Review Texts as Data-Efficient Signals for LLM-based Personalization

Anonymous ACL submission

Abstract

Off-the-shelf large language models (LLMs) have been showing promising performance in personalization based on user preference. However, previous studies mainly discuss using numeric signals such as scores, which require many data points for satisfactory performance. Some systems based on fine-tuned LLMs have achieved reasonable performance by using review texts as additional information, but their use with off-the-shelf LLMs is underexplored. This work aims to clarify the effects of review texts on off-the-shelf LLM–based personalization from various perspectives. By comparing multiple prompt formats with different in-context information, we show that per-item review texts can improve the user rating prediction performance by off-the-shelf LLMs across different datasets and models, even with a few data points. We also find that instructing LLMs to write expected reviews can improve the performance, while general prompt engineering techniques such as zero-shot chain-of-thought can result in a worse performance. These results open the possibility of LLM-based personalization systems with fewer required data points.

1 Introduction

Recent large language models (LLMs) have demonstrated remarkable capabilities across various tasks without task-specific fine-tuning. Personalization is one such application. By providing user preferences in textual form and applying prompt engineering techniques, previous studies enabled off-the-shelf LLMs to align with individual preferences in tasks such as preferred item prediction (Zhang, 2024), user rating prediction (Kang et al., 2023), and item reranking (Xu et al., 2024; Hou et al., 2024).

For those tasks, user preference signals are typically provided as simple numeric preference scores (Harper and Konstan, 2015) or binary flags (Wu et al., 2020). However, each of these signals only contains limited information. As a result, the LLM-based prediction system can suffer from the "cold-start problem" (Schein et al., 2002; Zhang et al., 2024), in which reliable predictions are unavailable until a user accumulates sufficient data points.

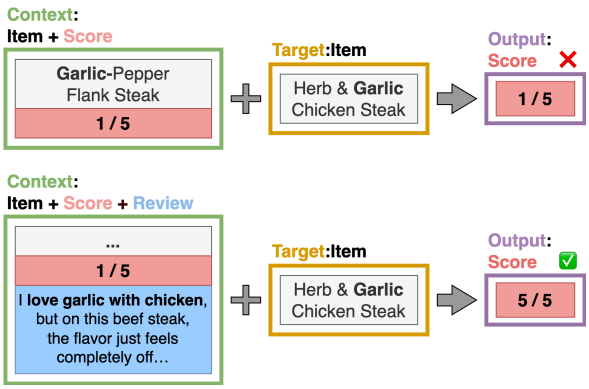


Figure 1: By leveraging the review text provided in the context, LLMs can more accurately infer a user's preference for the target item.

An alternative approach to mitigate this issue is to supplement the signals with richer data, such as user-generated review texts. Figure 1 shows that LLMs can improve their preference score predictions by leveraging review text in the input context. As shown in this example, review texts contain more concrete and detailed information about the user's preference than the numeric signals, so including review texts can enhance LLMs' performance on preference prediction even with a few data points. Writing a simple review can be less costly than sampling many items in domains such as movies or recipes. Therefore, using review texts can reduce the amount of required data points for LLM-based recommendation systems, making the system less burdensome to use.

Several user preference prediction datasets already provide review texts (Ni et al., 2019; Wang et al., 2024; Majumder et al., 2019). Prior work

has fine-tuned LLMs on one of those datasets and obtained reasonable preference prediction performance with only around three in-context examples (Wang et al., 2024). Therefore, the hypothesis that "review texts enable personalization with small examples" is verified in fine-tuning settings. However, the effect of review texts on off-the-shelf LLMs is under-explored.

In this paper, we investigate the effect of the user-written reviews on the preference alignment by off-the-shelf LLMs, specifically on user rating prediction tasks. We analyze the behavior of multiple LLMs on different datasets based on the following four research questions:

- *RQ1*: Do review texts contribute to the performance improvement of the rating prediction task by off-the-shelf LLMs?
- *RQ2*: How does the performance change on more difficult settings?
- *RQ3*: Can the prediction performance be further enhanced with prompt engineering techniques?
- *RQ4*: Do review texts have a better effect than preference described in other formats?

To answer *RQ1*, we compose different instruction prompts with and without review texts as the in-context information, and compare the user rating prediction performance with multiple datasets and LLMs. Then, for *RQ2*, we raise the difficulty of the problem by evaluating three independent settings with different in-context examples: fewer reviews, shorter reviews, and reviews written by non-target users. To address *RQ3*, we apply different prompt engineering strategies inspired by previous studies (Kojima et al., 2022; Xi et al., 2024; Lyu et al., 2024). Finally, for *RQ4*, we analyze the difference of the per-item review texts with "self-described preference" used in previous studies (Sanner et al., 2023; Eberhard et al., 2025) by converting the per-item reviews into the self-described preference format using LLMs.

Our key findings are the following:

- Across different datasets and off-the-shelf LLMs, per-item review texts consistently improve the performance on user rating prediction. Instructing the LLMs to write down the expected review text is also a promising method to improve the performance of review-based preference prediction.
- As long as the review texts are correctly paired with the corresponding integer scores, smaller amount of data is still effective.
- General prompt engineering strategies, such as zero-shot chain-of-thought (Kojima et al., 2022), do not straightforwardly improve the performance of the preference prediction task.
- Per-item review format has an advantage over the self-described preference format used by Sanner et al. (2023).

2 Related Work

2.1 Personalization with Off-the-Shelf LLMs

Many previous studies have analyzed the personalization performance by off-the-shelf LLMs, mainly with non-textual preference signals. Hou et al. (2024) used off-the-shelf LLMs for the personalized item ranking task based on the user's historical interaction with other items. Di Palma et al. (2023) analyze ChatGPT[1]'s performance on the top-N recommendation task based on historical interaction. Wu et al. (2024) show that providing the user's historical responses in the context improves off-the-shelf LLMs' performance on the LaMP (Salemi et al., 2024) dataset. Zhang (2024) proposes a method of instructing LLMs to summarize the user's past interactions in a specific manner to improve the performance of off-the-shelf LLMs on the multiple-choice preference prediction task. Xu et al. (2025) provide a large-scale performance analysis across different LLMs on item reranking tasks based on historical interactions.

Some work focuses on the preference described in textual forms. Eberhard et al. (2025) proposed a recommendation system based on free-form text user requests with off-the-shelf LLMs and basic prompt engineering techniques, such as few-shot or role-playing prompting. Sanner et al. (2023) collect self-described preferences of users to enhance the item reranking performance by LLMs. However, since these studies are limited to simpler tasks such as top-N recommendation or item reranking, whether the same method is applicable to more complex settings such as user rating prediction is unknown. Comparison with other forms of preference data such as per-item reviews is also not explored.

