RecSys Arena: Pair-wise Recommender System Evaluation with Large Language Models

Anonymous ACL submission

Abstract

Evaluating the quality of recommender systems is critical for algorithm design and optimization. Most evaluation methods are computed based on offline metrics for quick algorithm evolution, since online experiments are usually risky and time-consuming. However, offline evaluation usually cannot fully reflect users' preference. Moreover, many offline metrics such as AUC do not offer sufficient information for comparing the subtle differences between two competitive recommender systems in different aspects, which may lead to substantial performance differences in long-term online serving. Fortunately, due to the strong commonsense knowledge and role-play capability of large language models (LLMs), it is possible to obtain simulated user feedback on offline recommendation results. Motivated by the idea of LLM Chatbot Arena, in this paper we present the idea of RecSys Arena, where the recommendation results given by two different recommender systems in each session are evaluated by an LLM judger to obtain fine-grained evaluation feedback. More specifically, for each sample we use LLM to generate a user profile description based on user behavior history or off-the-shelf profile features, which is used to guide LLM to play the role of this user and evaluate the relative preference for two recommendation results generated by different models. Through experiments, we demonstrate that LLMs can not only provide evaluation results consistent with accuracy and diversity metrics, but also effectively distinguish between algorithms while offering nuanced insights into subjective dimensions.

1 INTRODUCTION

Accurate and comprehensive evaluation of recommendation algorithms is essential in practical recommender system design and optimization (Zangerle and Bauer, 2022; Bauer et al., 2024). However, recommender system evaluation is very challenging due to the complexity of user feedback and rapid shift of data distribution. Recommender system evaluation is typically divided into two main categories: online evaluation and offline evaluation. Although online experiments such as A/B testing can give direct assessments about the overall performance of different recommendation algorithms, it is relatively time-consuming to accumulate sufficient user behaviors to obtain confident results. To facilitate algorithm optimization, researchers often rely on offline evaluation using historical user behavior logs to obtain preliminary assessments before deploying models in real-world settings (Beel et al., 2013). For example, metrics such as AUC and nDCG are widely used in different domains to indicate the ranking quality of recommender systems (Zhu et al., 2022). In addition, researchers devise various metrics to quantitatively measure the behaviors of models in different aspects, such as coverage, diversity, novelty and serendipity (Zangerle and Bauer, 2022). However, these offline metrics may not well reflect user satisfaction and long-term user experience. In addition, many ranking metrics such as AUC are not sufficiently sensitive to distinguish the real quality of different recommender systems, which may not provide valuable reference for picking promising candidate algorithm for online experiments.

In recent years, researchers explore the use of large language models (LLMs) in evaluation due to their rich general knowledge memorization and human-oriented behavior alignment (Kocmi and Federmann, 2023; Zhang et al., 2024; Wang et al., 2023; Oosterhuis et al., 2024; Song et al., 2024). For example, zhang et al. (Zhang et al., 2024) proposed that LLMs can achieve a comparable or even better evaluation accuracy compared to traditional methods in the task of assessing recommendation explanation quality. In addition, they claimed that using the voting results of multiple LLMs can improve the accuracy of evaluations. Wang et al. (Wang et al., 2023) proposed leverag-

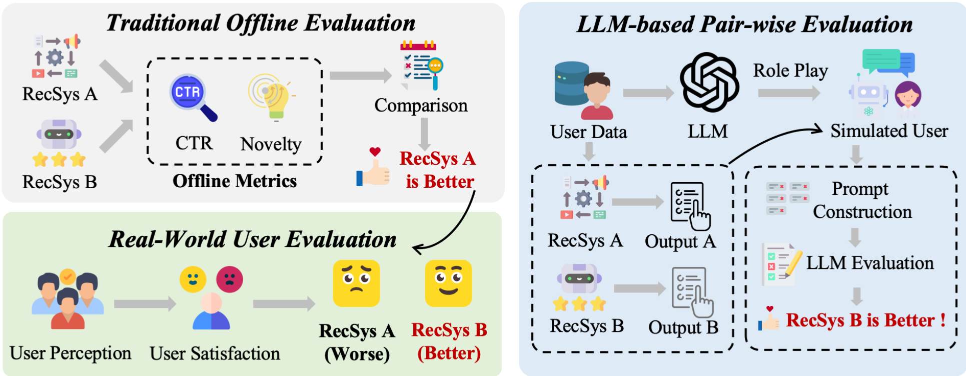

Figure 1: The differences among traditional offline evaluation, real-world user evaluation, and LLM-based pair-wise evaluation

ing LLMs' role-play capability as user simulators to evaluate conversational recommender systems (CRSs). These methods typically rely on LLMs to generate absolute scores for individual algorithms, which fails to capture the relative quality differences between two recommender systems.

