is an interesting direction for future work.

4.2 Comparison with Zero-shot LLMs

To further verify the effectiveness of our approach, we compare our approach with proprietary LLMs in the zero-shot setting on offline datasets (He et al., 2023). As shown in Table 3, our approach achieves superior performance compared to strong proprietary LLMs. Notably, our method outperforms these proprietary models despite using significantly smaller models. This result highlights the importance of sufficient training in adapting LLMs for conversational recommendation tasks.

5 In-depth Analysis

In this section, we conduct an in-depth analysis of how our approach improves the overall performance of recommenders and the rationale behind our approach. We investigate how augmented data improve the performance of recommenders in Section 5.1, how our two-stage training scheme helps recommenders capture collaborative information in Section 5.2, how our approach outperforms the self-distillation baseline in Section 5.3, and how the quantity and quality of synthetic data affect the performance of recommenders in Section 5.4.

5.1 Effect of Augmented Data

Our approach can effectively retrieve multiple semantically relevant items based on user preferences, making the recommender's recommendations more semantically relevant, and alleviates the false negative issue. To evaluate the CRS's ability to provide semantically relevant items, we conduct experiments to assess the semantic relevance between the dialogue context and the recommended items. Inspired by the effectiveness of LLMs as automatic evaluators for conversational recommendations (Wang et al., 2023b; Huang et al., 2024), we use GPT-4 to assign relevance scores (0-4) to the top-k recommended items based on the dialogue context. As shown in Table 4, our approach consistently improves the recommenders' capacity to recommend items that are semantically relevant to the dialogue context, which is important for effective recommendations. To verify the reliability of GPT-4's assessments, we sample 625 dialogue-item pairs for human evaluation. The Pearson correlation between humans and GPT-4 is 0.7, indicating high agreement. Moreover, some of these augmented items are likely to be false negative samples, alleviating the false negative issue.

Several cases can be found in Appendix D to illustrate how augmented items affect the performance of the recommender.

Models	ReDial		INSPIRED	
	S@1	S@5	S@1	S@5
BARCOR	2.94	2.79	1.05	1.26
BARCOR + ours	**3.46**	**3.11**	**2.79**	**2.67**
UniCRS	3.04	2.80	1.82	1.90
UniCRS + ours	**3.29**	**3.04**	**2.20**	**2.15**
Llama2	3.21	2.97	0.79	1.23
Llama2 + ours	**3.34**	**3.08**	**3.17**	**2.85**

Table 4: Results on semantic relevance. S@k means the average relevance score of top-k recommendations.

Models	R@10	R@50	CS@10	CS@50
BARCOR	17.34	36.32	3.49	2.36
BARCOR + ours	**21.26**	**43.84**	**3.59**	**2.41**
w/o FT	12.55	32.27	1.80	1.38
mix	13.1	34.4	1.87	1.52
UniCRS	19.6	40.5	3.49	2.34
UniCRS+ours	**20.93**	**42.74**	**3.58**	**2.38**
w/o FT	12.09	30.94	1.73	1.36
mix	13.2	34.9	1.89	1.54
Llama2	20.74	41.34	**3.60**	**2.48**
Llama2+ours	**22.37**	**44.20**	3.53	2.44
w/o FT	11.52	31.10	1.62	1.29
mix	12.49	33.45	1.76	1.42

Table 5: Comparison of different training schemes on the ReDial dataset. "w/o FT" denotes removing the fine-tuning stage on the original dataset, "mix" denotes directly mixing augmented labels and original labels to train the recommender. "CS@k" denotes the average collaborative score of the top-k recommendation.

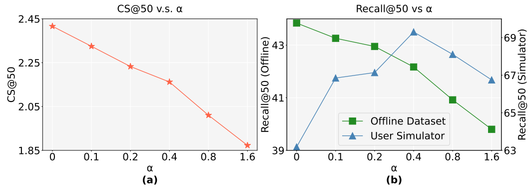

Figure 3: CS@50 and the Recall@50 v.s. different value of α.

5.2 Effect of Training Process

Our two-stage training process helps the recommender to capture collaborative information, resulting in more appropriate recommendations. To verify the importance of collaborative information in generating appropriate recommendations and to assess the effectiveness of our training scheme, we introduce two variants on ReDial dataset: One is to remove the fine-tuning stage, and the other is to directly mix both synthetic data and real data to train the recommender. To quantize how much collaborative information the recommender can capture, we first train a collaborative filtering model (e.g. SASRec (Kang and McAuley, 2018)) on the original dataset to model item co-occurrence patterns, which is one of the most important collaborative information (Sarwar et al., 2001; He et al., 2023). After that, we use the model to generate collaborative scores (denoted as CS) by calculating the co-occurrence probability (unnormalized logits) between the items recommended by the recommender and the items mentioned in the dialogue context, which assess the recommender's ability to capture item co-occurrence pattern. As shown in Table 5, training only with the synthetic dataset or mixing the two datasets results in low collaborative scores, leading to lower overall performance compared to our approach. This finding verified the importance of collaborative information in making appropriate recommendations and demonstrates the effectiveness of our approach in integrating collaborative information.