[1] https://openai.com/index/chatgpt/

2.2 Incorporation of Textual Preference Description in LLM-based Personalization

If not limited to the off-the-shelf LLMs, many LLM-based personalization systems use the per-item reviews as the preference signals. Most notable ones are from task-specific fine-tuning. Zhang et al. (2023) build an LLM-based recommendation system by instruction-tuning with various formats of prompts that describe user preferences.

Per-MPST (Wang et al., 2024) is a user rating prediction dataset with past user reviews as part of its inputs. The authors also propose PerSE as the framework for solving the problem with fine-tuned LLMs and achieving reasonable prediction performance with a few in-context examples. If a similar method also works only with off-the-shelf LLMs, we can save training cost for fine-tuning.

2.3 Prompt Engineering on Preference Prediction

Prompt engineering has been actively studied to enhance the LLMs' performance on various tasks. Chain-of-thought (CoT; Wei et al., 2022; Kojima et al., 2022) is one of the most notable ones, boosting LLMs' performance in multiple domains. However, a recent study (Sprague et al., 2024) suggests that CoT only works effectively on domains that require math or logic.

Zhang (2024) instructs LLMs to generate intermediate outputs from a specific viewpoint. The prompt used by Wang et al. (2024) has LLMs explicitly write down the expected review texts from the users. Those two can be considered as task-specific prompt engineering techniques.

Another line of work uses LLMs for data augmentation so that the downstream LLM-based systems can use more data. Knowledge Augmented Generation (KAR; Xi et al., 2024), LLM-Rec (Lyu et al., 2024), and UR4Rec (Zhang et al., 2025) generate intermediate texts with LLMs to increase the input data to the fine-tuned recommendation models. Sun et al. (2025) and Richardson et al. (2023) summarize the user-generated texts with them so that their retrieval-augmentation-based system can use the texts effectively. However, whether these techniques are effective for off-the-shelf LLMs to handle the review texts is still unknown.

3 Problem Formulation

We investigate the effect of review texts on the personalization performance by off-the-shelf LLMs. More specifically, we focus on the user rating prediction task. The task is formulated as follows.

Let the target LLM be $\mathcal{M}$. For each given target item description x_u and the user u, the goal of the task is for $\mathcal{M}$ to predict the preference score y_u that u assigns to x_u. y_u always takes an integer value between y_{min} and y_{max} inclusive, where y_{min} and y_{max} denote the minimum and maximum scores defined for each dataset, respectively.

For each prediction, $\mathcal{M}$ gets two additional parameters: p_u, a set of texts that contains u's personal preference information, such as u's past review history (user profile), and I, an instruction that specifies the input and output formats of the tasks. Based on those inputs, $\mathcal{M}$ gives an output o_u as

$$o_u = \mathcal{M}(I, x_u, p_u) \tag{1}$$

Note that o_u could contain additional texts other than the predicted score, depending on the instruction I. Therefore, the predicted score y'_u can be obtained by the instruction-specific extraction function f_I as

$$y'_u = f_I(o_u). \tag{2}$$

We collect a set of users $\mathcal{U}$ and prepare $\mathcal{D} = \{(x_u, p_u, y_u)\}_{u \in \mathcal{U}}$ as an evaluation dataset. The final performance is measured based on the comparison of $\{(y'_u, y_u)\}_{u \in \mathcal{U}}$. We control the format of p_u and I and see the effects on the performance.

4 Experimental Settings

4.1 Datasets

We use three datasets to evaluate the rating prediction performance of LLMs.

Per-MPST (Wang et al., 2024) (Movies) is a movie review dataset based on the IMDb[2] data. Each data point consists of the textual description of the movie plot, a user's review text, and a review score from 1 (lowest) to 10 (highest). The dataset provides five subsets based on the number of in-context examples k used for querying. We use the test split of the $k = 5$ version.

Different data splits are provided based on the number of in-context examples used for querying, and we use $k = 5$ test split for this experiment.

[2]https://www.imdb.com/

We also use Recipe[3] (Majumder et al., 2019) and the Book Category of Amazon Reviews'23[4] (Ni et al., 2019) (Books) as the target datasets. To analyze the difference between the target domains without being affected by other factors such as data formats, we postprocess those two datasets to align with the format of the Movies dataset. We concatenate multiple properties in the original datasets (name, description, and steps for Recipe, title, subtitle, and features for Books) to craft a single "item description text", filter out review texts with less than 200 characters, then randomly pick 1000 instances with $k = 5$ in-context examples respectively. See Section B.3 for detailed dataset statistics.

4.2 Models

We use five open-source LLMs to test the effectiveness of the review text data. Among the well-known open-source series widely used in recommendation systems, we choose the latest versions available at the time across different parameter sizes: Llama 3.1 8B, Llama 3.3 70B (Grattafiori et al., 2024), Gemma 3 12B, Gemma 3 27B (Gemma Team et al., 2025). We use instruction-tuned versions of those four models. We also use QwQ 32B (Qwen Team, 2024) to test the effect on reasoning models. See Section B.1 for detailed configurations.

4.3 Evaluation Methods

We report Spearman and Kendall-Tau correlation coefficients as the evaluation metrics of the agreement between the ground truth labels and the integer preference scores predicted by LLMs.

Part of the responses from some models cannot be parsed as integer scores due to problems such as an infinite loop in review generation settings. Since the number of such data points is minimal for each model, we exclude these data points when calculating the correlations. Section C.1 shows detailed results, including the parse failure rate.

Our model configurations involve inherent randomness, but we only report the values obtained by a single run as the main results. See Section C.2 for the verification of the score robustness.

[3]https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions

[4]https://amazon-reviews-2023.github.io/

5 *RQ1*: Effect of review texts

5.1 Comparison Method

First of all, we verify whether the per-item review texts improve the personalization performance by off-the-shelf LLMs. We accomplish this by comparing the LLMs' performance based on the three following prompting formats.

First, we only use the user's past numeric scores as the preference information in p_u. More concretely, $p_u = \{(x_u^{(i)}, y_u^{(i)})\}_{i=1}^{k}$, where k is the number of in-context examples given to the model, $x_u^{(i)}$ is the description of i-th item, and $y_u^{(i)}$ is the numeric score u assigned to it in the past. The LLM only outputs the predicted score y'_u. We write this format as S $\rightarrow$ S (Score $\rightarrow$ Score) .

