PrePPER: A Preference Pattern-based Profiling Framework for Explainable Recommendation

Taisuke Usumi, Akiko Masaki, Sanae Muramatsu, Akira Sakamoto, Takeharu Eda
Software Innovation Center, NTT Corporation, Japan
{taisuke.usumi, akiko.masaki, sanae.muramatsu, akira.sakamoto, takeharu.eda}@ntt.com

Abstract

Large Language Models (LLMs) have demonstrated remarkable performance across diverse tasks, drawing increasing attention to their application in recommendation systems. In particular, recommendation systems using natural language-based user profiles have attracted attention for improving transparency and scrutability. However, existing methods fail to fully leverage the recommendation capabilities of LLMs due to the unspecified importance of user preferences within user profiles and unmatched preference types between user profiles and item profiles. To address these challenges, we propose PrePPER, a novel preference pattern-based profiling framework designed to explicitly capture the importance of user preferences and enhance the alignment between user profiles and item profiles. PrePPER enables the extraction of users' preference patterns, which denote characteristic tendencies in user preferences, and the determination of their importance by clustering users' preferences. Specifically, we first extract users' preferences from their reviews and perform clustering on the extracted preferences. Based on the clustered preferences, we then infer users' preference patterns along with their relative importance, and construct user and item profiles using this information. Our proposed profiles incorporate the importance of user preferences and enhance the relatedness between user and item profiles, thereby improving the recommendation performance of existing recommender systems.

Keywords: Recommender Systems, Large Language Models, Preference Pattern-based Profiles

1. Introduction

Large Language Models (LLMs) have exhibited remarkable capabilities across a wide range of tasks, including natural language processing (Touvron et al., 2023; Achiam et al., 2023) and information retrieval (Liu et al., 2024; Zhu et al., 2023). Recently, their application in recommender systems has attracted significant research attention (Wu et al., 2024). However, many existing LLM-based recommender systems are limited to generating rating scores, item names, or simple "yes"/"no" answers as recommendations (Kang et al., 2023; Ji et al., 2024; Bao et al., 2023). Such outputs provide little insight into the underlying reasoning, leading to limited interpretability and transparency in recommendations.

Motivated by these challenges, researchers have explored approaches to improving interpretability by leveraging the strong generative capabilities of LLMs to provide not only rating scores or item names but also explanations for those recommendations (Li et al., 2023; Geng et al., 2022; Kim et al., 2025). However, most of these methods rely on user and item identifiers (IDs) as input, making it difficult to flexibly adapt to changes in user preferences.

To address these limitations, recent studies focus on the use of language-based user profiles, where user preferences are explicitly described in text (Ramos et al., 2024; Kim et al., 2025). Using user profiles, these methods not only improve inter-

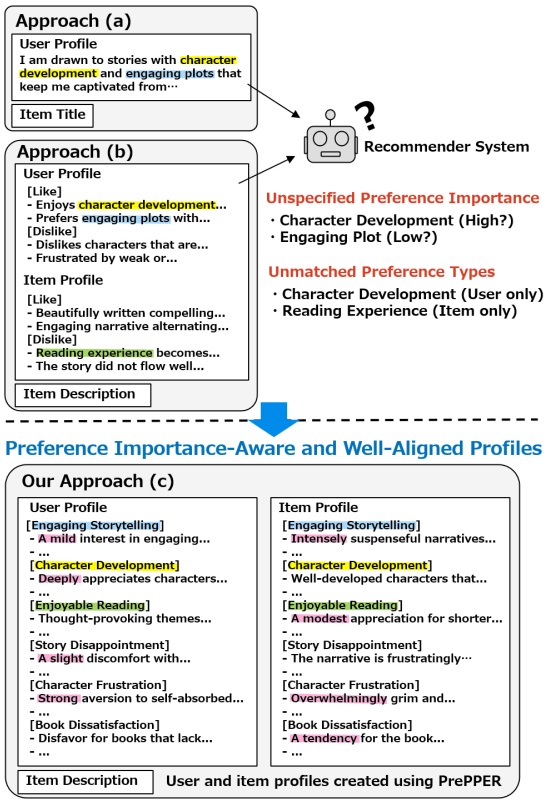

Figure 1: Comparison between existing profile-based recommender systems (a, b) and our proposed profiles (c). Our proposed user and item profiles, structured based on preference patterns, explicitly represent the importance of preferences and enhance the relatedness between profiles.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 705–715
11–16 May 2026.

pretability but also allow users to modify their profiles when their preferences change. UPR (Ramos et al., 2024) shows that a profile-based recommender system enables recommendations to be updated according to users' current preferences simply by modifying the user profile, without additional fine-tuning of the model. Furthermore, EXP3RT (Kim et al., 2025) constructed user and item profiles based on key preferences extracted from user reviews. By combining user profiles, item profiles, and item descriptions, they achieved highly accurate recommendation performance.