Inspired by the relative evaluation methods of LLM such as Chatbot Arena (Chiang et al., 2024) and AlpacaEval (Dubois et al., 2024), we propose a practical LLM-based pair-wise evaluation framework named RecSys Arena, which aims to evaluate the relative performance of recommendation methods on each sample. As demonstrated by the existing work (Dai et al., 2023), in recommendation task, LLM is good at pair-wise ranking while less good at point-wise ranking. In the relative evaluation task, LLMs can simultaneously access information from two recommendation results, facilitating a more granular comparative analysis and uncovering subtle differences to assess their alignment with user preferences.

Figure 1 presents the overview of our approach RecSys Arena. To tackle the limitation of offline evaluations in accurately reflecting user perceptions, we utilize LLMs to simulate users. More specifically, we extract user information from various data sources, including behavioral history and existing profile features. This data is then used to construct a detailed user profile description. By simulating the role of the user, the LLM can generate personalized evaluation. Next, to facilitate pair-wise evaluation, we provide the LLM with the recommendation result lists generated by two different recommender systems when constructing the prompts. Compared to absolute evaluation, relative evaluation offers more contrast information, allowing the LLM to perform a finer-grained assessment of the recommendation results. This approach enhances the LLM's ability to distinguish subtle differences in recommendation results. At the same time, compared to online evaluations, the LLM-based pair-wise evaluation method offers greater feasibility and efficiency. This method allows for rapid testing of different recommendation scenarios, enabling researchers to analyze large datasets and assess various recommendation model performances. Furthermore, by leveraging LLMs, the evaluation process can be conducted at scale, providing a more comprehensive understanding of user preferences while reducing the time and resources typically required for online evaluations. In summary, we leverage the human-like and role-play capabilities of LLMs to conduct pair-wise evaluations of recommendation results, assessing which of the two recommender systems performs better based on an understanding of the user's personal attributes and preferences. Moreover, LLMs are pre-trained on vast data corpora in a self-supervised manner, allowing them to capture extensive domain knowledge. This exposure helps them learn intricate patterns and contextual cues, enhancing their reasoning abilities (Brown, 2020). Additionally, with billions of parameters fine-tuned during training, these models can effectively encode and recall information, facilitating reasoning processes. For example, for categories of items that do not appear in the historical interactions, the LLM will conduct a potential inferential analysis of whether the user might be interested in the item based on personal attributes or other information.

We conducted experiments to demonstrate the effectiveness of the LLM-based method in pair-wise evaluation of recommender systems. In our study, we considered different types of recommendation models, including factorization machines (Rendle, 2010), ID-based recommendation model (Guo et al., 2017), content-based recommendation model (Wu et al., 2019), sequence recommendation model (Kang and McAuley, 2018), and graph neural network-based recommendation model (He et al., 2020). We used two public content recommendation datasets (i.e. MovieLens (Harper and Konstan, 2015) and MIND (Wu et al., 2020)) for evaluation. In our study, we considered both open-source and closed-source LLMs across various sizes, ranging from 8 billion to 236 billion parameters. Additionally, we designed six aspects for evaluating the quality of recommendation results from the user's perspective.

Our study makes the following findings:

- Large language models, leveraging their reasoning capabilities, world knowledge, text generation capabilities, and role-play capabilities, can generate reasonable pair-wise evaluation results. Moreover, when comparing two recommendation models, these results align with the trends observed in offline metrics, such as AUC and Diversity.

- Different large language models exhibit varying effectiveness in the task of recommendation quality evaluation, with larger models generally performing better.

- Pair-wise evaluation based on large language models offers a more nuanced distinction between two different recommendation models with similar performance in terms of AUC and nDCG. RecSys Arena can uncover subtler differences in recommendation results that existing offline metrics might overlook.

2 METHODOLOGY

In this article, we propose a novel and practical approach , called RecSys Arena, to utilize the LLM to conduct pair-wise evaluations of the two recommender systems.

2.1 Problem Formulation

We use $U = (u_1, u_2, ..., u_{|U|})$ to denote the set of users in a recommender system (RS). Input to the RS includes the user's personal attribute information S and viewing histories H of users respectively. The RS recommend multiple items to each user u, which are defined as $I_u = (i_1, i_2, ..., i_{|I_u|})$.

Given two RSs R_A and R_B, we use I_{R_A} and I_{R_B} to represent the corresponding recommendation result lists generated by systems R_A and R_B, respectively. Let $f(\cdot)$ represent the evaluation method. The pair-wise evaluation results of systems R_A and R_B, as provided by the LLM, can be expressed as:

$$f_{LLM}(U, R_A, R_B) = LLM(S_U, H_U, I_{R_A}, I_{R_B}, \mathcal{P}) \quad (1)$$

where $\mathcal{P}$ denotes the prompt template.

