XRec: Large Language Models for Explainable Recommendation

Qiyao Ma, Xubin Ren, Chao Huang*
University of Hong Kong
{martin.qyma, xubinrencs, chaohuang75}@gmail.com

Abstract

Recommender systems help users navigate information overload by providing personalized recommendations aligned with their preferences. Collaborative Filtering (CF) is a widely adopted approach, but while advanced techniques like graph neural networks (GNNs) and self-supervised learning (SSL) have enhanced CF models for better user representations, they often lack the ability to provide explanations for the recommended items. Explainable recommendations aim to address this gap by offering transparency and insights into the recommendation decision-making process, enhancing users' understanding. This work leverages the language capabilities of Large Language Models (LLMs) to push the boundaries of explainable recommender systems. We introduce a model-agnostic framework called XRec, which enables LLMs to provide comprehensive explanations for user behaviors in recommender systems. By integrating collaborative signals and designing a lightweight collaborative adaptor, the framework empowers LLMs to understand complex patterns in user-item interactions and gain a deeper understanding of user preferences. Our extensive experiments demonstrate the effectiveness of XRec, showcasing its ability to generate comprehensive and meaningful explanations that outperform baseline approaches in explainable recommender systems. We open-source our model implementation at: https://github.com/HKUDS/XRec.

1 Introduction

With the overwhelming abundance of content and products available in online platforms, users frequently encounter the daunting challenge of information overload. In response, recommender systems emerge as indispensable tools that aim to alleviate this burden. These systems effectively filter through the vast array of options and present users with tailored recommendations that are both relevant and personalized, aligning with their unique preferences and interests (Zhang et al., 2019b).

Among the diverse range of recommendation techniques available, Collaborative Filtering (CF) framework emerges as a prominent and widely embraced approach within the field of recommender systems. CF operates on the fundamental premise that users who have demonstrated similar preferences in the past, such as common item ratings or similar purchase histories, are likely to exhibit comparable preferences when it comes to future recommendations (Chen et al., 2021).

In recent years, the field of collaborative filtering algorithms has undergone a remarkable revolution with the emergence of deep learning techniques. This transformative wave has brought about the integration of diverse neural network architectures, including Attention mechanisms (Chen et al., 2017), Graph Neural Networks (GNNs) (He et al., 2020), and Self-Supervised Learning (SSL) (Xia et al., 2023). Notably, the incorporation of GNNs in collaborative filtering models has yielded significant advancements by effectively capturing complex relational information and enhancing recommendation performance while preserving high-order collaborative dependencies. Moreover, self-supervised recommender systems have emerged as a promising solution to address the challenge of data sparsity. These systems leverage self-supervised learning signals to augment the available data, aiming to enhance recommendations.

While existing collaborative filtering models excel at providing accurate recommendation results, there remains a critical aspect that has not received adequate attention: understanding the underlying reasons behind observed user-item interactions. Explainable recommendations aim to address this gap by providing transparency to users, offering insights into the decision-making process behind recommendations. This not only enhances users' understanding of their own preferences, but also fosters trust in recommendation algorithms.

Findings of the Association for Computational Linguistics: EACL 2024, pages 391–402
November 12-16, 2024

Several research studies have dedicated their focus to generating explanations for user-item interactions. Notably, Att2Seq (Dong et al., 2017) and NRT (Li et al., 2017) employ attention mechanisms and recurrent neural networks (RNNs) to generate textual explanations. Recent advancements have further explored the utilization of Transformer (Li et al., 2021) and GPT2 (Li et al., 2023) for text generation, providing valuable insights into recommendation results. However, these approaches face a common challenge arising from the limited availability of explanation data, which hinders their ability to generate high-quality explanations. It is also important to emphasize that ID-based methods heavily rely on ID embeddings, resulting in limited generalization capabilities and difficulties when adapting to unseen users and items in a zero-shot recommendation scenario.

Presented Work. In light of the recent advancements in Large Language Models (LLMs), our primary objective is to push the boundaries of explainable recommender systems by harnessing the exceptional language capabilities of LLMs. To this end, we introduce model-agnostic XRec, a groundbreaking collaborative instruction-tuning framework that enables large language models to provide comprehensive explanations for user behaviors within recommender systems. Within our XRec framework, we equip LLMs with the unique ability to comprehend the intricate patterns of user-item interactions through the integration of collaborative signals via a collaborative instruction-tuning paradigm. In order to bridge the representation space of collaborative relationships and the language semantic space, we design a lightweight collaborative adaptor that incorporates behavior-aware collaborative signals into the LLMs, facilitating a deeper understanding of user preferences.

We conducted a series of thorough experiments to validate the effectiveness and superior performance of our proposed framework, XRec, in generating comprehensive and meaningful explanations within recommender systems. In addition to that, we conducted ablation studies and investigated the robustness of our model, providing further evidence of its effectiveness.

2 Preliminaries

2.1 Explainable Recommendation

Explainable Recommendation is essential in recommender systems as it clarifies the underlying reasons for user-item interactions. Our primary goal is to create clear textual explanations that allow us to understand the rationale behind each recommendation. Specifically, for each interaction between a user u and an item i, the explanations generated can be described as follows:

$$\textit{explanation}(u, i) = \textit{generate}(u, i, \mathcal{X}_u, \mathcal{X}_i, \tau) \quad (1)$$

In this context, $\mathcal{X}_u$ and $\mathcal{X}_i$ represent the interaction histories of user u and item i, respectively. The symbol τ denotes any additional side information related to both the users and the items.

Building upon recent advancements in text-based profile generation (Ren et al., 2024; Xi et al., 2023), we enhance the explanation generation paradigm in recommender systems. Our method involves incorporating textual information into the generation of item profiles, utilizing a predefined item prompt ($\mathcal{P}_I$) and item description ($\mathcal{D}$).

$$\mathcal{I} = \textit{LLMs}(\mathcal{P}_I, \mathcal{D}) \quad (2)$$

In addition to item descriptions, we extend our approach to user profiles by considering their interactions with previously profiled items. This is achieved by sampling the items that the user has interacted with, resulting in a more comprehensive representation of their preferences and interests.

$$\mathcal{U} = \textit{LLMs}(\mathcal{P}_U, \{\mathcal{I}_i : i \in \mathcal{N}_u\}) \quad (3)$$

We denote the set of items interacted with by user u as $\mathcal{N}_u$, and use $\mathcal{P}_U$ as the user profile prompt.