Recommender systems that utilize user profiles face several challenges, as shown in Figure 1(a) and (b), which correspond to the approaches UPR and EXP3RT, respectively. First, unspecified preference importance in user profiles prevents recommender systems from accurately capturing user preferences. Typically, user profiles contain multiple types of preferences related to items (e.g., preferences for characters or storylines in the book domain). However, the relative importance of each preference is not explicitly represented in the user profile. Consequently, recommender systems face difficulty in determining which preferences should be prioritized for recommendations. Second, unmatched preference types between user profiles and item information (e.g., item profiles) hinder accurate recommendations. When using user profiles, recommender systems compare user profiles with item information to determine whether the user is likely to prefer a given item. However, since preferences involve multiple types (e.g., characters or storylines), the preference types represented in the user profile and the item information are not always aligned. As shown in Figure 1(b), the user profile contains "character development", which is not included in the item profile, while the item profile mentions "reading experience", which is absent from the user profile. These unmatched preference types between user profiles and item information result in reduced recommendation accuracy.

To address these challenges, we propose **PrePPER**, **Pre**ference **P**attern-based **P**rofiling framework for **E**xplainable **R**ecommendation, which is a framework for constructing profiles that clearly indicate preference importance and increase the alignment between user and item profiles. As shown in Figure 1(c), our proposed profiles explicitly represent the importance of user preferences and are structured using preference patterns, which improves the relatedness between user and item profiles. The PrePPER framework consists of four distinct steps: (1) *preference extraction*, the first step which extracts user preferences regarding items from their reviews, (2) *preference clustering*, the second step that clusters positive and negative user preferences separately, (3) *preference pattern extraction*, the third step that extracts preference patterns that represent the characteristics of each cluster, (4) *profile construction*, the final step which constructs user and item profiles by aggregating user preferences based on the preference patterns. During the profile construction process, the importance of each preference pattern is determined for every user and item, which is then integrated into the profiles.

Our experiments demonstrate that incorporating our proposed profiles into existing recommender systems enhances their performance on recommendation tasks.

The key contributions of this work are summarized as follows:

- We propose a novel profile structure based on preference patterns to enhance the alignment between user and item profiles.
- We introduce a method to determine the importance of each preference pattern for every user and item, and incorporate this information into the profiles.
- The proposed profile can be easily integrated into existing recommender systems that utilize user profiles, enabling an improvement in their performance.

2. Related Work

2.1. LLM-based Recommender Systems

With the advancement of LLMs, research on LLM-based recommender systems has been growing rapidly in recent years (Wu et al., 2024). These studies are broadly categorized into two groups, depending on whether model parameters are tuned or not.

In non-tuning-based approaches, several studies have employed in-context learning to perform recommendation tasks by prompting LLMs (Dai et al., 2023; Liu et al., 2023; Gao et al., 2023; Liang et al., 2025). For rating prediction tasks, Dai et al. (2023) predicted ratings using users' past interaction histories, while Liu et al. (2023) proposed both a zero-shot method that uses only item titles and categories, and a few-shot method that leverages users' past item ratings in addition to item titles. These prompt-based approaches have the advantage of eliminating the need for parameter tuning, and thus can be applied without the need for large domain-specific datasets. However, to further improve recommendation performance within specific domains, fine-tuning the model using domain-specific datasets is crucial.

In contrast, tuning-based approaches focus on fine-tuning LLMs using domain-specific datasets