Including soft labels can control the integration of collaborative information, offering flexible control to adapt to different application scenarios To illustrate the impact of soft labels as described in Equation 3.2.1, we conduct additional experiments using both the offline dataset and the user simulator settings on the ReDial dataset. We use BARCOR trained with different α values in Equation 3.2.2 to illustrate how soft labels affect the recommender. Figure 3 (a) indicates that increasing α decreases the ability of the recommender to capture collaborative information, enabling it to focus more on semantic relevance while providing recommendations. This verifies the effectiveness of our approach to balancing semantic relevance and collaborative information through soft labels.

Figure 3 (b) shows that when α is less than 0.4, increasing α positively impacts the performance in the user simulator but negatively impacts the performance in the offline dataset. This difference arises because the user simulator tends to mention fewer items and provide more detailed textual preferences (Wang et al., 2023b), making semantic relevance more important than collaborative information when making recommendations. In contrast, users in the offline dataset tend to mention more items and offer less detailed textual preferences, so collaborative information is more important when making recommendations. An example can be found in Appendix F to illustrate the difference between the offline dataset and the user simulator. However, when α is larger than 0.4, continually increasing α can decrease the performance in both evaluations, which indicates the importance of collaborative information even when detailed textual preference is provided. Our findings suggest that the importance of collaborative information may vary across different scenarios, emphasizing the value of our approach to offer control over the integration of collaborative information.

5.3 Comparison with Self-distillation

Disregarding collaborative information and focusing on semantic relevance in the item retrieval stage help us cover a wider range of items, leading to better long-tail performance. As mentioned previously, the key difference between our approach and self-distillation lies in the item retrieval stage. In our approach, we use a semantic retriever and do not incorporate collaborative information from the dataset, while self-distillation uses the base recommender as the retriever, which takes collaborative information into account. To figure out how our strategy leads to better overall performance, we conduct an additional analysis on the ReDial dataset. Table 6 presents the number of relevant items retrieved by different retrievers, the number of distinct items covered by each retriever, and the long-tail performance for models using different retrievers (we denote items mentioned fewer than 4 times as long-tail items following Zhao et al.

Model	Count	Coverage	TR@10	TR@50
BARCOR+SD	368815	51.4%	1.24	3.52
BARCOR+ours	377313	98.1%	**2.14**	**8.7**
UniCRS+SD	385567	57.8%	1.90	6.03
UniCRS+ours	377313	98.1%	**2.71**	**9.77**
Llama2+SD	415950	71.7%	2.89	7.82
Llama2+ours	377313	98.1%	**3.01**	**9.37**

Table 6: Comparison with self-distillation. "Count" and "Coverage" indicate how many relevant items are retrieved by different retrievers and how many distinct items can different retrievers cover. "TR@k" denotes Tail_Recall@k, which computes the recall rate only for long-tail items.

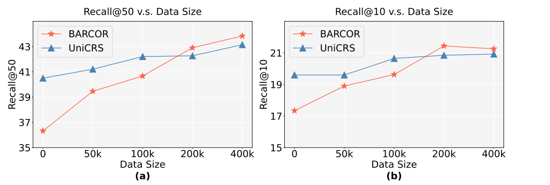

Figure 4: Performance v.s. different amounts of synthetic data on the ReDial dataset. "Data Size" denotes the number of synthetic data used during pre-training.

(2023)). While using the base recommender as a retriever can retrieve a similar number of relevant items, it covers significantly fewer distinct items compared to our semantic retriever. This is due to the popularity bias introduced by collaborative information, which causes the retriever to over-prioritize popular items while neglecting less popular ones. As a result, our approach shows better long-tail performance, which leads to better overall recommendation performance.

5.4 Effect of Data Quantity and Quality

Including more synthetic data can lead to better performance. As shown in figure 4, as the amounts of synthetic data increase, the performance of the recommender also increases, and the performance does not seem to converge. This finding supports our assumption that incorporating a variety of labels can help boost the recommender's recommendation performance. Meanwhile, the training time is directly proportional to the number of training samples. Therefore, in practice, others may need to balance the trade-off between training time and the performance of the CRS.

Without the LLM relevance scorer, the performance drops due to the noisy training data. As shown in table 7, if we remove the LLM filter, the performance drops consistently. This finding verifies the importance of fine-grained relevance estimation with the LLM relevance scorer, which can effectively filter irrelevant items.