The primary goal of pair-wise LLM-based evaluation is to 1) measure, from the user's perspective, which recommender system, R_A or R_B, has better overall performance for the same user. Note that the overall performance refers to the comparison results that take into account multiple evaluation dimensions (to be introduce in 2.2). Along with the measurable overall performance, 2) LLM-based evaluation method also reports a detailed, interpretable qualitative analysis explaining the evaluation reasons for each dimension, which could facilitate the developer to further make targeted improvements to the recommender system.

2.2 Evaluation Dimensions

To address the issue that existing offline evaluation metrics cannot evaluate the quality of recommendation results from the user's perception, we primarily focus on user experience when designing the evaluation dimensions. We assume that users of the recommender system serve as the most accurate evaluators of the recommendation results. We consider both mainstream dimensions of concern (e.g., accuracy, satisfaction) and dimensions that traditional offline metrics cannot evaluate (e.g., inspiring content, positive impact). Therefore, for the paired recommendation result I_{R_A} and I_{R_B}, the LLM is asked to give a comparative evaluation result from the following 6 aspects:

Accuracy: This recommendation result list aligns well with my interests.

Satisfaction: I am satisfied with the results provided by this recommender system.

Inspiration: The recommended items inspire me to think, promote further exploration, and enhanced my willingness to interact with the recommendation platform.

Content Quality: The recommended items are of high quality.

Transparency: The recommendation results are associated with one of my personal information or an interaction history, and it is evident which

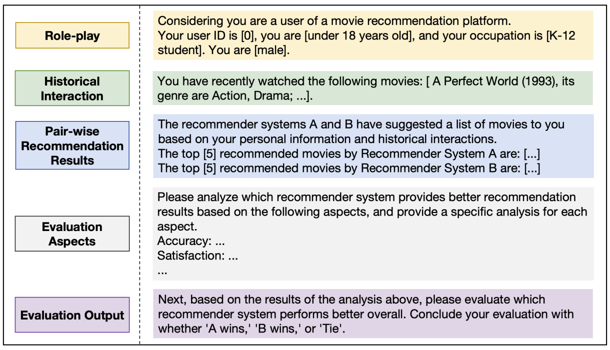

Figure 2: The outline of evaluation prompt template applied in our study

feature is relevant.

Impact on users: The impact of this recommendation result list on me is positive.

2.3 Prompt Construction

In this section, we introduce the construction of the prompt $\mathcal{P}$, designed to guide LLM in evaluating the quality of pair-wise recommendations from specific aspects, based on user profiles and viewing histories. As shown in Figure 2, $\mathcal{P}$ consists of five components.

In the first part of the prompt, we leverage the role-play capability of LLMs to facilitate a personalized evaluation of recommendation results. To do this, we provide the LLM with the user's personal attribute information, including age, occupation, gender, and other relevant details. The second part of the prompt consists of the user's viewing history, such as information on movies they have watched or news they have clicked on. This content is included to allow the LLM to perceive the user's preferences, enabling it to conduct subsequent evaluations from the user's perspective. Please note that the MIND dataset does not contain any personal user information. Instead, we provide the historical records of news articles that users have browsed, allowing the LLM to understand the user profile. This approach also helps the LLM gain insights into user preferences and behaviors. Next, the recommendation results from the two systems, R_A and R_B, are presented to the LLM via the prompt. These recommendation results will include specific item information, such as the titles and genres of the movies. In the evaluation section of the prompt, we list the descriptions of each evaluation dimension to assist the LLM in understanding the specific content that requires assessment for each dimension. Additionally, the evaluation dimensions in this prompt template can be dynamically adjusted, further enhancing the scalability of the evaluation framework. This flexibility allows researchers to tailor the evaluation criteria to suit different contexts and objectives, making it applicable across a wide range of scenarios. The main objective of this section is to allow the LLM to make evaluative judgments based on its analysis of each evaluation aspect. We aim to guide the LLM through a step-by-step thought process, similar to the Chain-of-Thought (Wei et al., 2022). This method encourages deeper reasoning and enhances the quality of the evaluation by building on prior insights. Finally, the LLM is asked to output the qualitative analysis for each dimension, along with an overall comparative evaluation of the pair-wise recommendation results from systems R_A and R_B.

2.4 LLM Evaluator Construction

We utilize pre-trained LLMs to provide comparative evaluations for paired recommender systems. The LLM receives the user's personal attribute information, viewing histories, and recommendation results from the two recommender systems R_A and R_B under test, accompanied by the prompt $\mathcal{P}$ to describe the evaluation instruction. LLMs trained on massive corpora of unlabelled data possess a wealth of general knowledge, which aids them in understanding recommended items, such as movies. The reason for their strong power can be concluded as they do not need task-specific training data and can be pre-trained on tremendous in-the-wild data in a self-supervised manner (a.k.a. pre-training), so that sufficient domain knowledge can be captured (Radford, 2018; Devlin, 2018; Brown, 2020).