Second, we also put the user's past review texts into p_u. p_u can be written as $p_u = \{(x_u^{(i)}, t_u^{(i)}, y_u^{(i)})\}_{i=1}^{k}$, where $t_u^{(i)}$ is u's textual review for $x_u^{(i)}$. The LLM only outputs the predicted score y'_u as well. We write this format as RS $\rightarrow$ S (Review + Score $\rightarrow$ Score).

Third, in addition to the RS $\rightarrow$ S settings, we modify the instruction I for the LLM to output (t'_u, y'_u), where t'_u is the review text that the LLM expects u to write for the target item x_u. This is the format used by PerSE with fine-tuned LLMs (Wang et al., 2024), and we investigate whether the virtual review written by the off-the-shelf LLM can further enhance its performance. We write this format as RS $\rightarrow$ RS (Review + Score $\rightarrow$ Review + Score). See Section B.5 for more detailed prompt formats.

5.2 Results and Analysis

Figure 2 shows the Spearman correlations obtained with different prompting styles on all combinations of the datasets and the off-the-shelf LLMs. Section C.1 reports concrete numbers including the Kendall-Tau correlations.

In all of the 15 combinations, RS $\rightarrow$ RS outperforms S $\rightarrow$ S. RS $\rightarrow$ S also shows better performance than that of S $\rightarrow$ S. This result suggests that utilizing the review texts written by the users improves the rating prediction accuracy across different datasets and models.

It is also notable that smaller models can outperform larger models with stronger reasoning capabilities under the settings with in-context review texts. In particular, Gemma 3 12B with RS $\rightarrow$ RS on Recipe and Books datasets outperforms all the other models, including larger models known

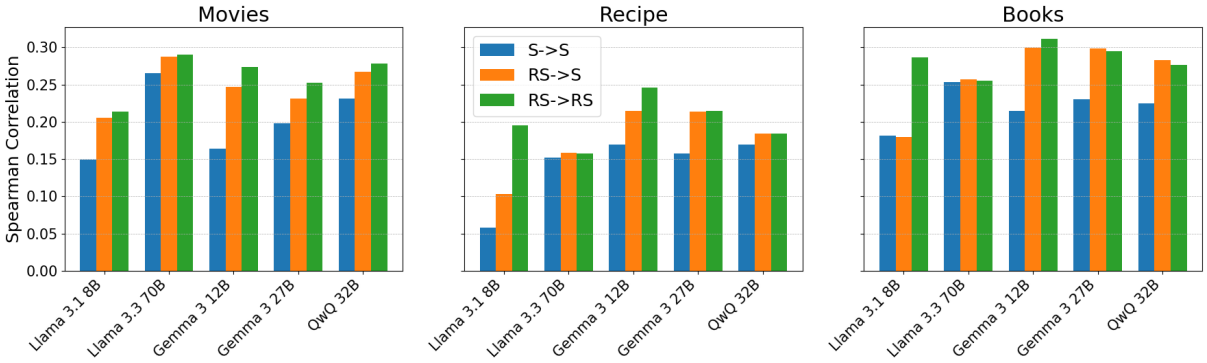


Figure 2: Comparison of S → S , RS → S , and RS → RS prompting on different models and datasets. RS → RS and RS → S show significant performance improvement from S → S, which suggests the impact of the review texts.

for more substantial reasoning capabilities, such as Gemma 3 27B or QwQ 32B. The result indicates that the ability to extract the preference information from the texts can be different from the general reasoning capability.

Another important finding is that instructing the LLMs to write the expected reviews explicitly can further enhance the performance. For most datasets and models combinations, the models perform better with RS → RS than with S → S. The effect is more significant with smaller models such as Llama 3.1 8B and Gemma 3 12B. The preference information given to the models in those two settings is exactly the same, so the difference is made by the instruction to write down the expected review.

We hypothesized that a possible explanation for this phenomenon is that writing the review allows LLMs to predict more extreme scores. Figure 7 shows the comparison of output score distribution of Gemma 12B for the Movies dataset with RS → S and RS → RS prompts. While with RS → S (Figure 7b) the model outputs "neutral" scores such as six or seven very frequently, RS → RS (Figure 7c) results in a flatter output distribution, which is similar to the ground truth. This difference implies that the review writing process widens the possible score range that the LLMs can output. We leave deeper analysis as our future work.

6 *RQ2*: Difficult Settings

6.1 Variants of In-Context Examples

To answer *RQ2*, we make the preference prediction problem more difficult by providing the in-context preference information in the following ways and compare the results with *RQ1*.

Fewer First, we investigate the effect of the number of in-context examples. With the same datasets introduced in Section 4.1, we reduce the number of in-context examples to $k = 1, 3$, and compare the results with Section 5.2, which uses $k = 5$.

Shorter Second, we examine the performance change in the situation where each review is a shorter text. We create the Books (Short) dataset by sampling reviews with less than 200 characters from the same Amazon Reviews'23 (Ni et al., 2019), which is also used for the standard Books dataset. To exclude extremely short reviews, such as single words, we also set a length of 10 as the lower threshold. See Section B.3 for more detailed statistics of the dataset.

Shuffle Third, we randomly shuffle the in-context review texts to verify whether LLMs improve user rating prediction performance by identifying target user characteristics from user review contents.

We create the Movies (Shuffle) dataset, which is made by shuffling the in-context examples of the Movies dataset in Section 4.1. Therefore, in RS → RS and RS → S settings, the target user's past review scores are paired with unrelated reviews written by other users.

6.2 Results and Analysis

Figure 3, 10, and 5 show the results on the two settings with Llama 3.1 8B and Gemma 3 12B. Both models perform better with RS → RS and RS → S compared to S → S, even with fewer in-context examples such as $k = 1, 3$. RS → RS also marks higher performance than RS → S. The results suggest that the findings in Section 5.2 still hold with extremely a small number of in-context examples.

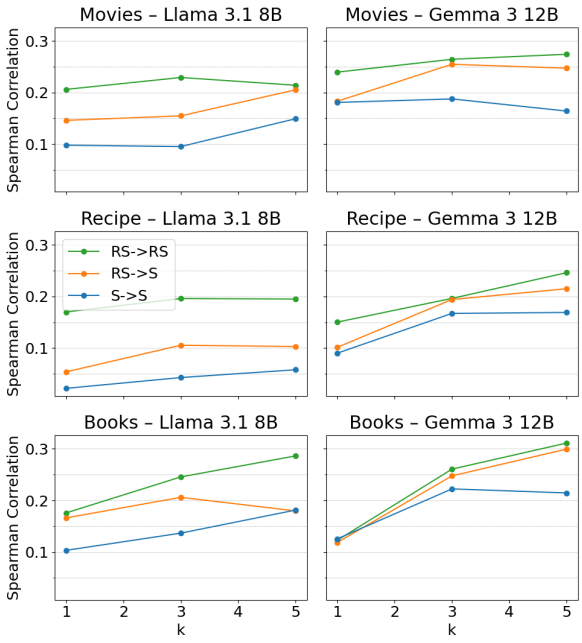


Figure 3: Comparison of the results with $k = 1, 3, 5$. RS $\rightarrow$ RS and RS $\rightarrow$ S enhance the performance even with a fewer in-context examples.