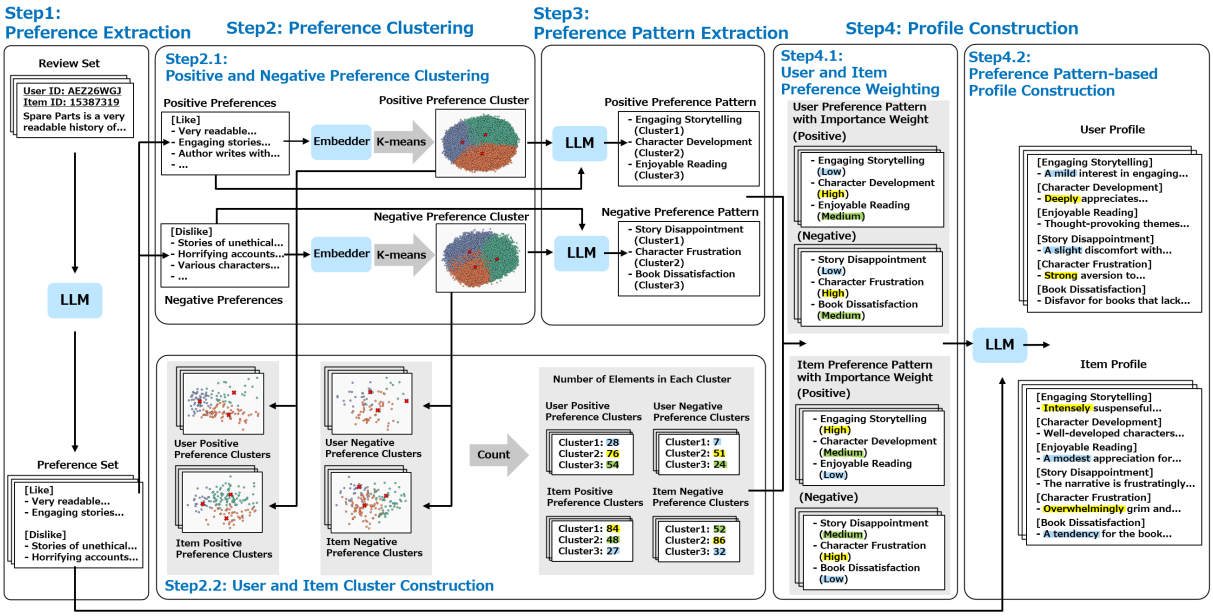

Figure 2: Illustration of our PrePPER framework. PrePPER framework consists of four steps: (1) Extracting users' preferences from their reviews, (2) clustering the preferences separately for positive and negative sets, (3) infering the preference patterns represented by each cluster, (4) constructing user and item profiles by aggregating preferences based on preference patterns.

(Geng et al., 2022; Bao et al., 2023; Ramos et al., 2024; Kim et al., 2025). In the field of recommender systems, research increasingly focuses on enhancing interpretability by generating explanations that indicate not only the recommendation outcomes but also the reasons why particular items are recommended (Geng et al., 2022; Li et al., 2023; Kim et al., 2025). For instance, Li et al. (2023) proposed a method that generates natural language explanations for recommendations using user and item ID vectors, while Geng et al. (2022) generated explanations by leveraging user and item IDs along with feature words. However, these methods rely on user and item IDs as input, which limits their ability to flexibly adapt to changes in user preferences.

2.2. LLM-based Natural Language Profile Methods

LLM-based natural language profile methods represent an approach that enhances the interpretability of recommender systems using user profiles representing preferences for items as input to LLMs (Ramos et al., 2024; Kim et al., 2025). UPR (Ramos et al., 2024) extracted feature words from user reviews that best reflect each user's preferences, and constructed user profiles using reviews that include these feature words. They further demonstrated that by simply modifying user profiles, recommendations can be adapted to changes in user preferences without the need for fine-tuning. Similarly, EXP3RT (Kim et al., 2025) obtained users' positive and negative preferences toward items from their reviews and aggregated this information for each user and item to construct user and item profiles.

However, existing studies construct user and item profiles without considering the multiple types of user preferences toward items. As a result, misalignments occur between user and item profiles, potentially leading to degradation of recommendation performance. Moreover, since the importance of user preferences is not explicitly indicated in the user profiles, it is unclear which preferences should be prioritized for recommendations. To address these issues, our work focuses on constructing structured profiles that incorporate both users' multifaceted preferences and the relative importance of each preference, thereby improving recommendation performance.

3. Methodology

In this section, we introduce PrePPER, a novel framework for constructing structured user and item profiles based on preference patterns, explicitly capturing the importance of user preferences and improving the alignment between user and item profiles. We aim to leverage these profiles within existing recommender systems to enhance recommendation performance.

The overview of our PrePPER framework is depicted in Figure 2. Our PrePPER framework consists of four main steps: (1) *preference extraction*, (2) *preference clustering*, (3) *preference pattern extraction*, (4) *profile construction*. Specifically, we

first extract users' preferences from their reviews. Next, we split the extracted preferences into positive and negative sets and cluster each set separately. From the clustered preferences, we extract elements for each user and item, and count the number of elements within each cluster. Then, we extract representative preference patterns from positive and negative clusters. Finally, we determine the importance of preference patterns for every user and item based on the number of elements in each cluster, and construct user and item profiles using these weighted preference patterns. We describe the details of each step in the following subsections.