Model	R@1	R@10	R@50
BARCOR+ours	**4.31**	**21.26**	**43.84**
w/o scorer	3.92	21.21	43.29
UniCRS+ours	**3.76**	**20.93**	**42.74**
w/o scorer	3.37	18.26	41.39

Table 7: Ablation study on the LLM relevance scorer on ReDial dataset. "w/o" scorer denotes removing the LLM relevance scorer.

6 Discussions on Dialogue Generation

Most CRS include a recommendation module to provide recommendations and a conversation module to generate response for the user. Unlike previous work that treats these two modules as a unified system (Wang et al., 2022a,b), we follow the modular design principle suggested by Zhang et al. (2024b) and optimize the recommendation module independently, enabling the recommendation module to flexibly collaborate with various generation modules, such as filling recommended items into template utterances or prompting LLMs to generate responses based on the recommendations. Although our approach focuses solely on recommendation quality, we still offer several suggestions and observations related to dialogue generation: (1) The modular design can potentially enhance dialogue generation quality by allowing the recommendation module to be combined with more advanced generation modules, rather than treating both modules as a unified system. (2) Improved recommendation quality can benefit dialogue generation and making the generated responses more coherent. For further illustrations, see Appendix E.

7 Conclusion

We present a data augmentation approach to augment multiple labels for each dialogue in the CRS training dataset. We first identify multiple semantically relevant items with an LLM-based semantic retriever and relevance scorer for each dialogue, which serve as augmented labels. Then, we design a two-stage training scheme to first pre-train the recommenders with augmented data to capture semantic information, and then fine-tune them with real-world data to integrate collaborative information in a controllable manner. Experimental results have shown a significant improvement in various settings, highlighting our approach's effectiveness..

Limitations

Though our approach shows promising results. However, we acknowledge two limitations of our approach. Firstly, we only focus on the movie domain, which is the most commonly used domain in conversational recommendation. In the future, if widely used datasets in other domains are available, we should adapt our approach to other domains to verify the generalization ability of our approach. Secondly, including additional training samples can add up to additional training costs. We leave it to future work to develop more cost-efficient training technology.

References

Bo Chen, Xinyi Dai, Huifeng Guo, Wei Guo, Weiwen Liu, Yong Liu, Jiarui Qin, Ruiming Tang, Yichao Wang, Chuhan Wu, et al. 2024. All roads lead to rome: Unveiling the trajectory of recommender systems across the llm era. *CoRR*.

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)*, pages 1803–1813.

Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems*, 33:1094–1105.

Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. 2024. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135*.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212*.

Luke Friedman, Sameer Ahuja, David Allen, Terry 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 preprint arXiv:2305.07961*.

Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan 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)*, pages 8142–8152.

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. *arXiv preprint arXiv:2308.10053*.

Zhankui He, Zhouhang Xie, Harald Steck, Dawen Liang, Rahul Jha, Nathan Kallus, and Julian McAuley. 2024. Reindex-then-adapt: Improving large language models for conversational recommendation. *arXiv preprint arXiv:2405.12119*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Chen Huang, Peixin Qin, Yang Deng, Wenqiang Lei, Jiancheng Lv, and Tat-Seng Chua. 2024. Concept–an evaluation protocol on conversation recommender systems with system-and user-centric factors. *arXiv preprint arXiv:2404.03304*.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):1–36.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Minjin Kim, Minju Kim, Hana Kim, Beong-woo Kwak, SeongKu Kang, Youngjae Yu, Jinyoung Yeo, and Dongha Lee. 2024. Pearl: A review-driven persona-knowledge grounded conversational recommendation dataset. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1105–1120.

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*, pages 304–312.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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

Tingting Liang, Chenxin Jin, Lingzhi Wang, Wenqi Fan, Congying Xia, Kai Chen, and Yuyu Yin. 2024. Llm-redial: A large-scale dataset for conversational recommender systems created from user behaviors with llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8926–8939.

Guo Lin and Yongfeng Zhang. 2023. Sparks of artificial general recommender (agr): Early experiments with chatgpt. *arXiv preprint arXiv:2305.04518*.

Xiaoyang Liu, Chong Liu, Pinzheng Wang, Rongqin Zheng, Lixin Zhang, Leyu Lin, Zhijun Chen, and Liangliang Fu. 2023. Ufnrec: utilizing false negative samples for sequential recommendation. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 46–54. SIAM.

Yu Lu, Junwei Bao, Zichen Ma, Xiaoguang Han, Youzheng Wu, Shuguang Cui, and Xiaodong He. 2023. August: an automatic generation understudy for synthesizing conversational recommendation datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10538–10549.

Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. Revcore: Review-augmented conversational recommendation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1161–1173.

Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *ICLR 2024 Workshop: How Far Are We From AGI*.