2.2 Graph Collaborative Filtering

Graph Neural Networks (GNNs) have proven to be effective frameworks for capturing collaborative relationships between users and items, taking into account high-order dependencies. Through multiple rounds of message passing, nodes in the user-item interaction graph assimilate information and generate embeddings that capture these collaborative relationships (Wang et al., 2019). To encode the user-item interaction graph $\mathcal{G}$ using L layers of GNNs, the l^{th} layer embedding of a user node u or an item node i is computed as follows:

$$e_u^{(l)} = \textit{AGG}\left(e_u^{(l-1)}, \{e_i^{(l-1)} \mid i \in \mathcal{N}_u\}\right) \quad (4)$$

In the context of graph collaborative filtering, $\mathcal{N}_u$ refers to the neighborhood of node u, while $\textit{AGG}(\cdot)$ is the aggregation function, which can vary across

different models. By utilizing L layers of propagation, graph neural networks (GNNs) generate $L+1$ distinct embeddings for each node, namely $(e^{(0)}, e^{(1)}, \ldots, e^{(L)})$. These embeddings are then concatenated to form the final node embedding.

The embeddings of users and items are generated with the objective of maximizing the probability, considering the historical user interactions:

$$p(\mathbf{e}|\mathcal{X}) \propto p(\mathcal{X}|\mathbf{e})p(\mathbf{e}) \tag{5}$$

For each user in the set $U = u_1, u_2, \ldots, u_m$ and each item in the set $I = i_1, i_2, \ldots, i_n$, where $\mathcal{X}$ represents the historical interactions between users and items. The predictive scoring function, which is determined by the inner product of the user and item embeddings, can be defined as:

$$\hat{y}_{ui} = \mathbf{e}_u^T \cdot \mathbf{e}_i \tag{6}$$

Assuming normalization, the range of $\hat{y}_{ui}$ is between 0 and 1, representing the predicted likelihood of user u interacting with item i.

3 Methodology

In this section, we provide a comprehensive overview of our XRec, which is specifically designed to generate explanations for user-item interactions. The goal of our model is to uncover the underlying reasons behind these interactions and shed light on the factors influencing user behavior. By unifying graph collaborative filtering and large language models, our XRec aims to provide insightful explanations which help users understand why certain interactions occur and enhance the transparency of the recommendation process.

3.1 Collaborative Relation Tokenizer

To efficiently capture the collaborative relationships between a large number of users and items, and to reflect their interaction patterns, traditional natural language processing approaches often fall short. Instead, representations provide a powerful alternative. Graph Neural Networks (GNNs) excel in capturing high-order dependencies among users and items by effectively modeling the intricate connections and interactions within a network.

Unlike conventional methods, GNNs can learn from the structure of the graph itself, allowing them to identify not just direct relationships, but also indirect ones that contribute to a user's preferences. This ability to aggregate information from a user's neighborhood in the graph helps in uncovering hidden patterns and correlations that would otherwise remain unnoticed. In our XRec, we harness the capabilities of graph neural networks as the tokenizer to encode the high-order collaborative relational information into a latent embedding space, enabling effective modeling of complex user preference.

Graph-based Message Passing. The collaborative graph tokenizer in our approach utilizes message passing mechanisms to propagate and aggregate information across the user-item interaction graph, facilitating the learning of representations for user and item nodes. In our framework, we employ LightGCN (He et al., 2020) as the backbone for effective collaborative information aggregation.

$$\begin{aligned} \mathbf{e}_u^{(l+1)} &= \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(l)}, \\ \mathbf{e}_i^{(l+1)} &= \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(l)} \end{aligned} \tag{7}$$

The user and item final embeddings are computed by averaging all layer embeddings.

$$\mathbf{e}_u = \sum_{k=0}^{K} \frac{1}{K+1} \mathbf{e}_u^{(k)}, \quad \mathbf{e}_i = \sum_{k=0}^{K} \frac{1}{K+1} \mathbf{e}_i^{(k)} \tag{8}$$

Tokenizer Optimization with CF signals. To optimize our collaborative graph tokenizer using implicit feedback signals from user interactions, we utilize the Bayesian Personalized Ranking (BPR) loss as a supervision signal to guide the generation of user and item embeddings, which is defined as:

$$L_{BPR} = -\sum_{u=1}^{m} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{u,i} - \hat{y}_{u,j}) \tag{9}$$

$\hat{y}_{u,i}$ denotes the prediction score (inner product) between user u and item i, and σ denotes the sigmoid function. Additionally, we include a regularization loss to maintain the norm of the embeddings:

$$L_{reg} = \lambda(\|\mathbf{e}_u^{(0)}\|^2 + \|\mathbf{e}_i^{(0)}\|^2) \tag{10}$$

By combining these terms, our joint optimization loss function is formulated as:

$$L = L_{BPR} + L_{reg} \tag{11}$$

3.2 Collaborative Instruction Tuning for Large Language Models (LLMs)

To enable LLMs to understand collaborative information among users and items, we introduce