Previous research on evaluation based on LLMs has mostly involved absolute evaluation (Zhang et al., 2024; Kocmi and Federmann, 2023). Our approach differs from previous studies in that we ask the LLM to conduct a comparative evaluation of two recommendation results, thereby providing a relative assessment. LLMs perform better on pair-wise tasks (Dai et al., 2023). On one hand, using relative evaluation allows the LLM to simultaneously access information from two recommendation results, facilitating a more nuanced comparative analysis. This enables the LLM to uncover subtle differences and assess how well each result aligns with user preferences. On the other hand, absolute scoring evaluations often provide limited context, making it challenging for the model to identify and distinguish between the merits of individual recommendations. By leveraging relative evaluation, we enhance the LLM's capacity to perform finer-grained assessments.

We conduct a statistical analysis of the evaluation results generated by the LLM. To measure the degree of victory between the two models more precisely, we designed the quantile $\mathcal{Q}$ metric. Specifically, we calculate the quantile $\mathcal{Q}$ using the following formula:

$$\mathcal{Q} = \frac{(N_{win} + N_{tie})}{(N_{lose} + N_{tie})} \tag{2}$$

where N_{win} denotes the number of samples in the test set where the RS is deemed to have won, N_{tie} indicates the number of samples where the RSs tied, and N_{lose} represents the number of samples in which the RS lost. A larger value of $\mathcal{Q}$ indicates a greater degree of victory for the RS.

3 EXPERIMENTAL SETUP

We conduct a series of experiments to answer the following research questions (RQs):

RQ1: What is the overall performance of the LLM-based evaluation method for recommender systems?

RQ2: What is the performance of the LLM-based evaluation method for recommender systems in different evaluation sub-dimensions?

RQ3: Can the evaluation results of our method align with the users' explicit evaluations?

RQ4: How does our evaluation method distinguish between different recommendation models?

3.1 Evaluation Metrics

We can initially assess the effectiveness of evaluation based on LLMs by determining whether the results provided by the LLMs align with offline metrics. We consider the three popular metrics: AUC (Area under the Curve) (Ling et al., 2003), nDCG@k (Normalized Discounted Cumulative Gain for the top k recommendations) (Järvelin and Kekäläinen, 2002), URD (User Recommendation Diversity) (Qin and Zhu, 2013).

3.2 Recommender Systems and Datasets

In this study, we use two content recommendation datasets (i.e., Movielens and MIND) released by the previous study (Harper and Konstan, 2015; Wu et al., 2020) as our evaluation datasets, which have been widely used in the existing studies (Lin et al., 2022; Wu et al., 2019; An et al., 2019; Xie et al., 2022). More specifically, MovieLens (Harper and Konstan, 2015) comprises data from 1 million movie ratings provided by 6,040 users across 3,883 movies. Within this dataset, user's attributes include gender, age, and occupation. MIND (Wu et al., 2020) is a news dataset and was collected from the user behavior logs of Microsoft News. The dataset contains 161,013 news and 24,155,470 reading records. We followed existing work (He et al., 2017; Wu et al., 2020) for the splitting of the training, validation, and test sets.

Table 1: The recommendation accuracy of recommender systems in terms of AUC and nDCG@5

RS	MovieLens		MIND	
	AUC	nDCG@5	AUC	nDCG@5
FM	0.5701	0.0557	0.4857	0.1966
NRMS	0.7521	0.1216	0.5004	0.2242
LightGCN	0.6824	0.1101	0.4990	0.2197
SASRec	0.6772	0.1086	0.4985	0.2190
DeepFM	0.6146	0.0970	0.4956	0.2183

In our study, we use 5 categories of recommender systems as the subject of the evaluation in total. We considered different types of recommendation models under test: Factorization Machines, the content-based model, the graph-based model, the sequence recommendation model, and the ID-based model. We describe the specific information of the recommendation models as follow. Rendle et al. propose factorization machine (FM) (Rendle, 2010), which improve upon logistic regression models by addressing the challenge of training model parameters in sparse data scenarios. NRMS (Wu et al., 2019) is a representative of content-based recommendation models that employs multi-head self-attention mechanisms for encoding content, such as news titles. LightGCN (He et al., 2020) is a model that simplifies Graph Convolutional Networks (GCNs) by focusing solely on the core component of neighborhood aggregation for collaborative filtering. SASRec (Kang and McAuley, 2018) is a sequence recommendation model based on self-attention mechanisms. DeepFM (Guo et al., 2017) is an extension of Wide&Deep that synergistically integrates factorization machines for recommendation and deep learning for feature learning, emphasizing both low- and high-order feature interactions.