Peixin Qin, Chen Huang, Yang Deng, Wenqiang Lei, and Tat-Seng Chua. 2024. Beyond persuasion: Towards conversational recommender system with credible explanations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4264–4282.

Mathieu Ravaut, Hao Zhang, Lu Xu, Aixin Sun, and Yong Liu. 2024. Parameter-efficient conversational recommender system as a language processing task. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–165.

Zhaochun Ren, Zhi Tian, Dongdong Li, Pengjie Ren, Liu Yang, Xin Xin, Huasheng Liang, Maarten de Rijke, and Zhumin Chen. 2022. Variational reasoning about user preferences for conversational recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–175.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, et al. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. 2022a. Barcor: Towards a unified framework for conversational recommendation systems. *arXiv preprint arXiv:2203.14257*.

Xi Wang, Hossein Rahmani, Jiqun Liu, and Emine Yilmaz. 2023a. Improving conversational recommendation systems via bias analysis and language-model-enhanced data augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3609–3622.

Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023b. Rethinking the evaluation for conversational recommendation in the era of large language models. *arXiv preprint arXiv:2305.13112*.

Xiaolei Wang, Kun Zhou, Xinyu Tang, Wayne Xin Zhao, Fan Pan, Zhao Cao, and Ji-Rong Wen. 2023c. Improving conversational recommendation systems via counterfactual data simulation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2398–2408.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022b. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1929–1937.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 806–815.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Memocrs: Memory-enhanced sequential conversational recommender systems with large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2585–2595.

Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. 2022. Improving conversational recommendation systems' quality with context-aware item meta-information. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 38–48.

Ting Yang and Li Chen. 2024. Unleashing the retrieval potential of large language models in conversational recommender systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 43–52.

An Zhang, Leheng Sheng, Zhibo Cai, Xiang Wang, and Tat-Seng Chua. 2024a. Empowering collaborative filtering with principled adversarial contrastive loss. *Advances in Neural Information Processing Systems*, 36.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B 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*, pages 270–278.

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*, pages 177–186.

Zeyuan Zhang, Tanmay Laud, Zihang He, Xiaojie Chen, Xinshuang Liu, Zhouhang Xie, Julian McAuley, and Zhankui He. 2024b. Recwizard: A toolkit for conversational recommendation with modular, portable models and interactive user interface. *arXiv preprint arXiv:2402.15591*.

Zhipeng Zhao, Kun Zhou, Xiaolei Wang, Wayne Xin Zhao, Fan Pan, Zhao Cao, and Ji-Rong Wen. 2023. Alleviating the long-tail problem in conversational recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 374–385.

Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2022. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):9920–9931.

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 & data mining*, pages 1006–1014.

Yuanhang Zhou, Kun Zhou, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and He Hu. 2022. C^2-crs: Coarse-to-fine contrastive learning for conversational recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1488–1496.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative large language model for recommender systems. In *Proceedings of the ACM on Web Conference 2024*, pages 3162–3172.

Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 85–93.

Jie Zou, Evangelos Kanoulas, Pengjie Ren, Zhaochun Ren, Aixin Sun, and Cheng Long. 2022. 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*, pages 2319–2324.

A Details of the Relevance Estimation Step

As discussed in Section 3.1.2, we employ an LLM-based relevance scorer to filter noisy candidates. First, we sample five items pre-dialogue and use GPT-4 (GPT-4o-mini) to assign each a relevance score (0–4) based on its alignment with the dialogue. The scoring prompt is provided in Table 18. These scored samples are then used to train a Gemma2-9b model for efficient inference, optimized via MSE loss.

$$\mathcal{L}_{ce} = \frac{1}{N} \sum_{i=1}^{N} \left((f_\theta(c_i, l_i) - s_i)^2 \right) \tag{5}$$

Where $f_\theta(c_i, l_i)$ denotes the score assigned by the Gemma2-9b model to the dialogue-product pair. Specifically, the computation of $f_\theta(c_i, l_i)$ is performed as follows: the dialogue c_i and item description l_i are concatenated into the prompt sequence $t_{1:n}$, where n denotes the length of the prompt. This sequence is then fed into the Gemma2-9b model. The last hidden state h_n from the final hidden layer of the model is extracted and passed through a trainable fully connected layer W, which maps the 4096-dimensional h_n to a 1-dimensional score output $f_\theta(c_i, l_i)$:

$$f_\theta(c_i, l_i) = W(h_n) \tag{6}$$

The Gemma2-9B model is fine-tuned using LoRA (Hu et al.) with a rank of 8.

B Standard Deviations

We report the mean, the standard deviations, and the significance test in our main experiments with offline dataset evaluation in Table 8.