Figure 4: Comparison of the results with the Books and the Books (Short) datasets. Shorter reviews still lead to the performance improvement.

The short review experiment also supports a similar conclusion. Both LLMs show improved performance on the Books (Short) dataset with RS $\rightarrow$ RS compared to S $\rightarrow$ S. This suggests that even short review texts can contribute to the rating prediction task performed by off-the-shelf LLMs.

Although the degree of improvement looks smaller than that with the standard Books dataset, direct comparison is not appropriate because of the difference in rating prediction difficulty in both datasets. As shown in Section B.3, users extracted for the Books (Short) dataset show smaller variance in their integer preference scores, which makes it easier to predict the scores in the Books (Short) dataset solely from the numeric ratings. We leave a more rigorous comparison for future work.

In the shuffle setting, performance improvement by using review texts cannot be observed. On the

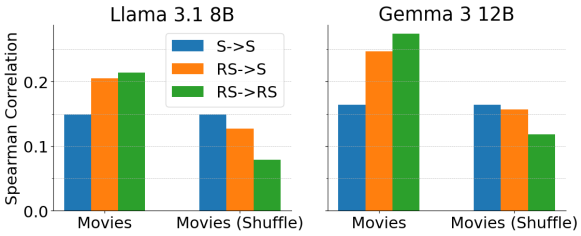


Figure 5: Comparison of the results with the Movies and the Movies (Shuffle) datasets. RS $\rightarrow$ S and RS $\rightarrow$ RS prompting worsen the performance on the Movies (Shuffle) dataset, which suggests that the LLMs actually reference the review contents to predict the target user's preference.

Movies (Shuffle) dataset, since the review texts are more incorporated into the prediction process in RS $\rightarrow$ S and RS $\rightarrow$ RS prompting settings, a significant drop in the prediction performance is observed for the Shuffle dataset, contrary to the improvement in the standard dataset. This result indicates that the LLMs actually reference the review contents to predict the target user's preference, which means that giving the correct reviews as in-context examples is at least required for performance enhancement.

7 *RQ3*: Prompt Engineering

7.1 Prompt Engineering Techniques

We pick various prompt engineering techniques from the related work introduced in Section 2.3 and compare the impact on the user rating prediction performance. All the prompt engineering techniques below are implemented as an extension of the RS $\rightarrow$ RS format. We present the concrete prompts used for this section in Section B.6.

Zero-shot CoT Following Kojima et al. (2022), we add "Let's think step by step" to the end of the prompt and try to trigger the reasoning capability of the LLMs. We investigate whether the prompt engineering techniques for reasoning tasks are effective for the user rating prediction tasks.

Score Range Summary In this format, we first use the LLMs to output the range of user ratings (the most common positive and negative scores). This is initially introduced by Richardson et al. (2023) to summarize the user ratings obtained with retrieval augmentation, but adding an explicit step to summarize the trend of scores could also improve the performance in our settings.

Preference Summary This utilizes a prompt used for KAR (Xi et al., 2024) to summarize the

user preference first, then predict the scores as the next step. Although the original work consumed the summary with fine-tuned models, summarizing preference information in the context can also benefit off-the-shelf LLMs.

Preference Summary + Item Recommendation In addition to the preference summary, we also use a prompt for LLM-Rec (Lyu et al., 2024) to ask LLMs to write the recommendation text for the target item, then perform the rating prediction with both intermediate outputs in the context. We expect this to trigger the LLMs' ability to capture the correlation between the user preference and the preferred item features.

7.2 Results and Analysis

We present the comparison of different prompt engineering techniques for each dataset with Llama 3.1 8B and Gemma 3 12B in Figure 6. The original RS $\rightarrow$ RS prompting achieves the best performance in three out of six settings. The preference summary prompting on the Movies dataset with Llama 3.1 8B shows a noticeable improvement, but the performance enhancements observed in the other cases are negligible, even if they exist.

In particular, Zero-shot CoT prompting leads to worse performance in five out of six combinations of datasets and models. This result may support the findings by Sprague et al. (2024) that CoT leads to better performance mainly for math or logic tasks. Figure 7 illustrates the possible cause of that difference. Although applying RS $\rightarrow$ RS prompting (Figure 7c) widens the range of predicted scores compared to RS $\rightarrow$ S (Figure 7b), applying CoT on it (Figure 7d) makes LLMs predict the neutral scores such as seven more often again. This distribution change could reverse the positive effect obtained with RS $\rightarrow$ RS prompting. With CoT, LLMs tend to output the analysis results for both likes and dislikes of the users at the same time, which may result in the "balanced" output score. We show concrete examples in Section C.3.

8 *RQ4*: Comparison with Self-described Preference

8.1 Self-Described Preference

We verify that the per-item review format is more effective than the self-described preference format used by Sanner et al. (2023). Self-described preference is the text in which the target user describes the sort of items they like. The text typically starts with "I like..." and the preference description is not based on any specific items (see the detailed difference of the two formats in Figure 17 of Appendix B.7).

Sanner et al. (2023) show that the self-described preference text improves the LLMs' performance of top-N prediction, and also claim that the text is more effective than per-item binary preference. However, their problem settings and preference signals are much simpler than ours, so whether the self-described preference format is still adequate for our problem settings is unclear.

8.2 Settings

Datasets Although comparison of the two formats is necessary, the dataset used for their experiments does not contain the per-item review texts, and the ones we use do not have the self-described preference style text either. To fill the gap, we use the Gemma 3 12B and Llama 3.1 8B to transform the per-item reviews to the self-described preference. Implementation details and the example outputs are listed in Section B.7. Using those generated self-described preferences, we query the LLMs to predict the preference scores and measure the correlations with the ground truth labels. We use the preference text generated by the rating prediction model itself.

Prompting Formats As the prompting format, we introduce a $\emptyset \rightarrow$ S format, in which per-item scores are removed from the S $\rightarrow$ S prompt. We combine this with the generated self-described preference. Here, LLMs need to predict the preference scores only based on the self-described preference text. We also combine this self-described preference with the three prompting formats introduced in Section 5.1 and check if adding the self-described preference improves the performance.