3.1. Preference Extraction

In the first step, as shown in Step 1 of Figure 2,we extract users' preferences for items from their reviews. Since raw reviews often contain information that does not explicitly indicate user preferences, LLMs are impeded from accurately capturing user preferences. Therefore, it is important to extract users' preferences from their reviews that are useful for recommendation tasks.

In this study, we follow the EXP3RT approach to extract users' preferences from their reviews. This approach represents the extracted user preferences in "Like" and "Dislike" format.

3.2. Preference Clustering

In the second step, as illustrated in Step 2 of Figure 2, we prepare the clusters which will be used in the following steps to infer preference patterns and their relative importance. Specifically, we first cluster the users' preferences extracted in Section 3.1 separately for "Like" and "Dislike" (Step 2.1 in Figure 2). We then construct clusters for each user and each item by extracting the corresponding elements from clustered users' preferences. Finally, we count the number of elements in each cluster to determine the importance of preference patterns for each user and item (Step 2.2 in Figure 2). The detailed procedures of Step 2.1 and Step 2.2 are described in the following subsubsections.

3.2.1. Positive and Negative Preference Clustering

To cluster users' preferences according to positive and negative sentiments, we first aggregate all user preferences extracted in Section 3.1 into "Like" and "Dislike" categories, creating positive and negative preference sets. We then convert the natural language preferences within each preference set into vector representations using text embedding models. Finally, we cluster the positive and negative preference sets to group users' preferences into more fine-grained preference types.

Note that clustering the "Like" and "Dislike" preferences together leads to an increased number of clusters due to differences in preference polarity (positive and negative). As a result, the clustering process becomes more complex. Therefore, we perform clustering separately for "Like" and "Dislike" preference sets, enabling the extraction of clearer preference patterns.

3.2.2. User and Item Cluster Construction

We construct clusters for each user and each item by retrieving the corresponding user and item elements from the clusters of all users' preferences using their IDs. We then count the number of elements assigned to each cluster to identify which preference patterns are more frequently represented for each user and each item.

3.3. Preference Pattern Extraction

In the third step, we extract preference patterns based on the clustering results of all users' preferences presented in Section 3.2. A preference pattern denotes a characteristic tendency in user preferences, such as a preference for engaging storytelling or character development. We identify these preference patterns by extracting the representative characteristics from each cluster of all users' preferences.

A straightforward approach is to input all preferences contained in each cluster into the LLM, which then infers the cluster's characteristics to identify preference patterns. However, this approach results in excessively long input contexts and includes preferences that do not explicitly represent the characteristics of the cluster.

Therefore, we input only the preferences that are close to the cluster centroid into the LLM, enabling efficient inference of representative preference patterns for each cluster. Furthermore, to clarify differences between preference patterns across clusters, we aggregate the preferences close to the cluster centroids from all positive preference clusters and input them together into the LLM, and do the same for the negative preference cluster. We then instruct the LLM not to assign the same preference pattern to multiple clusters unless they are truly identical, to extract distinct preference patterns for each cluster. The prompt used for this process is provided in Table 1(a). Specifically, for each cluster, we fill the corresponding placeholder in Table 1(a) (i.e., {sentences for cluster 1}, {sentences for cluster 2}, and {sentences for cluster 3}) with preference sentences close to the cluster centroid. For example, such sentences include "*Engaging stories about scientists and inventors*".

Prompts	
(a) Preference Pattern Extraction	Given three separate clusters of short sentences describing what users like about books and each cluster reflects a different pattern of book preferences. Analyze each cluster independently and identify the overarching preference pattern that best represents the user's book tastes. DO NOT assign the same preference pattern to multiple clusters unless they are truly identical. Output only the preference pattern as a short label. Sentences for Cluster 1: {sentences for cluster 1} Sentences for Cluster 2: {sentences for cluster 2} Sentences for Cluster 3: {sentences for cluster 3}
(b) User Profile Construction	Given a book preference pattern and a set of sentences describing what a user likes within that preference pattern. Analyze the sentences and summarize up to two core positive preferences in bullet points. {phrase to strengthen or soften preferences} Avoid using the phrase "The user likes...". Keep each bullet point short and concise. Preference Pattern: {preference pattern} Sentences: {sentences} Output Format: - Summarize the user's core positive "preference" in bullet points.
(c) Item Profile Construction	Given a book preference pattern and a set of sentences describing what users like within that preference pattern. Analyze the sentences and summarize up to two core positive preferences in bullet points. {phrase to strengthen or soften preferences} Avoid using the phrase "The user likes...". Keep each bullet point short and concise. Preference Pattern: {preference pattern} Sentences: {sentences} Output Format: - Summarize the core "preference" users like about the item in bullet points.