C Introduction and Implementation Details of Baselines

C.1 Baseline CRS Models

We consider three baseline CRS models:

- BARCOR (Wang et al., 2022a): It Implements both the recommendation and conversation modules in a unified BART (Lewis et al., 2019).
- UniCRS (Wang et al., 2022b): It implements both the recommendation and conversation modules in a unified DialoGPT (Zhang et al., 2020) with knowledge-enhanced soft prompting.
- Llama2 (Touvron et al., 2023): It preform recommendation by fine-tune Llama2 on the target dataset. We use Llama2-7b-chat-hf version.

These models are either the most representative CRS models or the most representative large language model. For BARCOR, since it is not open-sourced, we follow Wang et al. (2023b) and Wang et al. (2023c) to implement it. For Llama2, we use Lora (Hu et al.) with rank 4 to fine-tune it on downstream datasets.

C.2 False Negative Detection Baseline

Recently, as part of their contributions, Wei et al. (2024) introduced a false negative detection approach in traditional recommender systems to first retrieve potential false negative samples with a supervised recommender, and then use an LLM to score the relevance between the items and the user preference. These items with high relevance scores will be used as augmented labels to construct the augmented dataset. After that, they use a one-stage training scheme to mix a certain proportion (refers to ω_3 in the original paper) of labels from the original dataset and labels from the augmented dataset together to train the recommender.

Our implementation of this baseline is as follows. Firstly, we use the base recommender trained on the target dataset itself to retrieve items. Then, we use the same LLM relevance scorer as our approach to score the relevance between the items and the dialogues. These items with high relevance scores will be used as augmented labels to construct the augmented dataset. After that, the same two-stage training scheme as our approach is used.

There are two differences between our implementation and the original implementation. Firstly, in the item retrieval stage, we use a different recommender from the original implementation. Secondly, we use a different training scheme from the original implementation. The reason why we use such an implementation is that. For the item retrieval stage, since Wei et al. (2024) and us do not share the same task, we can not use the same recommender as Wei et al. (2024) to retrieve items. So, we simply use the base recommender to implement the idea of considering both collaborative information and semantic information in the item retrieval stage. For the training stage, the reason why we use the same two-stage training scheme as our approach instead of the original one-stage

Models	ReDial			INSPIRED		
	R@1	R@10	R@50	R@1	R@10	R@50
Offline Dataset						
BARCOR	3.13 (0.27)	17.34 (0.29)	36.32 (0.21)	2.86 (1.29)	11.06 (1.83)	30.81 (2.83)
BARCOR + SD	3.52 (0.31)	19.95 (0.26)	41.08 (0.63)	2.48 (1.13)	19.38 (1.29)	35.78 (2.87)
BARCOR + CFCRS	3.50 (0.11)	18.98 (0.33)	38.46 (0.46)	**4.72 (0.56)**	20.50 (1.96)	35.16 (1.62)
BARCOR + ours	**4.31***† **(0.21)**	**21.26***† **(0.49)**	**43.84***† **(0.26)**	3.73 (1.52)	**21.12*** **(0.98)**	**43.11***† **(1.68)**
UniCRS	3.53 (0.12)	19.60 (0.31)	40.50 (0.25)	3.97 (1.05)	20.00 (1.18)	**40.66 (1.37)**
UniCRS + SD	3.08 (0.17)	20.27 (0.39)	42.26 (0.51)	4.64 (0.66)	20.53 (0.66)	38.94 (0.98)
UniCRS + CRCRS	**3.76 (0.23)**	19.91 (0.22)	40.60 (0.35)	**7.55 (0.76)**	22.78 (1.79)	**44.24 (1.44)**
UniCRS + ours	**3.76**† **(0.21)**	**20.93*** **(0.59)**	**42.74*** **(0.39)**	5.43 (1.36)	**22.91***† **(1.37)**	39.47 (0.36)
Llama2	3.93 (0.35)	20.74 (0.53)	41.34 (0.79)	4.46 (1.21)	11.68 (1.02)	34.16 (1.58)
Llama2 + SD	3.97 (0.22)	21.77 (0.40)	43.58 (0.43)	4.72 (0.94)	25.84 (1.13)	40.75 (1.84)
Llama2 + CRCRS	4.27 (0.29)	21.57 (0.34)	42.24 (0.53)	5.22 (1.29)	19.35 (1.24)	37.39 (1.11)
Llama2 + ours	**4.46***† **(0.30)**	**22.37***† **(0.31)**	**44.20*** **(0.73)**	**9.32***† **(0.88)**	**28.26*** **(2.18)**	**50.93***† **(1.13)**

Table 8: Mean and standard deviations (stds.) of main experimental results in offline dataset evaluation. The numbers outside the parentheses are the mean and the numbers inside the parentheses are stds. $*$denotes significant improvement compared with the base model, and † denotes significant improvement compared with self-distillation. ($p<0.05$)