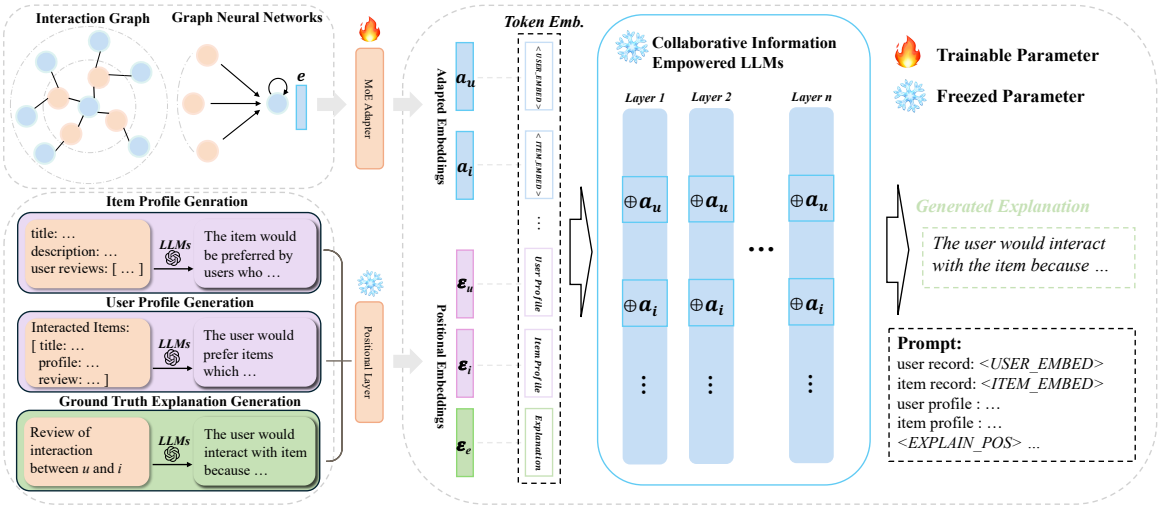

Figure 1: The overall architecture of our XRec. (i). **Collaborative Relation Tokenizer**: Transforms complex user-item relationships into latent embeddings using GNNs; (ii) **Collaborative Information Adapter**: A lightweight adapter that integrates collaborative signals into LLMs. (iii) **Unifying CF with LLM**: Integrates collaborative filtering insights directly into large language models, enabling them to generate insightful explanations.

a collaborative instruction tuning paradigm. This approach aligns behavior-level information with language-level semantics, thereby incorporating user preferences into the knowledge within LLMs.

3.2.1 Collaborative Information Adapter

Given the potentially divergent semantic representation spaces between the behavior-level collaborative information and the textual semantics associated with users and items, our XRec is equipped with a lightweight yet effective adapter. This adapter serves to align these different modalities, enabling our model to effectively leverage both the collaborative signals and the textual semantics.

To bridge the semantic gap between the input of large language models (LLMs) and our behavior-aware collaborative relation tokens, and to enhance the model's generalization capabilities, we apply a Mixture of Experts (MoE) approach (Hou et al., 2022) for embedding space adaptation. In this Mixture of Experts architecture, each expert is represented by a linear layer that captures different semantic dimensions, and these experts are then integrated using a learnable gating router mechanism. This allows the model to adaptively combine the different semantic representations encoded by the various experts, effectively bridging the gap between the behavior-aware collaborative relation tokens and textual language tokens.

3.2.2 Unifying CF with LLM

With the newly adapted embeddings, we are now ready to infuse collaborative information into LLMs. We introduce special tokens to reserve space in the input prompt, and after transforming the prompt into token embeddings, we inject the adapted embeddings into these reserved positions.

However, a challenge arises as each node embedding is represented by only a single token in the input prompt. As the input length increases, the attention weight allocated to each embedding token inevitably diminishes, leading to a potential loss of collaborative information. To address this dilution of influence, we take inspiration from (Qin et al., 2023) and extend the injection of adapted embeddings beyond the initial input prompt. Specifically, we incorporate them into every layer of the LLM at reserved positions. To facilitate this, we modify the key, query, and value projection functions of every layer within the LLMs as follows:

$$f_{\{q,k,v\}}(\mathbf{x}_i) \quad += \quad \mathbf{W}_{\{q,k,v\}} \cdot \mathbf{a}_i \qquad (12)$$

Let's denote the projection matrices for the query, key, and value as $\mathbf{W}_{q,k,v}$, and $\mathbf{a}_i$ as the adapted embedding. Our approach ensures that the large language models (LLMs) continuously access and integrate the collaborative information throughout their entire structure, not just at the input stage. By injecting the graph-based knowledge into all layers of the LLMs, we not only maintain a robust representation of the collaborative context, but also enable more effective gradient flow directly back to the Mixture of Experts (MoE) module. This innovative integration of language modeling and graph representation learning allows our model to leverage the deep contextual insights provided by the LLMs, while benefiting from the structural pat-

System Instruction:
Explain why the user would interact with the item within 50 words.
Input Prompt:
user record: <*USER_EMBED*> item record: <*ITEM_EMBED*>
item name: … user profile: … item profile: … <*EXPLAIN_POS*> …
Output:
The user would interact with the item because …

Figure 2: A depiction of model prompt instruction.

terns recognized by the Graph Neural Networks.

Structured Prompt Embedding. We employ a structured prompt, illustrated in Figure 2, which integrates various data elements. This process involves tokenizing the prompt and converting it into an embedding space representation. To ensure the special tokens within the prompt are recognized as unique entities, we incorporate them into the tokenizer of the LLM. These specialized tokens are then replaced by their corresponding adapted embeddings in the transformed token embeddings.

Specifically, we define the input prompt as $\mathcal{P} = [p_1, \ldots, p_u, \ldots, p_i, \ldots, p_e, \ldots, p_l]$, where each element p represents one input token. The tokens p_u, p_i, p_e denote <*USER_EMBED*>, <*ITEM_EMBED*>, <*EXPLAIN_POS*> respectively. After processing through the positional embedding layer, we denote the output as $\mathcal{E} = [\epsilon_1, \ldots, \epsilon_u, \ldots, \epsilon_i, \ldots, \epsilon_e, \ldots, \epsilon_l]$, where each ϵ is the token embedding of its corresponding token. Subsequently, ϵ_u and ϵ_i are replaced by the adapted embeddings a_u and a_i, to form the final embedding layer $\mathcal{E}' = [\epsilon_1, \ldots, a_u, \ldots, a_i, \ldots, \epsilon_e, \ldots, \epsilon_l]$, which is further used as input of LLMs.