8.3 Results and Analysis

Figure 8 compares the user rating prediction performance of Llama 3.1 8B and Gemma 3 12B with and without the self-described preference text. Despite the observation by Sanner et al. (2023), $\emptyset \rightarrow$ S prompting with the self-described preference results in worse performance than RS $\rightarrow$ S prompting without the self-described preference for both models. This indicates that the self-described preference does not work as effectively as the per-item reviews under complex problem settings such as user rating prediction.

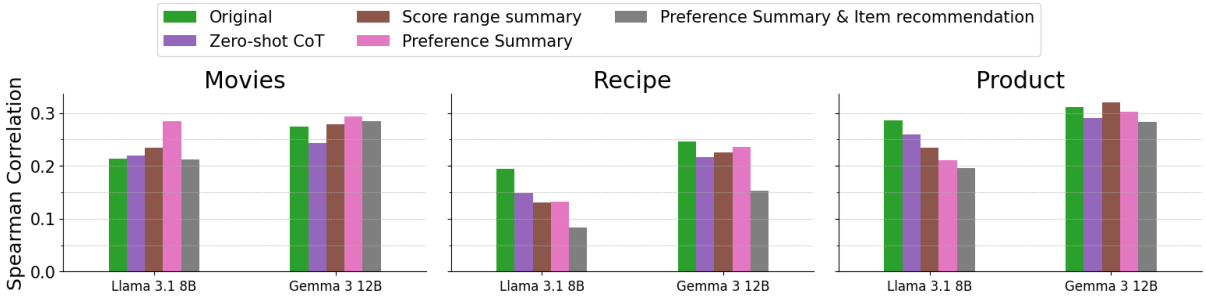
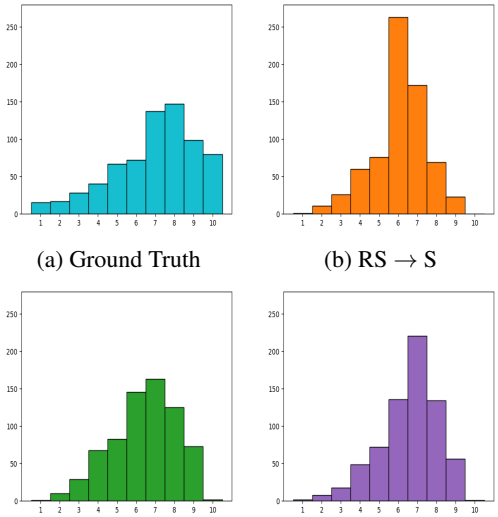


Figure 6: Comparison of the prompt engineering techniques on the user rating prediction task. In most cases the additional techniques do not result in the performance improvement compared to the original RS → RS prompting.



(a) Ground Truth (b) RS → S

(c) RS → RS (d) RS → RS + CoT

Figure 7: Output score distribution of Gemma3 12B on the Movies dataset with different prompting methods. RS → RS flatten the distribution compared to RS → S , but adding CoT partially reverts the effect.

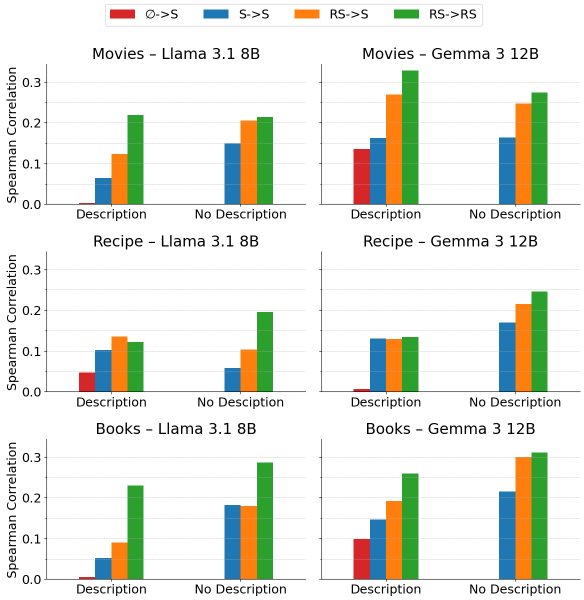


Figure 8: Comparison of rating prediction performance with and without the self-described preference generated by LLMs. The self-described preference does not work as effectively as the per-item reviews under the rating prediction settings.

When the self-described preference and the per-item review texts are combined, both models show performance improvement with RS → RS on the Movies dataset. Still, the performance drops for the other datasets. Since the Movies dataset has longer review texts, it is possible that summarizing the reviews in the self-described preference form helps the models to organize the preference data, while it might be noisy for shorter reviews.

9 Conclusion

In this work, we show that providing a few review texts written by the target user improves the performance of user rating prediction by off-the-shelf LLMs. The positive effect is observed across various models and datasets. We also find that further performance enhancement can be achieved by instructing LLMs to write down the expected review explicitly. The per-item review texts are still effective even if the amount of available preference information is small, as long as the reviews written by the target user are correctly given. Regarding the combination of review texts and existing prompt engineering techniques, zero-shot CoT does not always work effectively for the rating prediction task. Finally, we confirm the advantage of per-item review text over self-described preference used in prior studies.

Our results confirm that review texts are a powerful source for preference prediction and suggest an effective way to utilize the data with off-the-shelf LLMs. We hope these findings lead to the future implementation of data-efficient personalization systems based on off-the-shelf LLMs.

Limitations

Our model configurations are non-deterministic, so the results may differ with different random seeds. Moreover, excluding failed examples is not appropriate when evaluating correlation metrics, and this exclusion may have unexpectedly affected the results.

Another limitation concerns the comparison of datasets with different review lengths. Sampling users from distributions with different review lengths introduced disparities in the difficulty of rating prediction based on numeric information. A rigorous comparison requires a new dataset construction with different lengths of reviews from the same users.

Finally, the analysis of self-described preferences relies on text transformation performed by LLMs, which may affect the quality of the generated preference texts. Although we manually check the similarity of the generated texts with the examples used in previous studies, it is still possible that the artificially generated preference texts have qualitative differences from human-written texts. Again, a new dataset with different styles of preference text from the same user is needed for a more accurate comparison.

Ethical Considerations

The three datasets used in our study are based on user-generated contents crawled from online services. None of the datasets contains sensitive user information, and we ensure we do not disclose any personally identifiable information as part of our work.

In addition, providing the user information in the context of deployed LLM-based systems might result in an unexpected information leakage. Although our work expects the situation where only the data obtained from the target user is used, developers need to pay attention to handling sensitive data when implementing a similar system.

References

Dario Di Palma, Giovanni Maria Biancofiore, Vito Walter Anelli, Fedelucio Narducci, Tommaso Di Noia, and Eugenio Di Sciascio. 2023. Evaluating chatgpt as a recommender system: A rigorous approach. *arXiv preprint arXiv:2309.03613*.