Model	ω_3	R@1	R@10	R@50
Original One-stage Training				
BARCOR	0.2	2.85	16.38	37.41
BARCOR	0.4	3.07	17.78	38.45
BARCOR	0.6	3.07	17.66	38.33
BARCOR	0.8	2.70	18.14	39.12
BARCOR	1	2.80	18.38	39.74
BARCOR	2	2.87	17.89	40.14
BARCOR	5	2.45	17.98	40.70
BARCOR	10	2.40	17.91	**41.10**
Our Two-stage Training				
BARCOR	-	**3.52**	**19.95**	41.08

Table 9: Performance comparison of our two-stage training scheme and the original one-stage training scheme in implementing the self-distillation baseline with BARCOR on the ReDial dataset. ω_3 denotes the proportion of synthetic data per batch.

training scheme is twofold. Firstly, as shown in Table 9, we find that our two-stage training scheme works better than the original one-stage training scheme. Secondly, it helps us to fairly compare our item retrieval approach that disregards collaborative information with their item retrieval approach that regards collaborative information, which helps us gain a deeper understanding of how bias introduced by collaborative information could affect the final performance of the recommender.

D Case Study for Recommendation

To gain a qualitative understanding of how our approach enhances the recommender's performance, we present several cases in Tables 10, 11, and 12. As shown in the first lines of these cases, without augmented labels, the recommender struggles to suggest items that align semantically with the user's preferences. For instance, in the second case, when a user requests action and funny movies, the recommender, before data augmentation, suggests dramas instead, which are not semantically relevant to the user preference. This leads to an unsatisfactory user experience. after applying data augmentation, the recommender effectively suggests items that match the user's preferences more precisely. For example, in the second case, after data augmentation, the recommender successfully recommends several action comedy movies when the user requests action and funny movies, which are semantically relevant to the user preference. This leads to satisfactory user experience.

To understand the underlying reasons, we examined similar dialogues between users and the recommender from the training set. As shown in the second lines of these cases, a significant number of semantically relevant items are added to the augmented training dataset. This improvement makes the recommender more likely to suggest these semantically relevant items.

Meanwhile, in these cases, the items that are accepted by the user in the test set are added into the augmented label set for the user that expresses a similar preference, which indicates that these items are highly likely to be false negative samples. Without data augmentation, these items are treated as negative samples, which directly reduces the probability for the recommender to recommend these items when encountering a user with a similar preference, leading to decreased user satisfaction.

E Case Study for Dialogue Generation

Following the modular design principle proposed by Zhang et al. (2024b) (see Section 6), we independently optimize the recommendation module. Below, we first discuss the benefits of this design principle and then demonstrate how our approach—which improves recommendation quality—can also enhance dialogue generation performance.

Advantages of modular design. While prior work often treats the recommendation and generation modules as an unified system (Chen et al., 2019; Zhou et al., 2020; Wang et al., 2022a,b), Zhang et al. (2024b) argue that the two modules can be treated separately. We strongly endorse this principle, as pairing an existing recommendation module with a more powerful generation module (e.g., an LLM-based generator) can significantly improve how recommendations are presented to users.

To illustrate this, Table 13 compares responses generated by BARCOR as a unified system versus responses produced by combining BARCOR's recommendation module with a strong LLM-based generator (denoted as BARCOR (modular), where we use Llama2-7b-chat-hf for generation, see Table 17 for the prompt). The results show that the unified BARCOR system produces responses with limited informative explanations due to its weaker generation module. In contrast, when BARCOR's recommendation module is paired with an LLM-based generator, the responses become more detailed and engaging, leading to a better user experience.

How our approach improve dialogue generation performance. To further demonstrate how our approach can improve dialogue generation, we conduct a case study using Llama2-7b-chat-hf as the generation module and compare responses generated from two different recommendation modules: (1) the original BARCOR recommender and (2) the BARCOR recommender enhanced by our approach (see Table 17 for the prompt).

As shown in Table 14, the original BARCOR recommender fails to align with user preferences, resulting in misleading responses. For example, it incorrectly describes Dunkirk (2017) as having a mix of action and humor, despite the film containing almost no comedic elements. Such inaccuracies can degrade user trust and satisfaction. In contrast, our improved recommender provides more relevant recommendations, leading to responses that are both factually correct and coherent.

This case study highlights how enhancing recommendation quality can indirectly improve dialogue generation. Future work could further explore synergies between different recommendation and generation modules.

F An Example Dialogue of the ReDial Dataset and the User Simulator

To illustrate the difference between the ReDial dataset and the user simulator, we present a dialogue example from the ReDial dataset in Table15 and the user simulator in Table16, respectively. The user in the ReDial dataset tends to mention more items and provide less detailed textual preference, while the user simulator tends to provide more detailed textual preference and mention fewer items.