To improve the ability of large language models (LLMs) to generate contextually and syntactically coherent explanations, we aim to minimize the loss between the predicted probabilities of next tokens and the actual next tokens in the sequences. We utilize the negative log-likelihood (NLL) as our training loss, calculated as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C_i} y_{ic} \cdot \log(\hat{y}_{ic}) \qquad (13)$$

Here, N is the number of explanations, C_i is the character count in each explanation, and $y_{i,c}$ and $\hat{y}_{i,c}$ represent the actual and predicted tokens, respectively. To minimize training complexity, we freeze all parameters within the LLMs, excluding any interactions with the GNN training procedure. The only trainable parameters are those within the Mixture-of-Experts (MoE) model.

3.2.3 Ground Truth Explanation Generation

Prior research has directly used user reviews as ground truth explanations for recommender systems (Li et al., 2021). However, these reviews tend to be subjective and may only implicitly convey the user's intentions or sentiments. To address this limitation and improve the quality of ground truth explanations, the application of Large Language Models (LLMs) has been proposed to distill explicit user intentions from their raw reviews.

$$\textit{explanation}(u, i) = \textit{LLMs}(\mathcal{P}, r_{u,i}) \qquad (14)$$

where $r_{u,i}$ is the review of item i given by user u. An example case is shown in the Appendix.

4 Evaluation

4.1 Experimental Settings

Datasets. To evaluate XRec, we utilize three prominent public datasets that offer distinct perspectives on user-item interactions: **Amazon** (Ni et al., 2019): This dataset aggregates the purchasing behaviors of users within Amazon's books category. It includes not only user ratings but also the textual reviews they provide after making a purchase. **Yelp**: This dataset captures the interactions between users and businesses, with a focus on the service industry. It contains both user ratings and reviews. **Google** (Li et al., 2022; Yan et al., 2023): Centered on user interactions recorded through Google Maps, this dataset incorporates the metadata of businesses as well as the feedback provided by users. Detailed statistics of these datasets can be found in Table 2.

Evaluation Metrics. When evaluating our XRec, we employ a suite of metrics designed to capture the semantic explainability and stability of the generated explanations. Traditional n-gram based metrics like BLEU and ROUGE prove unsuitable, as they fail to grasp the underlying semantic meaning. For instance, "the weather is cold" and "it's freezing" convey the same meaning, yet would score poorly due to a lack of n-gram overlap.

Additionally, some adopt feature-based metrics such as FMR (Feature Matching Ratio), FCR (Feature Coverage Ratio), and DIV (Feature Diversity) (Li et al., 2021, 2023), which assess feature overlap but present significant challenges when applied to sentence representation. For example, in a dataset from Amazon, the feature might be "movie", and the ground truth sentence is "This is a fantastic movie for kids and adults of all ages". A plausible paraphrase could be: "This movie only suits

Table 1: Overall Comparison in Terms of Explainability and Stability. The superscripts *P*, *R*, and *F1* indicate Precision, Recall, and F1-Score, respectively. The subscript std denotes the standard deviation of each score. The best performances are highlighted in bold, and the second-best are underlined.

Metrics	Explainability ↑							Stability ↓					
	GPTScore	BERTScore^{P}	BERTScore^{R}	BERTScore^{F1}	BARTScore	BLEURT	USR	GPT_{std}	BERT^{P}_{std}	BERT^{R}_{std}	BERT^{F1}_{std}	BART_{std}	BLEURT_{std}
Amazon-books													
Att2Seq	76.08	0.3746	0.3624	0.3687	-3.9440	-0.3302	0.7757	12.56	0.1691	0.1051	0.1275	0.5080	0.299
NRT	75.63	0.3444	0.3440	0.3443	-3.9806	-0.4073	0.5413	12.82	0.1804	0.1035	0.1321	0.5101	0.3104
PETER	77.65	**0.4279**	0.3799	0.4043	-3.8968	-0.2937	0.8480	11.21	0.1334	0.1035	0.1098	0.5144	0.2667
PETER+	76.07	0.4119	0.3626	0.3876	-3.9647	-0.3293	0.4493	11.99	0.1576	0.1077	0.1245	0.5131	0.2805
PEPLER	78.77	0.3506	0.3569	0.3543	-3.9142	-0.2950	0.9563	11.38	0.1105	0.0935	0.0893	0.5064	0.2195
Ours (w/o profile)	81.77	0.4194	0.4004	0.4106	-3.8218	-0.1294	**1.0000**	**9.60**	**0.0819**	0.0955	**0.0786**	**0.4799**	0.1803
Ours	**82.57**	0.4193	**0.4038**	**0.4122**	**-3.8035**	**-0.1061**	**1.0000**	**9.60**	0.0836	**0.0920**	0.0800	0.4832	**0.1780**
Yelp													
Att2Seq	63.91	0.2099	0.2658	0.2379	-4.5316	-0.6707	0.7583	15.62	0.1583	0.1074	0.1147	0.5616	0.247
NRT	61.94	0.0795	0.2225	0.1495	-4.6142	-0.7913	0.2677	16.81	0.2293	0.1134	0.1581	**0.5612**	0.2728
PETER	67.00	0.2102	0.2983	0.2513	-4.4100	-0.5816	0.8750	15.57	0.3315	0.1298	0.2230	0.5800	0.3555
PETER+	67.98	0.2594	0.3097	0.2833	-4.3973	-0.5355	0.8637	13.80	0.2522	0.1174	0.1701	0.5665	0.3421
PEPLER	67.54	0.2920	0.3183	0.3052	-4.4563	-0.3354	0.9143	14.18	0.1476	**0.1044**	0.1050	0.5777	0.2524
Ours (w/o profile)	71.81	0.3879	0.3427	0.3657	-4.4035	-0.2486	**1.0000**	12.71	0.1087	0.1072	0.0919	0.5717	**0.2272**
Ours	**74.53**	**0.3946**	**0.3506**	**0.3730**	**-4.3911**	**-0.2287**	**1.0000**	**11.45**	**0.0969**	0.1048	**0.0852**	0.5770	0.2322
Google-reviews													
Att2Seq	61.31	0.3619	0.3653	0.3636	-4.2627	-0.4671	0.5070	17.47	0.1855	0.1247	0.1403	0.6663	0.3198
NRT	58.27	0.3509	0.3495	0.3496	-4.2915	-0.4838	0.2533	19.16	0.2176	0.1267	0.1571	0.6620	0.3118
PETER	65.16	0.3892	0.3905	0.3881	-4.1527	-0.3375	0.4757	17.00	0.2819	0.1356	0.2005	0.6701	0.3272
PETER+	66.74	0.4125	0.3975	0.4047	-4.1273	-0.3467	0.4997	15.23	0.1892	0.1244	0.1411	0.6515	0.3114
PEPLER	61.58	0.3373	0.3711	0.3546	-4.1744	-0.2892	0.8660	17.17	0.1134	**0.1161**	0.0999	0.6752	0.2484
Ours (w/o profile)	**69.71**	0.4427	**0.4187**	0.4310	**-4.1142**	**-0.2026**	**0.9997**	**14.09**	0.1180	0.1171	0.1034	**0.6465**	**0.2439**
Ours	69.12	**0.4546**	0.4069	**0.4311**	-4.1647	-0.2437	0.9993	14.24	**0.0972**	0.1163	**0.0938**	0.6591	0.2452