Table 1: Prompts used for positive preference pattern extraction and profile construction using positive preference patterns on the Amazon-Book. For negative preferences, the prompts are modified by replacing "like" with "dislike" and changing "positive" to "negative".

3.4. Profile Construction

In the final step, we first assign importance weights to the preference patterns extracted in Section 3.3 based on the number of elements in each user and item cluster obtained in Section 3.2 (Step 4.1 in Figure 2). We then construct user and item profiles based on the weighted preference patterns (Step 4.2 in Figure 2).

3.4.1. User and Item Preference Weighting

We first determine the importance of the preference patterns extracted in Section 3.3 for each user and item. To achieve this, we hypothesize that preference patterns that are important to a user tend to be mentioned more frequently in the user reviews. For example, users who place greater importance on character-related preferences tend to mention characters frequently in their reviews. Based on this hypothesis, we determine the importance of each preference pattern for users and items based on the number of elements in each user and item cluster obtained in Section 3.2. Specifically, preference patterns with a larger number of elements are considered important, while those with fewer elements are considered less important.

3.4.2. Preference Pattern-based Profile Construction

We construct user and item profiles by aggregating and summarizing preferences for each preference pattern. Table 1(b) and (c) provide the prompts for constructing user and item profiles, respectively. In order to reflect the importance of preference patterns within the profiles, we replace the placeholder {phrase to strengthen or soften preferences} in Table 1(b) and (c) with "*Use stronger, more emphatic adjectives to express strong preferences*" for

highly important preference patterns, thereby emphasizing these preferences. Conversely, for less important preference patterns, we replace it with "*Use softer, milder adjectives to express weak preferences*" to attenuate the preferences. In these prompts, the placeholder {preference pattern} is replaced with the preference pattern extracted in Section 3.3, such as "*Engaging Storytelling*", and the placeholder {sentences} is filled with the preference sentences assigned to the corresponding cluster for each user or item.

3.5. Recommendation Task

The recommendation task is conducted by replacing the profiles used in existing recommender systems with our proposed profiles. Specifically, we employ EXP3RT and UPR in terms of their model architectures and profile-based input design.

Since EXP3RT utilizes user profiles, item profiles, and item descriptions for the recommendation task, we replace its original user and item profiles with our proposed profiles. While the original UPR study utilizes user profiles and item titles for the recommendation task, we incorporate item profiles in addition to user profiles and item titles to examine the impact of enhanced user and item profile relatedness on recommendation performance.

4. Experiments

4.1. Dataset

We conduct our experiments using two datasets, IMDB (Kim et al., 2024) and Amazon-Book (Hou et al., 2024a). IMDB is a movie-domain dataset with ratings ranging from 1 to 10, while Amazon-Book is a book-domain dataset with ratings ranging from 1 to 5. Both datasets include user reviews, ratings, and item metadata.

For preprocessing, users without ratings and users and items with fewer than five interactions are excluded. As a result, IMDB consists of 859 users, 1,155 items, and 19,124 interactions, whereas Amazon-Book contains 10,761 users, 10,084 items, and 118,705 interactions. The datasets are split into training, validation, and testing sets in an 8:1:1 ratio according to interaction timestamps (Zhang et al., 2025), which simulates real-world recommendation scenarios and avoids data leakage.

4.2. Baselines

We adopt the following rating prediction methods as baselines.

- **MF** (Koren et al., 2009): Matrix Factorization is a collaborative filtering (CF) method that learns latent user and item embeddings.
- **P5** (Geng et al., 2022): An LLM-based recommendation framework capable of performing multiple tasks such as rating prediction and explanation generation. T5-base (Raffel et al., 2020) is used as the backbone LLM.
- **LLMRec** (Liu et al., 2023): A prompting-based recommendation approach utilizing the LLM. Both zero-shot (ZS) and few-shot (FS) approaches are applied with GPT-4.

In addition, we compare our method with the following profile-based recommendation approaches.

- **UPR** (Ramos et al., 2024): A profile-based recommendation method that constructs user profiles from user reviews containing feature words that best reflect users' preferences, and performs recommendation tasks by fine-tuning GPT-2 using these profiles. Unlike the original UPR, we use GPT-4o-mini for both feature word extraction and profile generation.
- **EXP3RT** (Kim et al., 2025): An LLM-based recommendation method that distills knowledge from GPT-3.5 into LLaMA3-8B-Instruct for both profile construction and preference reasoning. It utilizes structured user and item profiles and employs QLoRA(Dettmers et al., 2023) to fine-tune LLaMA3-8B-Instruct.