A Dialogue Context in the Test Set	BARCOR	BARCOR+ours
User: Hi I am looking for a movie like Super Troopers (2001)	1. Bad Boys (1995) 2. The Waterboy (1998) 3. Jerry Maguire (1996) 4. Click (2006) 5. American Pie (1999)	1. Super Troopers 2 (2018) 2. 21 Jump Street (2012) 3. 22 Jump Street (2014) 4. Lethal Weapon (1987) 5. Police Academy (1984)
A Similar Example in the Training Set	**Original Label**	**Augmented Labels**
User: Hi there! **Recommender:** hello, what kind of movies are you looking to watch? **User:** Could you recommend a comedy movie? I'm looking for some silly cop movies for my husband, something like Super Troopers (2001) or Hot Fuzz (2007)	Lethal Weapon (1987)	Let's Be Cops (2014) Super Troopers 2 (2018) 21 Jump Street (2012) Police Academy (1984) The Other Guys (2010) ... (other 8 movies)

Table 10: A case study. Lethal Weapon (1987) and Police Academy (1984) are finally accepted by the user.

A Dialogue Context in the Test Set	BARCOR	BARCOR+ours
User: Hello! How are you? **Recommender:** Hi I am good How are you **User:** I'm fine. Do you have any good suggestions for a good movie? **Recommender:** What kinds of movies do you like? Do you like dramatic movies? **User:** I love action and funny movies, so I looking for something that combines the two,	1. Titanic (1997) 2. Dunkirk (2017) 3. 50 First Dates (2004) 4. Juno (2007) 5. Mr. & Mrs. Smith (2005)	1. Jumanji (2017) 2. Rush Hour (1998) 3. Deadpool (2016) 4. Thor: Ragnarok (2017) 5. Black Panther (2018)
A Similar Example in the Training Set	**Original Label**	**Augmented Labels**
User: Hello **Recommender:** Hello! how are you? **User:** I'm good thanks **Recommender:** What can I recommend for you today? **User:** I enjoy action and comedy movies.	Superbad (2007)	Spy (2015) Knight and Day (2010) Central Intelligence (2016) Rush Hour (1998) Let's Be Cops (2014) ... (other 27 movies)

Table 11: A case study. Rush Hour (1987) and Deadpool (2016) are finally accepted by the user.

A Dialogue Context in the Test Set	BARCOR	BARCOR+ours
Recommender: Hello! Do you like comedies? **User:** I do! I really enjoy stoner comedies! **Recommender:** Me too! **User:** Know any good ones?	1. Dumb and Dumber (1994) 2. Bridesmaids (2011) 3. Superbad (2007) 4. The Heat (2013) 5. Office Space (1999)	1. The Big Lebowski 2. This Is the End (2013) 3. Dazed and Confused (1993) 4. Pineapple Express (2008) 5. Superbad (2007)
A Similar Example in the Training Set	**Original Label**	**Augmented Labels**
User: Hey! hows the day going **Recommender:** Pretty good and yours/ **User:** would you be able to recommend some good movies today? im pretty chill here **Recommender:** Me too, do you want some stoner comedy suggestions? **User:** Actually yes that would be lovely	Super Troopers (2001)	Up in Smoke (1978) Harold & Kumar How High (2001) Pineapple Express (2008) The Big Lebowski ... (other 27 movies)

Table 12: A case study. Pineapple Express (2008) is finally accepted by the user.

Context	User: Hi I am looking for a movie like Super Troopers (2001), please give me a suggestion.
Responses	**BARCOR:** Hi, I have some good movies. Have you seen 21 Jump Street (2012)?
	BARCOR (modular): Sure, I'd be happy to help! Based on your dialogue history, it seems like you're looking for a movie similar to Super Troopers (2001). Given that, I would recommend the movie 21 Jump Street (2012) to you. Here's why: 21 Jump Street is a comedy film that, like Super Troopers, is a spoof of action movies and law enforcement. It features a similar blend of humor, action, and memorable characters, making it a great match for fans of Super Troopers. The movie also has a similar tone and style to Super Troopers, with a focus on absurd humor and over-the-top situations. I hope you enjoy 21 Jump Street! Let me know if you have any other questions or if you'd like more recommendations.

Table 13: A case study of generated response. BARCOR denotes treating BARCOR as a unified system, while BARCOR (modular) denotes combining BARCOR recommender with the LLM generator.