Table 2: Statistics of the experimental datasets.

Dataset	#Users	#Items	#Interactions
Amazon	15,349	15,247	360,839
Yelp	15,942	14,085	393,680
Google	22,582	16,557	411,840

adults", which mentions "movie" but changes the sentence's meaning, highlighting the limitations of these metrics. Furthermore, in our dataset, we focus on generating longer explanations (50-word sentences), making it even more difficult to represent a sentence with a single word. As a result, these metrics are not well-suited for our task.

Instead, we employ advanced metrics that incorporate semantic understanding: GPTScore (Wang et al., 2023) aligns with human judgment by comparing the semantic similarity between generated and ground truth explanations. BERTScore (Zhang et al., 2020) utilizes contextual embeddings from BERT to compute token-level cosine similarity. BARTScore (Yuan et al., 2021) leverages the BART model, conceptualizing evaluation as a text generation task that assigns scores based on the probability of regenerating reference texts. BLEURT (Sellam et al., 2020) employs a novel pre-training approach with synthetic data to enhance generalization. USR (Li et al., 2021) (Unique Sentence Ratio) measures the uniqueness of generated explanations by calculating the ratio of unique to total sentences.

To further assess quality stability, we analyze the standard deviations of these scores, with lower values indicating more consistent performance. The overall results are shown in Table 1.

Compared Methods. We compare our model's performance against the following baselines:

- **Att2Seq** (Dong et al., 2017): Utilizes an attention-based attribute-to-sequence model to generate reviews based on attribute information.
- **NRT** (Li et al., 2017): Predicts ratings and generates abstractive tips for recommendations using multi-task learning to optimize parameters.
- **PETER** (Li et al., 2021): PETER is a personalized transformer model designed for explainable recommendations. It maps user and item IDs to generated explanation text, creating a connection between the IDs and words. Additionally, it incorporates a straightforward yet effective learning objective that uses these IDs to predict the words in the target explanation.
- **PETER+** (Li et al., 2021): To address the absence of baseline methods that use sentence-level text inputs, we adapted PETER+, a variant of the original PETER model, which was initially designed only for word-level inputs, to accept sentence-level inputs in our new datasets.
- **PEPLER** (Li et al., 2023): PEPLER leverages pretrained transformer to generate explainable recommendations based on prompts that incorporate user and item ID vectors. To bridge the gap between these prompts and the pretrained model, the approach proposes sequential tuning and recommendation as regularization strategies. There are several variants of the PEPLER, and in

this case, the researchers have opted to use the continuous prompt learning version.

- **Ours (w/o profile)**: For a fair comparison across all the models, we remove the user and item profiles from the input to our model, as the other baselines do not have access to this information.

Implementation Details. When generating the graph embeddings, we configure the embedding dimension to 64 and use a batch size of 1024. To optimize the training process, we implement an early stopping mechanism based on Recall@20, allowing for up to 10 patience steps. In our Mixture of Experts (MoE) setup, we utilize 8 experts and incorporate a dropout rate of 0.2, with an additional noise factor of 0.01 at the gating router. The LLM we use is built upon the LLaMA2-7B architecture. Additionally, we employ the gpt-3.5-turbo model for generating datasets and for computing the GPTScore. For the explanations, we ensure they are no longer than 50 words for both the ground truth and the generated explanations.

4.2 Performance Comparison

To demonstrate the superiority of our model in explainability and stability, we conduct comparative analyses against several baseline methods across three datasets. The results are summarized in Table 1 and reveal several key findings:

- Our model demonstrates exceptional performance in both explainability and stability. Its stability guarantees consistent and reliable results across various datasets, while the enhanced explainability offers deeper insights into user preferences and behaviors. This integration of improved semantic alignment and robust performance makes our model not only more interpretable but also more dependable in a wide range of recommendation scenarios.

- Our model consistently outperforms the baselines. Even after removing user and item profiles, the Ours (w/o profile) variant still demonstrates significant superiority. This success can be attributed to two key factors: i) Our model effectively captures collaborative information, enhancing the representation of rich semantics from user interaction behaviors, going beyond just textual information. ii) The model achieves strong alignment between the behavior-level collaborative information and text-level semantics.

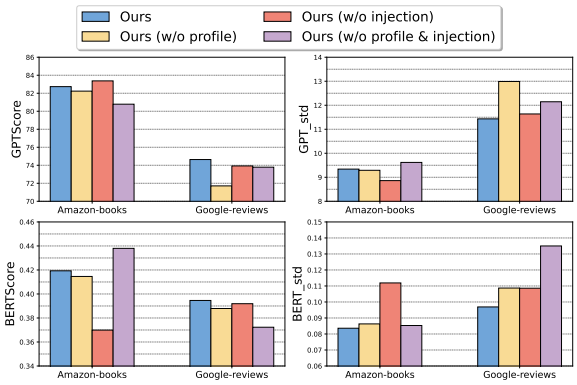

Figure 3: Ablation Study on Variant Models: Higher scores in GPTScore and BERTScore suggest improved explainability, while lower scores in GPT_std and BERT_std indicate enhanced stability.