4.3. Evaluation Metrics

To measure the rating prediction performance, we employ two metrics, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

4.4. Implementation Details

We employ GPT-4o-mini and LLaMA3-8B-Instruct as the LLMs used within our PrePPER framework. For transforming users' preferences into vector representations, we utilize two text embedding models, OpenAI text-embedding-3-small and Sentence-BERT (Reimers and Gurevych, 2019). To distinguish profiles constructed using open-source models from those constructed using closed-source models, we employ text-embedding-3-small as the text embedding model when constructing profiles with GPT-4o-mini, and Sentence-BERT when constructing profiles with LLaMA3-8B-Instruct. We adopt the K-means++ algorithm (Arthur and Vassilvitskii, 2007) for preference clustering and set the number of clusters to 3. For preference pattern extraction, 200 preference sentences closest to each cluster centroid are used when employing GPT-4o-mini, whereas 150 preference sentences are used with LLaMA3-8B-Instruct.

For the recommendation tasks using our proposed profiles with UPR model architecture, we

Methods	IMDB		Amazon-Book					
			Total		Warm-start		Cold-start	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MF	1.9530	1.4771	0.6683	0.4572	0.6663	0.4588	0.6769	0.4501
P5	2.5471	1.8604	0.8868	0.4756	0.8873	0.4810	0.8846	0.4508
LLMRec (ZS, GPT-4)	2.1586	1.6261	0.9986	0.8331	0.9941	0.8257	1.0187	0.8670
LLMRec (FS, GPT-4)	2.1410	1.6334	0.8386	0.5004	0.8370	0.5010	0.8460	0.4974
UPR (GPT-4o-mini)	1.9503	1.4696	0.6514	0.4256	0.6429	0.4208	0.6907	0.4459
EXP3RT	1.9122	1.4691	0.6115	0.3649	0.6196	0.3746	0.5760	0.3207
PrePPER (UPR, GPT-4o-mini)	1.9273	1.4357	0.6592	0.4191	0.6519	0.4165	0.6921	0.4311
PrePPER (EXP3RT, Llama3-8B)	**1.8915**	1.4678	0.6035	0.3484	0.6135	0.3601	**0.5575**	**0.2980**
PrePPER (EXP3RT, GPT-4o-mini)	1.9180	**1.4331**	**0.5888**	**0.3457**	**0.5952**	**0.3548**	0.5684	0.3116

Table 2: Rating prediction performance on IMDB and Amazon-Book. The boldface indicates the best result and the underlined indicates the second best.

Number of Clusters	IMDB		Amazon-Book	
	RMSE	MAE	RMSE	MAE
K=2	1.9557	1.4356	**0.5766**	0.3398
K=3	**1.9180**	**1.4331**	0.5888	0.3457
K=4	1.9847	1.4967	0.5912	**0.3347**
K=5	1.9773	1.4769	0.6051	0.3492
K=6	1.9652	1.4643	0.5988	0.3427

Table 3: Recommendation performance comparison for different numbers of clusters. The best and the second best values are in bold and underlined, respectively.

fine-tuned GPT-2 using the AdamW optimizer with a learning rate of 3e-4. The maximum epoch is set to 10, and early stopping is applied with a patience of 3. When utilizing EXP3RT model architecture, we fine-tuned Llama3-8B-Instruct with QLoRA using the AdamW optimizer with a learning rate of 2e-4. The QLoRA hyperparameters are set as follows: for IMDB, lora_r = 512, lora_alpha = 128, and lora_dropout = 0.1; for Amazon-Book, lora_r = 128, lora_alpha = 32, and lora_dropout = 0.1.

When applying our proposed profiles within the EXP3RT model architecture, we do not follow the original distillation procedure for profile construction. Instead, we utilize the profiles generated by GPT-4o-mini and LLaMA3-8B-Instruct to examine whether strong performance can be achieved without the computationally expensive distillation step.

5. Results and Analysis

5.1. Recommendation Performance

Table 2 shows the rating prediction results of our proposed method compared with the baseline models. We observe that our method achieves the best performance across both datasets. In particular, when integrating our proposed profiles into EXP3RT, the highest performance is achieved across all evaluation metrics. Notably, unlike original EXP3RT, which distills knowledge from a large teacher model such as GPT-3.5, our approach achieves higher accuracy using only LLaMA3-8B, without relying on large-model distillation. Moreover, when applying our proposed profiles to UPR, significant performance improvements are observed on IMDB, and comparable performance to the UPR is achieved on Amazon-Book. These results demonstrate that incorporating our proposed profiles into existing recommender systems can enhance their recommendation performance.