Table 1 lists the values of the evaluation metrics (i.e., AUC and nDCG@5) for the above five recommendation models, respectively on the Movielens and MIND datasets.

3.3 LLMs

We conduct experiments with three LLMs, including the open-source LLM and proprietary LLM. GPT-4o (gpt, 2024) is a proprietary LLM released by OpenAI. With targeted optimizations, GPT-4o delivers superior performance in generating accurate and contextually relevant responses, benefiting from refined training techniques and updates. DeepSeek-V2.5 (DeepSeek-AI, 2024) is a strong Mixture-of-Experts (MoE) langage model that excels in writing and instruction-following. It comprises 236B total parameters. Llama3.1-8B-Instruct (lla, 2024) is fine-tuned specifically to follow and execute user instructions more accurately, making it better at handling tasks that involve clear directives or specific commands. To make the output as deterministic as possible, we set temperature=0 when calling the API. In our experiments, we evaluated the top-5 items recommended by each recommender system.

4 EXPERIMENTAL RESULTS

4.1 RQ1: The Overall Performance of Evaluation

Our main results are displayed in Table 2. Specifically, we conducted pair-wise evaluations by separately comparing the ID-based recommendation model (i.e., DeepFM), content-based recommendation model (i.e., NRMS), sequential recommendation model (i.e., SASRec), and graph network-based recommendation model (i.e., LightGCN) with FM. First, we present the proportions of "**Win**", "**Tie**", and "**Lose**" for each RS relative to FM in the evaluation results provided by the LLM. Then, in Table 2, we calculated the quantile $\mathcal{Q}$ (i.e., $(N_{win} + N_{tie})/(N_{lose} + N_{tie})$) shown in Column "Q" and listed the rankings of the quantiles $\mathcal{Q}$ in Column "Rank".

To investigate the overall effectiveness of LLMs in evaluating recommendation quality, we use the AUC from traditional metrics as a reference. Specifically, we examine whether the ranking results of recommendation quality provided by LLMs are consistent with the ranking results based on traditional accuracy metric (i.e., AUC).

Impact of different LLMs. In our experiments, we examined differences across various LLMs. The ranking results provided by GPT-4o and DeepSeek-V2.5-236B are consistent with traditional ranking results (as shown in Table 1). However, the ranking results provided by Llama3.1-8B-Instruct show some discrepancies. Moreover, in the comparative evaluation results provided by Llama3.1-8B-Instruct, the likelihood of the tested models being tied is higher. This indicates that Llama3.1-8B-Instruct's evaluation criteria may be more lenient, potentially leading to a higher incidence of similar performance scores among different models. Larger LLMs are better equipped to evaluate recommendation quality. To analyze the correlation between LLM-based evaluations and offline metrics, we computed the Pearson correlation (Cohen et al., 2009) between AUC and $\mathcal{Q}$. Due to page limit, we present the results in Appendix B.

Impact of different Datasets. Furthermore, we considered the datasets conducting different tasks, including movies and news recommendations. Compared to the Movielens dataset, the MIND dataset provides more textual content, including news titles and abstracts, which is more conducive to the LLM's understanding of items. From the horizontal comparison of the results from both datasets, the LLM maintains a consistent win-loss ratio for evaluation of ID-based recommendation models, sequential recommendation models, and graph-based recommendation models. However, as shown in Table 2, content-based recommendation models perform better on the MIND dataset, indicating that different types of models have different applicable scenarios. The evaluation results can guide the selection of models in different scenarios.

Impact of different Models. In addition to recommendation models with significantly different recommendation accuracy (e.g., relative to FM), we also conducted pair-wise evaluations on recommendation models with similar AUC. We present the specific values of quantile $\mathcal{Q}$ for the pair-wise recommendation models in Figure 3. Figure 3 shows that for recommendation models with very similar AUC values, LLMs can effectively distinguish the quality of recommendations between them. In other words, it implies that the LLM-based pair-wise evaluations can more effectively identify subtle differences in performance that offline metrics might not capture.

4.2 RQ2: The Multiple-Aspect Performance of Evaluation

The multiple-dimensions we designed in Section 2.2 are intended to assist LLMs in deriving an overall evaluation result. Specifically, they guide the LLM to first consider the comparative evalu-