- A standout feature of our model is its Unique Sentence Ratio (USR), which is nearly 1. This indicates that XRec generates truly unique explanations for each distinct user-item interaction. This remarkable level of uniqueness in the generated explanations represents a significant breakthrough. No previous work has achieved such a high degree of personalization in model outputs.

Our model enhances learning efficacy and boosts overall performance by strategically exploiting the synergistic strengths across domains.

4.3 Ablation Study

In this section, we conduct ablation studies to explore the impact of two pivotal components in our model: user/item profiles and the injection of collaborative information. We compare four model variants - i) our complete model with all features (**Ours**); ii) **Ours (w/o profile)** which omits user and item profiles; iii) **Ours (w/o injection)** which retains aligned embeddings in the prompts but does not inject them into the LLM layers, and iv) **Ours (w/o profile & injection)** lacking both profiles and embedding injection. To rigorously assess explainability and stability, we evaluate these variants using GPTScore and BERTScore on the Amazon-books and Google-reviews datasets, including their standard deviations, which sheds light on the critical role each of these elements plays in driving the model's performance and capabilities.

The results in Figure 3 show our complete model (Ours) outperforms other variants in explainability and stability, highlighting its superior capability. i) While **Ours (w/o profile)** declines only slightly, it exhibits a significant reduction in stability compared to Ours, underscoring the critical

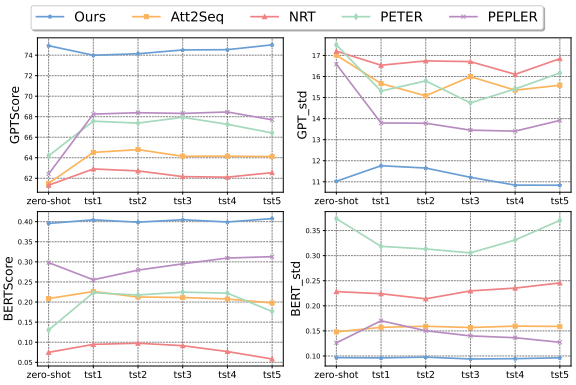

Figure 4: Experiments of different data sparsity.

importance of user/item profiles for explanation stability. ii) Similarly, **Ours (w/o injection)** reports much lower scores, emphasizing the value of incorporating neighborhood knowledge to improve performance. iii) Notably, **Ours (w/o profile & injection)** exhibits the lowest scores, confirming the synergistic combination of user/item profiles and knowledge injection is crucial for optimal performance. Although our model may not achieve the highest performance across all datasets, it demonstrates greater overall stability and consistently outperforms other variants. These findings underscore the complementary and synergistic contribution of these two critical components in driving the superior capabilities of our complete model.

4.4 Model Robustness against Data Sparsity

To evaluate our model's generalization capabilities, we conducted experiments across datasets with varying data sparsity. We segmented the testing data into five subsets (tst1 to tst5) based on the frequency of user appearances in the training data. This allowed us to systematically examine the model's effectiveness across a spectrum of user familiarity, from rare to frequent users. Additionally, we introduced a zero-shot testing dataset consisting solely of users not encountered during training, which tested the model's ability to address the cold-start problem. The evaluation results, summarized in Figure 1, highlight several key findings:

- The model demonstrates robust performance across all subsets, with noticeably better results as user frequency decreases. This trend suggests our model effectively leverages collaborative information, even with limited user interactions.

- In the zero-shot scenario, lacking any prior user data, our model not only outperforms baselines but also performs comparably to the other subsets, from tst1 to tst5. This capability is valuable for new user recommendations, highlighting the practical utility of our approach in real-world applications with incomplete user data.

These findings underscore the efficacy of our model in scenarios that traditionally challenge recommender systems, such as those involving new or infrequent users. The model's success in the zero-shot learning confirms its robust generalization capabilities and highlights its potential to mitigate the cold-start problem, where new users or items lack historical interaction data. By maintaining high levels of explainability and stability across diverse scenarios, the model proves its suitability for deployment in dynamic environments where user behaviors and item catalogs frequently change.

4.5 Case Study

In this case study, we explore a recommendation scenario on the Yelp platform to illustrate the practical application of our model in real-world settings. The goal is to show how the system provides transparent explanations, helping users understand the rationale behind specific recommendations.

In this example, a specific user and restaurant are considered. The user profile is described as:

This user is likely to enjoy casual American comfort food, barbecue with various meat options and tasty sauces, high-quality dining experiences with tasting menus, and authentic Italian food and beverages in cozy atmospheres.

The restaurant profile is summarized as

The restaurant, MD Oriental Market, is likely to attract fans of Asian cuisine, individuals looking for a variety of Asian products, and those seeking unique and ethnic food items would enjoy MD Oriental Market. Customers interested in a well-organized, spacious, and clean grocery store with a diverse selection of Asian ingredients and products would also appreciate this location.

Although the recommendation stems from the user's past interactions, the customer remains uncertain about its relevance and questions whether it is worth exploring. In this case, XRec steps in to offer a well-reasoned explanation for the recommendation, as illustrated below:

You would enjoy this business for its vast selection of Asian ingredients, including fresh produce, sauces, condiments, and spices, making it a go-to for authentic and diverse cooking options.

This explanation is grounded in the alignment between the user's preference for diverse, authentic food experiences and the restaurant's distinctive offerings. By delivering a clear, personalized rationale, XRec enhances the user's comprehension of the recommendation, addressing any uncertainties and elevating the overall user experience.

5 Related Work

5.1 Explainable Recommendation

Explainable recommender systems have attracted considerable attention due to their ability to enhance user satisfaction and provide transparency in recommendation processes. Early approaches primarily relied on extracting attributes from users and items' side information to generate explanations, employing techniques such as attention mechanisms (Dong et al., 2017) and recurrent neural networks (RNNs) (Li et al., 2017). In recent studies, researchers have further explored the application of advanced models like Transformer (Li et al., 2021) and GPT2 (Li et al., 2023) for offering explanations regarding user behaviors in recommender systems.