Lukas Eberhard, Thorsten Ruprechter, and Denis Helic. 2025. Large language models as narrative-driven recommenders. In *Proceedings of the ACM on Web Conference 2025*, pages 4543–4561.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. 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.

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*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. LLM-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 583–612, Mexico City, Mexico. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. 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 5976–5982, Hong Kong, China. Association for Computational Linguistics.

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

Kong, China. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392.

Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large language models are competitive near cold-start recommenders for language- and item-based preferences. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 890–896, New York, NY, USA. Association for Computing Machinery.

Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi Fung, Hou Pong Chan, Kevin Small, ChengXiang Zhai, and Heng Ji. 2025. Persona-DB: Efficient large language model personalization for response prediction with collaborative data refinement. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 281–296, Abu Dhabi, UAE. Association for Computational Linguistics.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and 1 others. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

Danqing Wang, Kevin Yang, Hanlin Zhu, Xiaomeng Yang, Andrew Cohen, Lei Li, and Yuandong Tian. 2024. Learning personalized alignment for evaluating open-ended text generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13274–13292, Miami, Florida, USA. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Bin Wu, Zhengyan Shi, Hossein A Rahmani, Varsha Ramineni, and Emine Yilmaz. 2024. Understanding the role of user profile in the personalization of large language models. *arXiv preprint arXiv:2406.17803*.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 12–22, New York, NY, USA. Association for Computing Machinery.

Lanling Xu, Junjie Zhang, Bingqian Li, Jinpeng Wang, Mingchen Cai, Wayne Xin Zhao, and Ji-Rong Wen.

2024. Prompting large language models for recommender systems: A comprehensive framework and empirical analysis. *arXiv preprint arXiv:2401.04997*.

Lanling Xu, Junjie Zhang, Bingqian Li, Jinpeng Wang, Sheng Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Tapping the potential of large language models as recommender systems: A comprehensive framework and empirical analysis. *ACM Trans. Knowl. Discov. Data*. Just Accepted.

Haobo Zhang, Qiannan Zhu, and Zhicheng Dou. 2025. Enhancing reranking for recommendation with llms through user preference retrieval. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 658–671.

Jiarui Zhang. 2024. Guided profile generation improves personalization with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4005–4016, Miami, Florida, USA. Association for Computational Linguistics.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *ACM Transactions on Information Systems*.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, and 1 others. 2024. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*.

A Additional Experiment

A.1 Self-described Preference Generated by Different Models

In Section 8.3, we only use the self-described preference generated by the same model as the one that performs the preference prediction. To check whether the quality of the text transformation affects the result of the user rating prediction, we repeat the same experiment as Section 8.3 with Gemma 3 12B as the self-described preference generator and Llama 3.1 8B as the user rating predictor.

Figure 9 reports the result. Although the performance with $\emptyset \rightarrow$ S is slightly improved when Gemma 3 12B is used as the self-described preference generator model, it is still worse than the RS $\rightarrow$ S without the self-description text. This result suggests that the impact of the model selection on self-described preference generation is lower than that of the existence of the per-item review texts.

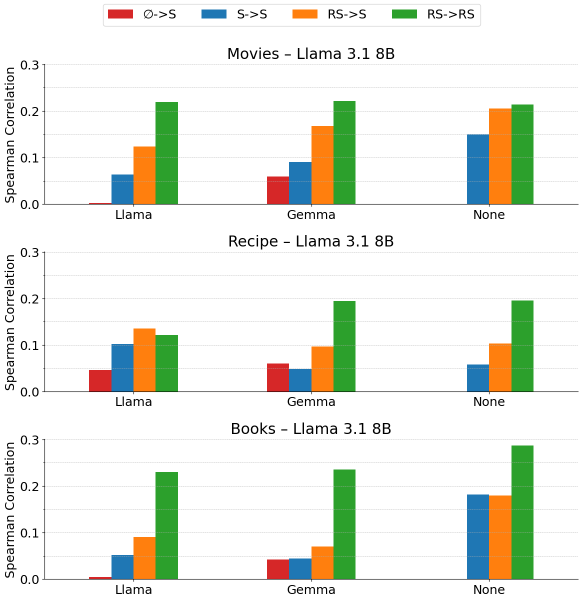


Figure 9: Comparison Llama 3.1 8B's performance with self-description generated by different LLMs

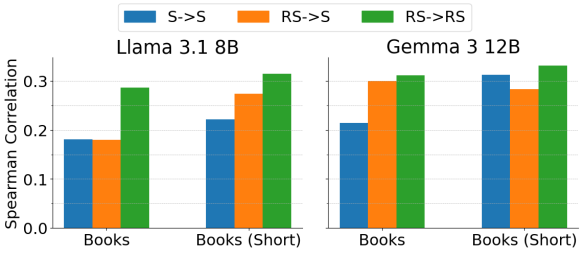


Figure 10: Comparison of the results with the Books and the Books (Short) datasets. Shorter reviews still lead to the performance improvement.

A.2 Experiment with Books (Short) Dataset

B Implementation Details

B.1 Models

During inference with models, we limit the maximum number of generated tokens to 768 for Llama and Gemma models. For QwQ-32B, we set this to 32768 to allow more extended reasoning.

We set the temperature to 0.01 for Llama models. Other parameters follow the default set on the huggingface pages[5][6][7][8][9] as of 2025 April.

[5] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[6] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

[7] https://huggingface.co/google/gemma-3-12b-it

[8] https://huggingface.co/google/gemma-3-27b-it

[9] https://huggingface.co/Qwen/QwQ-32B

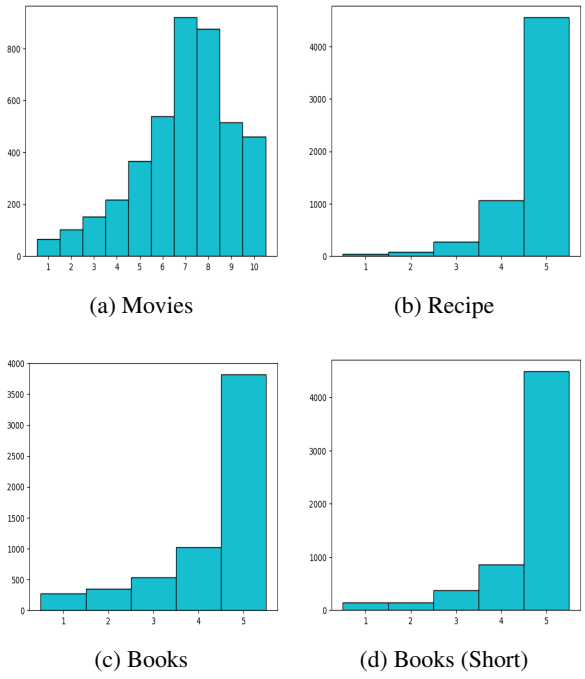
B.2 Computational Resources

We conducted the experiments with different numbers of NVIDIA A100 (40GB), depending on the LLM used for each run. We report the number of GPUs used and the maximum hours spent for each run in Section 5.2 with each model as follows:

- Llama 3.1 8B: 1 GPU, 2 hours
- Llama 3.3 70B: 4 GPUs, 6 hours
- Gemma 3 12B: 1 GPU, 4 hours
- Gemma 3 27B: 2 GPUs, 6 hours
- QwQ 32B: 2 GPUs, 24 hours

Each run in Section 6, Section 7, and **??** took the same number of GPUs and twice as much time as listed above because of the required intermediate outputs.