Table 2: The overall performance of LLM-based pair-wise evaluation

LLM	RS	MovieLens					MIND				
		Win(%)	Tie(%)	Lose(%)	Q	Rank	Win(%)	Tie(%)	Lose(%)	Q	Rank
GPT-4o	NRMS	61.1	14.7	24.2	1.9485	1	71.9	11.0	17.1	2.9501	1
	LightGCN	62.8	8.4	28.8	1.9139	2	60.5	20.9	18.6	2.0607	2
	SASRec	60.3	10.7	29.0	1.7884	3	59.9	19.7	20.4	1.985	3
	DeepFM	59.1	11.4	29.5	1.7237	4	55.0	15.2	29.8	1.56	4
DeepSeek-V2.5	NRMS	64.7	12.6	22.7	2.1898	1	66.7	10.3	23.0	2.3123	1
	LightGCN	62.2	9.7	28.1	1.9021	2	62.6	15.3	22.1	2.0828	2
	SASRec	61.3	9.2	29.5	1.8217	3	60.6	14.9	24.5	1.9162	3
	DeepFM	64.7	12.6	22.7	1.7680	4	58.7	12.4	28.9	1.7215	4
Llama3.1-8B	NRMS	50.8	28.6	20.6	1.6138	3	54.8	27.1	18.1	1.8119	1
	LightGCN	50.5	37.9	11.6	1.7858	1	43.7	40.7	15.6	1.4991	3
	SASRec	45.0	31.6	23.4	1.3927	4	50.3	35.6	14.1	1.7283	2
	DeepFM	49.9	38.7	11.4	1.7684	2	45.1	34.9	20.0	1.4571	4

* In the pair-wise evaluation, recommender system A R_A is one of NRMS, LightGCN, SASRec, or DeepFM, while recommender system B R_B is FM.

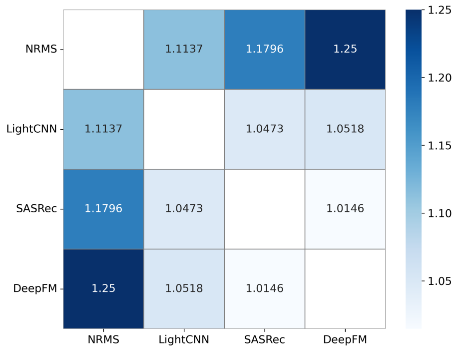

Figure 3: Values of Q in LLM-based pair-wise evaluation for recommender systems with similar AUC

Table 3: The performance of LLM-based pair-wise evaluation in terms of inspiration

LLM	RS	URD	Win(%)	Tie(%)	Lose(%)	Q	Rank
GPT-4o	SASRec	0.1968	69.0	10.8	20.2	2.5741	1
	LightGCN	0.1962	60.1	4.9	35.0	1.6290	2
	NRMS	0.1963	59.6	5.6	34.8	1.6138	3
	DeepFM	0.1954	57.3	11.0	31.7	1.5995	4
DeepSeek-V2.5	SASRec	0.1968	62.8	12.7	24.5	2.0295	1
	LightGCN	0.1962	58.9	20.5	20.6	1.9318	2
	NRMS	0.1963	61.0	8.3	30.7	1.7769	3
	DeepFM	0.1954	56.9	11.1	32.0	1.5777	4

* In the pair-wise evaluation, recommender system A R_A is one of NRMS, LightGCN, SASRec, or DeepFM, while recommender system B R_B is FM.

User: {age: 45~49}

GPT-4o Result:

Transparency: 'A wins' - Recommender system A seems to be more transparent as it is associated with one of the user's personal information (age) and it is clear which feature it is. Recommender system B doesn't provide any specific information about how the recommendations were made.

(a) **Case 1**

User: {age: 25~34; occupation: writer; gender: female}

GPT-4o Result:

Transparency: 'A wins' - Recommender system A seems to be more transparent as it is associated with the user's personal information (age, gender, etc.). System A provided a list that includes movies that are likely to be more appealing to the user, given their age and gender. The inclusion of 'The Little Mermaid' (1989), a classic animated movie, and 'Breakfast of Champions' (1999), a comedy, could be more satisfying to the user than the horror and thriller movies recommended by system B.

(b) **Case 2**

Figure 4: Two showcases of LLM-based pair-wise evaluation results for the transparency dimension

ation results of these 6 sub-dimensions, and then synthesize them to arrive at an overall evaluation result. This step-by-step thought process is more conducive to obtaining reliable and accurate evaluation results. However, we still need to verify the effectiveness of the sub-dimension evaluations. We chose to validate the dimensions of inspiration and transparency.

4.2.1 Inspiration

Table 3 presents the pair-wise evaluation results provided by GPT-4o and DeepSeek-V2.5 in terms of inspiration aspect. Table 3 shows that the evaluation trends provided by the LLM in terms of inspiration align well with the offline metric URD. Furthermore, in specific sub-dimensions (such as inspiration), the LLM's evaluation results also demonstrate better differentiation.

4.2.2 Transparency

The transparency metric is primarily used to measure whether the model has utilized sufficient and correct features to infer and predict the recommended item. We assess whether the LLM can effectively evaluate the transparency metric by investigating whether it can perceive changes in the features used during training. Specifically, we trained the DeepFM model using different features. One model was trained considering only the features $user\_id$ and $item\_id$. For the training of another model, we included more user features such as age, $gender$, and $occupation$.