Furthermore, to evaluate the robustness of the recommender systems under various recommendation scenarios, we conduct experiments on testing set divided into warm-start and cold-start subsets. The warm-start subset consists of users who have interacted with items more than three times in the training set, while the cold-start subset includes the remaining users. As a result, the integration of our proposed profiles into EXP3RT achieves the best performance under both warm-start and cold-start scenarios.

Note that this study uses the same datasets as EXP3RT, and the results for MF, P5, and LLMRec are cited from the original EXP3RT study.

5.2. Impact of Number of Clusters on Recommendation Performance

To investigate the impact of the number of clusters on the recommendation task, we vary the number of clusters as $K \in \{2, 3, 4, 5, 6\}$, construct profiles accordingly, and evaluate their recommendation performance using the EXP3RT model architecture.

The results are presented in Table 3. We observe that the highest performance is achieved with 3 clusters for IMDB and 2 clusters for Amazon-Book. Furthermore, increasing the number of clusters

Methods	IMDB				Amazon-Book			
	Readability	Informativeness	Diversity	Relevance	Readability	Informativeness	Diversity	Relevance
UPR (GPT-4o-mini)	4.97	4.70	4.12	-	**4.99**	4.46	3.87	-
EXP3RT	4.96	4.61	4.59	4.28	4.91	4.33	4.18	4.56
PrePPER (Llama3-8B)	4.85	4.93	4.97	4.40	4.78	4.53	4.51	**4.82**
PrePPER (GPT-4o-mini)	**5.00**	**5.00**	**5.00**	**4.92**	4.98	**4.69**	**4.80**	4.74

Table 4: LLM-as-a-judge comparing UPR (GPT-4o-mini), EXP3RT, and PrePPER (Llama3-8B, GPT-4o-mini) on the IMDB and Amazon-Book.

Methods	IMDB				Amazon-Book			
	Readability	Informativeness	Diversity	Relevance	Readability	Informativeness	Diversity	Relevance
UPR (GPT-4o-mini)	2.70	3.13	2.98	-	3.50	3.42	3.18	-
EXP3RT	**3.98**	3.45	3.48	3.10	**4.30**	3.95	3.95	3.72
PrePPER (GPT-4o-mini)	3.92	**4.05**	**3.82**	**3.63**	4.15	**4.20**	**4.13**	**3.90**

Table 5: Human evaluation comparing UPR (GPT-4o-mini), EXP3RT, and PrePPER (GPT-4o-mini) based on the average scores.

Methods	IMDB				Amazon-Book			
	Readability	Informativeness	Diversity	Relevance	Readability	Informativeness	Diversity	Relevance
1st	PrePPER	PrePPER	PrePPER/EXP3RT	PrePPER	PrePPER	PrePPER	PrePPER	PrePPER
2nd	EXP3RT	EXP3RT	-	EXP3RT	EXP3RT	EXP3RT	EXP3RT	EXP3RT
3rd	URP	URP	URP	-	URP	URP	URP	-

Table 6: Human evaluation comparing UPR (GPT-4o-mini), EXP3RT, and PrePPER (GPT-4o-mini) based on the Borda count.

leads to performance degradation in both datasets. This can be attributed to the fact that as the number of clusters increases, the similar preferences become dispersed across multiple clusters, thereby making the characteristics of each cluster less distinct. Therefore, we set the number of clusters to 3 for our experiments.

5.3. Profile Quality

We conduct two types of evaluations for the generated profiles, LLM-as-a-Judge and human evaluation. In both evaluation methods, we assess the profiles using the following four criteria.

- **Readability**: How clear, well-structured, and easy to understand the profiles are.
- **Informativeness**: How informative and specific the profiles are.
- **Diversity**: How well the profiles capture diverse and multifaceted perspectives on preferences.
- **Relevance**: How well the preference aspects described in the user profile and the item profile correspond to each other (applicable only for EXP3RT and PrePPER).

All criteria are scored on a scale from 1 to 5, where 5 represents the highest score. As the baselines, we employ EXP3RT and UPR. Since UPR does not utilize item profiles, the relevance evaluation is conducted only for EXP3RT and PrePPER.