However, a predominant issue with most existing solutions for explainable recommendations is their heavy reliance on ID-based approaches. This dependency significantly restricts their generalization ability, especially when confronted with challenges such as data sparsity and zero-shot recommendation scenarios. Furthermore, the scarcity of explanatory data presents additional obstacles, as it poses challenges for existing methods to deliver explanations of high quality and comprehensiveness. Given the aforementioned obstacles, we propose a new LLM as an explainer for recommenders. This model not only uncovers the underlying reasons behind user-item interactions, but also demonstrates robust generalization capabilities, even in zero-shot recommendation scenarios.

5.2 GNNs for Recommendation

Graph Neural Networks (GNNs) have become a core part of improving collaborative filtering models. They offer an effective way to capture the complex high-level interactions in recommendation systems (Ying et al., 2018; Ren et al., 2023). These networks leverage the natural relational structure of data, enabling a sophisticated understanding of the intricate dependencies that define user-item interactions. Recommender systems like LightGCN (He et al., 2020) and Star-GCN (Zhang et al., 2019a) have set the standard, using iterative message passing to model and enhance collaborative relationships. To address the challenge of data sparsity, researchers have integrated self-supervised learning with the graph-based collaborative filtering approach. This introduces new methods to enrich the learning process and improve recommendation quality (Yang et al., 2023a,b; Yao et al., 2021).

Drawing inspiration from the aforementioned research endeavors that emphasize GNN-enhanced recommender systems, we have successfully developed our advanced language model, XRec. By incorporating GNN as the collaborative relation encoders, our model excels at capturing intricate user dependencies at higher orders. Through our collaborative instruction-tuning framework, we equip LLMs with the ability to recognize and leverage collaborative signals among users, effectively aligning behavior-level user preferences with the language semantic space. This enables our model to provide comprehensive textual explanations that correspond to user interaction behaviors.

6 Conclusion

This work presents a novel framework, XRec, that seamlessly integrates the graph-based collaborative filtering paradigm with the capabilities of Large Language Models (LLMs) to generate comprehensive and insightful explanations for recommendation outputs. By leveraging the inherent collaborative relationships encoded within the user-item interaction graph, XRec is able to effectively capture the high-order dependencies that underlie user preferences and item associations. XRec introduces a specialized collaborative information adaptor, which serves as the critical bridge for establishing a strong connection between the collaborative signals and the rich textual semantics encoded within the LLM. Through extensive experiments, the study's findings underscore the significant advantages of the XRec framework. Not only does it enhance the explainability of the recommendation process, but it also ensures robustness, particularly in challenging zero-shot scenarios where the framework demonstrates strong generalization capabilities across unseen users and items.

7 Limitation

While XRec demonstrates promising advancements in explainable recommender systems, it also exhibits limitations in terms of data modality diversity. Currently, our approach is constrained to textual and graph-based data, excluding visual inputs such as images and videos. These visual modalities can provide extensive contextual information. For instance, images and videos can reveal aesthetic preferences, cultural trends, and emotional responses, capturing subtle cues that textual data might miss. They also help identify visual patterns that correspond to user behavior and preferences, such as color schemes, styles, and settings, which are crucial for personalization. Recognizing these limitations, future work could explore the integration of multimodal data processing techniques. This approach may potentially enhance the system's predictive accuracy and improve its ability to personalize recommendation explanations, by incorporating advanced image and video analysis to better understand and respond to user preferences.

References

Hanxiong Chen, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. 2021. Neural collaborative reasoning. In *WWW*, pages 1516–1527.

Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*, pages 335–344.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *EACL*, pages 623–632.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pages 639–648.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *SIGIR*, pages 585–593.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. *arXiv preprint arXiv:2202.13469*.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation.

Lei Li, Yongfeng Zhang, and Li Chen. 2023. Personalized prompt learning for explainable recommendation. *Transactions on Information Systems (TOIS)*, 41(4):1–26.

Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *SIGIR*, pages 345–354.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*, pages 188–197.

Yijian Qin, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2023. Disentangled representation learning with large language models for text-attributed graphs. *arXiv preprint arXiv:2310.18152*.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *WWW*, pages 3464–3475.

Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled contrastive collaborative filtering. In *SIGIR*, pages 1137–1146.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *ACL*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. In *EMNLP*.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*, pages 165–174.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *arXiv preprint arXiv:2306.10933*.

Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. 2023. Automated self-supervised learning for recommendation. In *WWW*, pages 992–1002.

An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian McAuley. 2023. Personalized showcases: Generating multi-modal explanations for recommendations. In *SIGIR*, pages 2251–2255.

Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023a. Generative-contrastive graph learning for recommendation. In *SIGIR*, pages 1117–1126.

Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. 2023b. Knowledge graph self-supervised rationalization for recommendation. In *KDD*, pages 3046–3056.

Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *CIKM*, pages 4321–4330.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, pages 974–983.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *NeurIPS*, volume 34, pages 27263–27277.

Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019a. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. In *IJCAI*.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019b. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *ICLR*.

A Appendix

The supplementary materials accompanying this work offer a comprehensive and meticulous exploration of the methodologies and techniques utilized in crafting the ground truth explanations and developing the detailed user/item profiles.

A.1 Generating Ground Truth Explanations

Figure 5 provides an illustrative example of the ground truth explanation generation process as applied to the Yelp dataset. To maintain a consistent approach across diverse user-item interactions, the instructions given to the language models remain uniform, guiding them to extract the most relevant information that accurately reflects the user's underlying intentions. Notably, the prompt used in this process consists solely of the user's review text, deliberately omitting any additional data about the user or the item being reviewed. This strategic decision helps to minimize the influence of extraneous information, allowing the language model to focus exclusively on discerning and articulating the implicit intentions behind the user's interaction.