B.3 Dataset Statistics



(a) Movies (b) Recipe

(c) Books (d) Books (Short)

Figure 11: Label distribution of each dataset (including the in-context examples)

We show the statistics about the datasets we used in the experiments in Table 1. We also present the numeric score distribution in Figure 11. Note that for the Movies (Shuffle) dataset, all the values are the same as those of the standard Movies dataset, since the dataset is just made by shuffling the review text data in the original dataset.

B.4 Other Software and Artifacts

We ran the code for all the experiments with Python 3.11.10. For LLM inference, we used PyTorch (Paszke et al., 2019) 2.6.0 and Transformers (Wolf et al., 2020) 4.50.0. We calculated the evaluation metrics with scikit-learn (Pedregosa et al., 2011) 1.6.1 and SciPy (Virtanen et al., 2020) 1.15.1.

B.5 RS → RS, RS → S and S → S Prompts

We present the base prompt used for Llama Models and Movies dataset with RS → RS settings in Figure 12. The prompt is adopted from PerSE (Wang et al., 2024). The "{plot}" variable is replaced with the target movie plot, and "{icl_example}" is filled with the list of in-context examples described with the template in Figure 13.

For RS → S and S → S settings, the "Review" part of the output format specifier is removed. For S → S, the "Review" part of the in-context example template is removed. Note that newlines are inserted accordingly on the paper to improve the visibility. When applying the prompt to other datasets, we replace words representing the target dataset's domain. The tags like "<|start_header_id|>" are also replaced for experiments with different models.

B.6 Prompt Engineering Techniques

In this section, we introduce detailed prompt templates used for experiments in Section 7.2

Zero-shot CoT We reused the prompt in Figure 12, except that the beginning of the assistant response is replaced with "Let's think step by step.".

Score Range Summary We use the prompt presented in Figure 14 adopted from Richardson et al. (2023) to generate the score range summary text, then add this intermediate output to the prompt in Figure 12 with the prefix "The trend of review scores given by this user is analyzed as follows:"

Preference Summary We use the prompt presented in Figure 18, originally used for KAR (Xi et al., 2024), to generate the analysis of the user preference. This output is added to the rating prediction prompt in Figure 12 with the prefix "The preference of him/her is analyzed as follows:".

Preference Summary + Item Recommendation In addition to the Preference Summary, we also add the item recommendation text generated with the prompt presented in Figure 16, which is originally used in LLM-Rec(Lyu et al., 2024).

Then the item recommendation text is also added to the bottom of the prompt in Figure 12, sur-

Dataset	Num of Examples	Avg Item Description Length	Avg Review Length	Avg Per-user Score Stddev
Movies	702	1142.0	752.8	1.54
Recipe	1000	766.8	370.6	0.44
Books	1000	1134.2	650.7	0.74
Books (Short)	1000	1075.2	84.8	0.49

Table 1: Dataset-level statistics: number of examples, average item-description length (characters), average review length (characters), and per-user score standard deviation.

```
<|start_header_id|>system<|end_header_id|>
You function as an insightful assistant whose
role is to assist individuals in making
decisions that align with their personal
preferences. Use your understanding of their
likes, dislikes, and inclinations to provide
relevant and thoughtful recommendations.
<|eot_id|>

<|start_header_id|>user<|end_header_id|>
[User Question] You will be presented with
several plot summaries, each accompanied by
a review from the same critic. Your task is
to analyze both the plot summaries and the
corresponding reviews to discern the
reviewer's preferences. Afterward, consider
a new plot and create a review that you
believe this reviewer would write based on
the established preferences.

{icl_example}

Please follow the above critic and give a
review for the given plot. Your response
should strictly follow the format:
```json
{{
  "Review": "<proposed review conforms to
         style demonstrated in the previous
             reviews>",
  "Score": <1-10, 1 is the lowest and
           10 is the highest>
}}
```
Please remember to replace the placeholder
text within the "<>" with the appropriate
details of your response.

[The Start of Plot]
{plot}
[The End of Plot]
<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
[Review] Here is the Json format of the review:
```

Figure 12: Query Prompt used for RS → RS examples

```
[The Start of Plot {n}]
{plot}
[The End of Plot {n}]
[Review]
```json
{{
  "Review": "{review}",
  "Score": {score}
}}
```
```

Figure 13: In-Context Example Template used for RS → RS examples

```
A critic's past movie reviews are listed
below:

{icl_example}

Based on this user's past reviews, what
are the most common scores they give
for positive and negative reviews?
Answer in the following form:

most common positive score:
<most common positive score>,
most common negative score:
<most common negative score>
```

Figure 14: Prompt used to generate the score range summarization text

```
A critic's past movie reviews
are listed below:

{icl_example}

Analyze the critic's preferences.
Provide clear explanations based
on details from the past reviews
and other pertinent factors.
```

Figure 15: Prompt used to generate the preference summary

```
The description of a movie plot is as follows:

{plot}

what else should I say if I want to
recommend it to others?
```

Figure 16: Prompt used to generate the item recommendation text

Figure 17: Comparison between self-described preference (top) and per-item review (bottom). Per-item review format can contain more specific preference information, and makes it easy to add more information if available.

rounded by "[The Start of Recommendation Text]" and "[The End of Recommendation Text]" tags.

B.7 Self-Described Preference

Figure 17 illustrates the difference between per-item review and self-described preference formats. For the experiments in Section 8.3, we use the prompt in Figure 15 to transform the per-item review text into the self-description style text. Example texts are listed in Table 3. LLMs successfully generate the self-description style text similar to the original example of Sanner et al. (2023) presented in Table 2.