In Case 1, given a user aged 45-49, the recommendation results from recommender system B (a model trained only with $user\_id$ and $item\_id$) included movies in the category of "Children's". GPT-4o provided an evaluation on the transparency dimension, stating "A wins." The rationale for this assessment is that Recommender System A's recommendation results are more aligned with the user's age. This example shows that LLMs are capable of distinguishing differences in recommendation results produced by recommendation models trained on different features. Furthermore, the LLM can pinpoint which specific feature is responsible for the observed differences.

In Case 2, the user's specific characteristics are: age 25-34, occupation as a writer, and gender female. However, System B recommended movies categorized as "Horror" to her, a genre that had not appeared in her historical viewing list. From the results of GPT-4o, it is evident that LLMs can analyze and perceive that recommending movies categorized as "Horror" to this user is inappropriate based on her personal attributes.

4.3 RQ3: Alignment with User-Centric Explicit Evaluations

To validate the effectiveness of our method in evaluating subjective dimensions (e.g., satisfaction), we use the Yelp dataset, which contains explicit user feedback (e.g., user ratings), to demonstrate its alignment with explicit user evaluation results. However, in practical scenarios, explicit user feedback is often difficult to obtain, and in most situations, only implicit feedback (e.g., user clicks) is available. The Yelp dataset is collected from a popular business review platform, encompassing a wide range of businesses such as restaurants, shopping malls, and hotels. It includes user feedback in the form of ratings and written reviews. The dataset covers over 160,000 businesses and contains more than 8.6 million reviews from eight different cities. In our study, we use explicit nDCG@5 (Liu et al., 2010) to measure the recommendation quality based on users' explicit feedback, as this metric reflects users' subjective perceptions. As shown in Table 4, the experimental results indicate that our evaluation method can serve as a substitute for explicit feedback-based evaluation, effectively assessing the aspect of user perception. Due to page limit, the experimental results on the MovieLens dataset are presented in Appendix C.

Table 4: The performance of LLM-based pair-wise evaluation in terms of explicit feedback-based evaluation on YELP.

LLM	RS	explicit nDCG@5	Win(%)	Tie(%)	Lose(%)	Q	Rank
GPT-4o	NRMS	0.2540	64.8	14.0	21.2	2.2386	1
	DeepFM	0.2431	63.2	14.8	22.0	2.1195	2
	LightGCN	0.2387	60.1	15.3	24.6	1.8897	3
	SASRec	0.2303	59.5	26.8	23.7	1.8859	4
DeepSeek-V2.5	NRMS	0.2540	65.9	13.5	20.6	2.3284	1
	DeepFM	0.2431	63.6	14.1	22.3	2.1346	2
	LightGCN	0.2387	62.2	15.1	22.7	2.0449	3
	SASRec	0.2303	58.7	17.4	23.9	1.8426	4

* In the pair-wise evaluation, recommender system A R_A is one of NRMS, LightGCN, SASRec, or DeepFM, while recommender system B R_B is FM.

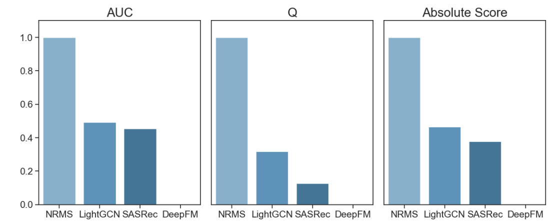

Figure 5: The differences among offline metric (AUC), LLM-based pair-wise evaluation reaults ($\mathcal{Q}$), and LLM-based absolute evaluation score

4.4 RQ4: Relative and Absolute Evaluation

To investigate the differences in differentiation between relative and absolute evaluation using LLMs, we conducted a study in which we employed GPT-4o for the absolute evaluation of four recommender systems (i.e., NRMS, LightGCN, SASRec, and DeepFM) with similar AUC values trained on the MovieLens dataset. For each recommendation result list, we assigned a score between 0 and 1 using GPT-4o. In Figure 5, we present the normalized results for the offline metric (i.e., AUC), the pair-wise evaluation quantiles $\mathcal{Q}$, and the absolute evaluation scores for the four recommendation models: NRMS, LightGCN, SASRec, and DeepFM. Figure 5 reveals that the LLM-based relative assessment provides better differentiation among these four models. For example, for the two most closely matched models in terms of recommendation accuracy, LightGCN and SASRec, the quantiles $\mathcal{Q}$ derived from the LLM-based pair-wise evaluation still provide a better way to distinguish them, capturing the subtle differences between them.

5 CONCLUSION

We propose a practical LLM-based pairwise evaluation method for recommender systems. It aligns with offline metrics while improving discrimination, capturing finer distinctions between models. Our experiments validate its effectiveness across various LLMs.