A.2 Item Profile Generation

Figure 6 illustrates the item profile generation process using the Yelp data. LLMs are fed metadata about the item, such as its name, location, and category, as well as user reviews. This approach allows

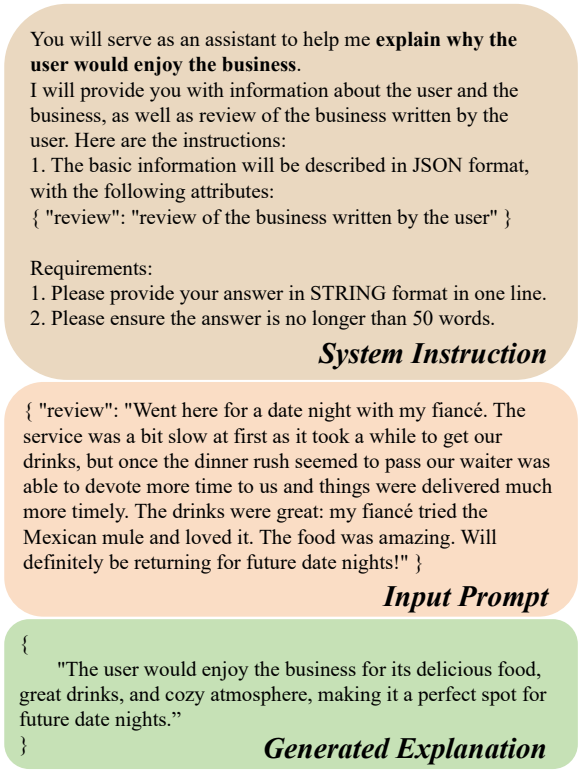

Figure 5: Case study on the generation of ground truth explanations for recommender systems on Yelp dataset.

the LLMs to gain a deeper understanding of the types of users who favor the business. By processing this multifaceted information, the LLMs generate comprehensive profiles that summarize the key characteristics of users inclined towards the item. For instance, the case study profiles capture insights into the preferences of users drawn to beer-related businesses. This granular understanding represents a significant advancement in modeling user-item interactions and preferences.

A.3 User Profile Generation

Figure 7 explores user profile generation on the Yelp dataset. Unlike items, users often lack extensive metadata, presenting a unique challenge. To deduce user preferences, the approach relies on Large Language Models (LLMs) to analyze the user's historical interactions. Crucially, the LLMs leverage item descriptions derived from previously generated profiles, refined to better characterize the items. This methodology, incorporating metadata like item titles and reviews, empowers the LLMs to identify the specific types of items a user prefers. By leveraging this multifaceted information, the system develops nuanced user profiles to enable personalized recommendations.

You will serve as an assistant to help me **summarize which types of users would enjoy a specific business.**
I will provide you with the basic information (name, city and category) of that business and also some feedback of users for it.
Here are the instructions:
1. The basic information will be described in JSON format, with the following attributes:
{ "name": "the name of the business", "city": "city where the company is located", "categories": "several tags describing the business" }
2. Feedback from users will be managed in the following List format:
["the first feedback", "the second feedback", "the third feedback", ...]
The information I will give you:
BASIC INFORMATION: a JSON string describing the basic information about the business.
USER FEEDBACK: a List object containing some feedback from users about the business.

Requirements:
1. Please provide your answer in JSON format, following this structure:
{ "summarization": "A summarization of what types of users would enjoy this business" }
2. Please ensure that the "summarization" is no longer than 50 words.

System Instruction

BASIC INFORMATION: { "name": "Restaurant Rebirth", "city": "New Orleans", "categories": "Cajun/Creole, Soup, Seafood, Restaurants, Steakhouses, Southern"}
USER FEEDBACK:
["Can't wait to get back to this place once restaurants reopen. Amazing casual but boujee ambiance on their breezy outdoor terrace ...",
"We ordered in for Sunday brunch this past weekend and for the most part, it hit the spot. We tried …",
"Absolutely zero shame in this ThankyouThursday post. This place is admittedly only a few blocks from my apartment, so convenience …"]

Input Prompt

{
"summarization": "Craft beer enthusiasts, casual beer drinkers, and those looking for a lively brewery experience with events, tastings, and a variety of beers to try would enjoy Sun King Brewery in Indianapolis."
}

Generated Item Profile

Figure 6: Case study of item profile generation on Yelp dataset.

You will serve as an assistant to help me **determine which types of business a specific user is likely to enjoy.**
I will provide you with information about businesses that the user has interacted, as well as his or her reviews of those businesses.
Here are the instructions:
1. Each interacted business will be described in DICTIONARY format, with the following attributes:
{ "title": "the name of the business", "description": "a description of what types of users will like this business", "review": "the user's review on the business" }
The information I will give you:
PURCHASED BUSINESSES: a list of DICTIONARIES describing the businesses that the user has interacted.

Requirements:
1. Please provide your answer in JSON format, following this structure:
{ "summarization": "A summarization of what types of business this user is likely to enjoy"}
2. Please ensure that the "summarization" is no longer than 50 words.

System Instruction

PURCHASED BUSINESSES:
[
{ "title": "Hub Coffee Roasters",
"description": "Craft beer enthusiasts, casual beer drinkers, and those looking for a lively brewery experience with … ",
"review": "Great little coffee shop. Love the back room. There's so much light there and … "},
{ "title": "1, 2, Tea",
"description": "1, 2, Tea in Reno is recommended for those who enjoy simple, reliable bubble tea with friendly service and … ",
"review": "I go here at least once a week if not more often. They have amazing drinks! Having lived in ... " },
{ "title": "Chicago's Pizza With A Twist - Reno, NV",
"description": "Those who enjoy unique pizza flavors like Indian-inspired toppings, fresh ingredients, and friendly service would love … ",
"review": "Indian food+pizza!!! It was love at first sight. And what's even more amazing is they have a gluten free crust … "}
]

Input Prompt

{
"summarization": "This user is likely to enjoy businesses with cozy atmospheres, quality beverages, friendly service, unique experiences like tea tastings, and activities that are both fun and interactive."}
}

Generated User Profile

Figure 7: Case study of user profile generation on Yelp dataset.