5.3.1. LLM-as-a-Judge

To evaluate the quality of the generated profiles, we conduct the LLM-as-a-judge evaluation on 200 samples from each dataset, consisting of 100 samples with high ratings and 100 samples with low ratings. Specifically, samples with ratings ≥ 4 in the Amazon-Book and ratings ≥ 7 in the IMDB are categorized as high-rated, while those with ratings ≤ 2 in Amazon-Book and ratings ≤ 4 in IMDB are categorized as low-rated. We employ GPT-4.1 as the LLM-as-a-judge for evaluating profile quality.

The results are presented in Table 4. We observe significant improvements in informativeness, diversity, and relevance on both the IMDB and Amazon-Book. Furthermore, substantial improvements are observed even when using Llama3-8B for profile construction. These results suggest that improved alignment between user and item profiles contribute to the enhanced recommendation performance.

5.3.2. Human Evaluation

We conduct a human evaluation using 20 samples from each dataset, consisting of 10 high-rated and 10 low-rated samples. Three human judges are asked to evaluate the samples for each dataset based on the given criteria.

The performance comparison is conducted using two evaluation methods, average scores and Borda count. The Borda count is employed to mitigate the influence of evaluators who provide extremely high or low scores. For each evaluator, we first sum the

scores for each criterion to determine the ranking of the profiles, and the final ranking is then obtained using the Borda count.

The results based on the average scores and Borda count are presented in Tables 5 and 6, respectively. Consistent with the LLM-as-a-judge findings, human evaluation also demonstrates substantial improvements in informativeness, diversity, and relevance on both the IMDB and Amazon-Book. For readability, our proposed profiles rank first according to the Borda count, indicating that a majority of evaluators find them more readable. The results indicate that PrePPER not only improves recommendation accuracy but also enhances explainability, as the generated user and item profiles provide informative and readable summaries of user preferences and item characteristics.

6. Conclusion

In this paper, we propose PrePPER, a novel framework for constructing structured user and item profiles based on preference patterns extracted from users' preferences. PrePPER enables the incorporation of preference importance within profiles, while also enhancing the alignment between user and item profiles. Experimental results demonstrate that our proposed profiles significantly improve performance of existing recommendation systems. Furthermore, the evaluation of profile quality demonstrates that improving the relatedness between user and item profiles contribute to enhanced recommendation performance. This study is the first to focus on both the importance of preferences within profiles and the relatedness between user and item profiles in recommender systems.

7. Limitation

Since PrePPER constructs profiles from user reviews, its recommendation performance highly depends on the quality of the reviews. Furthermore, as not all users write reviews, our method cannot be applied to users who do not provide any reviews. Therefore, incorporating additional sources of information such as past ratings or viewing histories to construct user profiles could be a promising direction for future work.

8. Ethics Statement

One critical challenge in LLM-based recommendation systems is popularity bias, in which popular items are more likely to be recommended (Hou et al., 2024b). Another significant challenge is hallucination, where the model generates information that appears plausible but is factually incorrect. Therefore, addressing these challenges is essential for developing more reliable recommender systems.

9. Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1007–1014, New York, NY, USA. Association for Computing Machinery.

Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1126–1132, New York, NY, USA. Association for Computing Machinery.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 299–315, New York, NY, USA. Association for Computing Machinery.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024a. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024b. Large language models are zero-shot rankers for recommender systems. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II*, page 364–381, Berlin, Heidelberg. Springer-Verlag.

Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2024. Genrec: Large language model for generative recommendation. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part III*, page 494–502, Berlin, Heidelberg. Springer-Verlag.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474*.

Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. 2025. Review-driven personalized preference reasoning with large language models for recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 1697–1706, New York, NY, USA. Association for Computing Machinery.

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, Bangkok, Thailand. Association for Computational Linguistics.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Lei Li, Yongfeng Zhang, and Li Chen. 2023. Personalized prompt learning for explainable recommendation. *ACM Trans. Inf. Syst.*, 41(4).

Yueqing Liang, Liangwei Yang, Chen Wang, Xiongxiao Xu, Philip S. Yu, and Kai Shu. 2025. Taxonomy-guided zero-shot recommendations with LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1520–1530, Abu Dhabi, UAE. Association for Computational Linguistics.

Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.

Zheng Liu, Yujia Zhou, Yutao Zhu, Jianxun Lian, Chaozhuo Li, Zhicheng Dou, Defu Lian, and Jian-Yun Nie. 2024. Information retrieval meets large language models. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1586–1589, New York, NY, USA. Association for Computing Machinery.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Jerome Ramos, Hossein A. Rahmani, Xi Wang, Xiao Fu, and Aldo Lipani. 2024. Transparent and scrutable recommendations using natural language user profiles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13971–13984, Bangkok, Thailand. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5).

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2025. Collm:

Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2329–2340.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.