Context	User: Hello! How are you? Recommender: Hi I am good How are you User: I'm fine. Do you have any good suggestions for a good movie? Recommender: What kinds of movies do you like? Do you like dramatic movies? User: I love action and funny movies, so I looking for something that combines the two,
Responses	**BARCOR:** Great! Based on your interest in action and funny movies, I would recommend "Titanic" (1997) and "Dunkirk" (2017). Both of these movies have a mix of action and humor that you might enjoy. "Titanic" is a classic romantic drama with plenty of action-packed scenes, while "Dunkirk" is a historical drama that also has some thrilling moments. I think you'll find them both to be great choices!
	BARCOR+ours: Great! Based on your interest in action and funny movies, I would recommend "Jumanji" (2017) and "Rush Hour" (1998). Both of these movies have a great balance of action and humor that I think you'll enjoy. "Jumanji" is a thrilling adventure film with plenty of laughs, while "Rush Hour" is a classic buddy cop movie with a great mix of action and comedy. I hope you check them out and let me know what you think!

Table 14: A case study of generated response. BARCOR+ours denotes applying our approach on BARCOR recommender.

User:	Hello
Recommender:	hi
User:	i'm open to any movie What would you recommend me? i liked Avengers: Infinity War (2018)
Recommender:	The Big Lebowski is my fav
User:	i also like scary movies i watched The Big Lebowski
Recommender:	how about House of 1000 Corpses (2003)
User:	i also watched It (2017) NO, i have not seen that one i guess i would try that one tomorrow i also watched It (1990) the old one Which another one would you recommend
Recommender:	its by rob Zombie
User:	A funny one Like Click (2006) by Adan Sandler

Table 15: An example from the ReDial dataset. Multiple items are mentioned by the user.

User:	Hi I am looking for a movie like Super Troopers (2001).
Recommender:	Hi, I have some good movies.
User:	Great! I love movies with a similar comedic style and maybe some wacky characters. What do you have in mind?
Recommender:	Have you seen Bridesmaids (2011)?
User:	No, I haven't seen that one. I'm really looking for something with a more slapstick, over-the-top kind of humor, maybe with a focus on a group of misfits trying to succeed in a challenging environment.
Recommender:	Have you seen Superbad (2007)?
User:	Sorry, No, I haven't seen that one either. I'm looking for something with a bit more of a classic, buddy-cop feel, with lots of physical comedy and maybe some training montage scenes.
Recommender:	Rush Hour (1998) is a good one.
User:	Sorry, I'm looking for something a bit more lighthearted and silly, with a focus on a group of recruits learning the ropes in a somewhat unconventional way.

Table 16: An example from the user simulator. The user simulator provide detailed textual preference.

Given dialogue history between a user and a recommender, generate a response that recommend the target items for the user. ~~~~~~~~~~~~~~~~~~ dialogue history: **{context}** ~~~~~~~~~~~~~~~~~~ items to recommend: **{item}**

Table 17: Prompt template for the LLM generator

You are provided with a dialogue history between a user and a recommender system, along with the system's top k movie recommendations. Your task is to score each recommendation on a scale from 0 to 4 based on how well it aligns with the user's expressed preferences.
∼∼∼∼∼∼∼
4: The recommendation perfectly aligns with all expressed aspects of the user's preferences.
3: The recommendation aligns well with the main aspects of the user's preferences but lacks one or two minor aspects.
2: The recommendation aligns with some but not all major aspects of the user's preferences.
1: The recommendation aligns minimally with the user's preferences, missing several key aspects.
0: The recommendation does not align with the user's preferences at all.
You can give non-integer scores if necessary
∼∼∼∼∼∼∼
Here is an illustration of the scoring criteria.:
If a user said, 'I like movies similar to Super Troopers (2001), especially its humorous style!' that may indicate the user's preference for the comedy genre, absurd humor, and perhaps law enforcement themes. Therefore, if a recommender system recommends movies such as 'Hot Fuzz (2007)' or 'The Other Guys (2010),' those should be 4-point recommendations because they align with the user's preferences for the comedy genre, absurd humor styles, and law enforcement themes. If a recommender system recommends a movie like 'Superbad (2007),' this should be a 3-point recommendation since it aligns with the user's preference for the comedy genre and absurd humor styles but lacks a law enforcement theme. If a recommender system recommends a movie like 'Police Story (1985),' this may be a 2-point recommendation since it has law enforcement themes and some comedy elements due to Jackie Chan's acting style. If a recommender system recommends a movie like 'Infernal Affairs (2002),' this may be a 1-point recommendation since it has law enforcement themes but lacks any comedy or humor elements. If a recommender system recommends a movie like 'It (2017),' this should be a 0-point recommendation since 'It (2017)' is not relevant to the user's preferences.
Dialogue history:
{context}
recommender's recommendation:
{rec}
Give brief reason and end with a JSON format as follows:
{"<movie>": <score>, "<movie>": <score>, ...}
Note: Replace <movie> and <score> with the movie name (year if exist) and actual score you have assigned to each movie.

Table 18: Prompt template for relevance estimation