At the inference time, the self-description text is added to the review prediction prompt in Figure 12 with the prefix "His / her self-description of the preference is as follows:".

```
A critic's past movie reviews are listed
below:

{icl_example}

Write the passage this person would write when
asked to describe their movie preferences.
The passage must start with "I like . . . " and
be no more than 300 characters long.
```

Figure 18: Prompt used to convert the per-item review to self-description style text

C Detailed Results

C.1 Detailed Results of Section 5.2

We report the concrete numbers of Spearman Correlation, Kendall-Tau correlation, and Failure Rate of the experiment of Section 5.2 in Table 4. The failure rate is highest (1.7%) with the combination of Llama 3.1 8B and Books dataset, but generally at an acceptable level.

C.2 Robustness of Metrics

To verify the robustness of the obtained scores, based on six runs, including those reported in Table 4, we report each output metric's average and standard deviation for each model, Movies dataset, and RS $\rightarrow$ RS setting. Compared to the standard deviation values, it is confirmed that the score improvement by incorporating the review data is not statistically negligible..

C.3 Concrete Outputs with Different Prompting Styles

Table 6 lists the outputs on a data point in the Movies dataset by Gemma 3 12B, based on different prompting styles. As the table shows, with RS $\rightarrow$ S the model predicts seven as a generally plausible score, while with RS $\rightarrow$ RS the model predicts three, which is close to the ground truth score. However, when Zero-shot CoT is also applied, the model lists up the user's dislikes and likes first, and predicts a more favorable score of ight as a result. This example aligns with the output distribution change illustrated in Figure 7.

Original Example	I like comedy genre movies, while watching comedy movies I will feel very happy and relaxed. Comedy films are designed to make the audience laugh. It has different kinds of categories in comedy genres such as horror comedy, romantic comedy, comedy thriller, musical-comedy.

Table 2: Example in the original dataset proposed by Sanner et al. (2023)

Gemma 3 12B	I like complex plots with suspense, intrigue, and a touch of action. Gritty noir films and thrillers with morally ambiguous characters are right up my alley! A good story is key.
Llama 3.1 8B	I like complex, suspenseful stories with intricate plots and unexpected twists. I'm drawn to films that explore the human condition, morality, and the blurred lines between right and wrong. I appreciate gritty, atmospheric settings and powerful filmmaking.

Table 3: Examples of self-description style preference generated by LLMs

D License and Intended Use of Scientific Artifacts

In this work, scientific artifacts including datasets (Section 4.1), models (Section 4.2), and other software (Section B.4) are used under the specified license and the terms of use.

E AI Assistance Usage

In this work, ChatGPT[10] has been used for writing elaboration. GitHub Copilot[11] has also been used as a coding assistant for the experiments.

[10] https://chatgpt.com/
[11] https://github.com/features/copilot

Dataset	Model	S → S			RS → S			RS → RS		
		ρ	τ	FR	ρ	τ	FR	ρ	τ	FR
Movies	Llama 3.1 8B	0.149	0.125	0.000	0.205	**0.171**	0.000	**0.215**	0.168	0.003
Movies	Llama 3.3 70B	0.265	0.214	0.000	0.287	0.234	0.000	**0.290**	**0.237**	0.000
Movies	Gemma 3 12B	0.164	0.132	0.000	0.247	0.198	0.001	**0.274**	**0.216**	0.003
Movies	Gemma 3 27B	0.198	0.157	0.000	0.231	0.183	0.001	**0.252**	**0.200**	0.000
Movies	QwQ 32B	0.231	0.183	0.007	0.267	0.216	0.013	**0.279**	**0.225**	0.009
Recipe	Llama 3.1 8B	0.058	0.057	0.000	0.103	0.100	0.000	**0.195**	**0.189**	0.010
Recipe	Llama 3.3 70B	0.152	0.148	0.000	**0.158**	**0.154**	0.000	0.157	0.153	0.001
Recipe	Gemma 3 12B	0.169	0.163	0.000	0.215	0.208	0.000	**0.246**	**0.239**	0.000
Recipe	Gemma 3 27B	0.157	0.151	0.000	0.214	0.205	0.000	**0.215**	**0.208**	0.005
Recipe	QwQ 32B	0.169	0.016	0.000	**0.185**	**0.180**	0.003	**0.185**	**0.180**	0.003
Books	Llama 3.1 8B	0.181	0.169	0.000	0.180	0.167	0.000	**0.286**	**0.258**	0.017
Books	Llama 3.3 70B	0.254	0.234	0.000	**0.257**	**0.237**	0.000	0.255	0.234	0.006
Books	Gemma 3 12B	0.215	0.193	0.000	0.300	0.268	0.000	**0.311**	**0.277**	0.001
Books	Gemma 3 27B	0.230	0.207	0.000	**0.299**	**0.266**	0.002	0.295	0.264	0.010
Books	QwQ 32B	0.225	0.206	0.000	**0.283**	**0.261**	0.005	0.276	0.257	0.002

Table 4: Performance by Dataset, Model and Prompting Method. Symbols: ρ = Spearman correlation, τ = Kendall–τ correlation, FR = failure rate.

Model	ρ	σ_ρ	τ	σ_τ	FR	σ_{FR}
Llama 3.1 8B	0.241	0.013	0.189	0.010	0.005	0.001
Llama 3.3 70B	0.305	0.010	0.250	0.009	0.000	0.000
Gemma 3 12B	0.274	0.007	0.217	0.006	0.002	0.001
Gemma 3 27B	0.251	0.006	0.198	0.005	0.002	0.001
QwQ 32B	0.302	0.012	0.244	0.010	0.007	0.002

Table 5: Statistics of 6 RS → RS runs on the Movies dataset with different models on Movies dataset. Symbols: ρ = Spearman correlation, σ_ρ = its standard deviation, τ = Kendall–τ correlation, σ_τ = its standard deviation, FR = failure rate, σ_{FR} = its standard deviation.

Prompting Method	Score	Raw Response
Ground Truth	1 / 10	this crap gives Dracula a bad name ||| This is one of the most inane films I have ever had the misfortune of viewing...
RS $\rightarrow$ S	7 /10	N/A
RS $\rightarrow$ RS	3 /10	""Review"": ""What a load of hooey! ||| This one was a real mess. Too many characters, too many ridiculous plot twists...
RS $\rightarrow$ RS + Zero-shot CoT	8 /10	Okay, analyzing the critic's preferences:... * **Dislikes:** ""Sugary,"" overly sentimental/romantic ... * **Likes:** Strong characters... ""Review"": ""Another bloodsucker on the loose ||| Well, at least this one doesn't insult the viewer's intelligence too much...

Table 6: Example responses by Gemma 3 12B on the Movies dataset with different prompting methods