Limitations

This paper mainly focuses on the evaluation of content recommender systems. However, our findings suggest that, given appropriate information, large language models (LLMs) can generate reliable evaluation results. This raises the possibility that the pairwise evaluation method we propose may be applicable to other types of recommender systems, such as social, product, or music recommendations, especially in light of the rapid development of multimodal large language models. Additionally, we are interested in exploring whether the inclusion of new types of information, such as social context, could contribute to more accurate evaluations. We leave this investigation for future work.

References

2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/.

2024. Introducing meta llama 3: The most capable openly available llm to date. https://ai.meta.com/blog/meta-llama-3/.

Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Annual Meeting of the Association for Computational Linguistics*, pages 336–345.

Christine Bauer, Eva Zangerle, and Alan Said. 2024. Exploring the landscape of recommender systems evaluation: Practices and perspectives. *ACM Transactions on Recommender Systems*, 2(1):1–31.

Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. 2013. Research paper recommender system evaluation: a quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, pages 15–22.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. *Noise Reduction in Speech Processing*, pages 1–4.

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 *ACM Conference on Recommender Systems*, pages 1126–1132.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *International Conference on Research on Development in Information Retrieval*, pages 639–648.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *International Conference on Data Mining*, pages 197–206.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520*.

Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, pages 2320–2329.

Charles X Ling, Jin Huang, Harry Zhang, et al. 2003. Auc: a statistically consistent and more discriminating measure than accuracy. In *International Joint Conference on Artificial Intelligence*, volume 3, pages 519–524.

Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. 2010. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1445–1448.

Harrie Oosterhuis, Rolf Jagerman, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2024. Reliable confidence intervals for information retrieval evaluation using generative ai. In *ACM Knowledge Discovery and Data Mining*, pages 2307–2317.

Lijing Qin and Xiaoyan Zhu. 2013. Promoting diversity in recommendation by entropy regularizer. In *International Joint Conference on Artificial Intelligence*. Citeseer.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Steffen Rendle. 2010. Factorization machines. In *International Conference on Data Mining*, pages 995–1000.

Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. Finesure: Fine-grained summarization evaluation using llms. In *Annual Meeting of the Association for Computational Linguistics*.

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

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

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *Conference on Empirical Methods in Natural Language Processing*, pages 6389–6394.

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

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *IEEE International Conference on Data Engineering*, pages 1259–1273.

Eva Zangerle and Christine Bauer. 2022. Evaluating recommender systems: survey and framework. *ACM Computing Surveys*, 55(8):1–38.

Xiaoyu Zhang, Yishan Li, Jiayin Wang, Bowen Sun, Weizhi Ma, Peijie Sun, and Min Zhang. 2024. Large language models as evaluators for recommendation explanations. *arXiv preprint arXiv:2406.03248*.

Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. Bars: Towards open benchmarking for recommender systems. In *International Conference on Research on Development in Information Retrieval*, pages 2912–2923.

Considering you are a user of a movie recommendation platform.
Your user ID is 0, you are under 18 years old, and your occupation is K-12 student. You are male.

You have recently watched the following movies: Face in the Crowd, A (1957), its genre is Drama. Star Wars: Episode V - The Empire Strikes Back (1980), its genre are Action, Adventure, Drama, Sci-Fi, War. Peacemaker, The (1997), its genre are Action, Thriller, War. African Queen, The (1951), its genre are Action, Adventure, Romance, War. Skin Game, The (1931), its genre is Drama.

The recommendation systems A and B have suggested a list of movies to you based on your personal information and historical interactions.
The top 5 recommended movies in the list given to you by Recommendation System A are: Pocahontas (1995), its genre are Animation, Childeren's, Musical, Romance. Batman Returns (1992), its genre are Action, Adventure, Comedy, Crime. Perfect World, A (1993), its genre are Action, Drama. Abyss, The (1989), its genre are Action, Adventure, Sci-Fi, Thriller. Farewell to Arms, A (1932), its genre are Romance, War.
The top 5 recommended movies in the list given to you by Recommendation System B are: Babe (1995), its genre are Children's, Comedy, Drama. Dear Jesse (1997), its genre is Documentary. Breathing Room (1996), its genre is Romance. Abyss, The (1989), its genre are Action, Adventure, Sci-Fi, Thriller. Little Boy Blue (1997), its genre is Drama.

Please analyze which recommendation system provides better recommendations based on the following aspects, and provide specific analytic insights for each aspect.
Accuracy: This list of recommendations aligns well with my interests.
Satisfaction: I am satisfied with these recommendation results.
Inspiring Content: Recommended movies provoke my thoughts, spark my curiosity, encourage further exploration, and enhance my interaction with the recommendation platform.
Content Quality: The recommended items are of high quality.
Explainability/Transparency: The recommendation is associated with one of my personal information or an interaction history, and it is clear which feature it is.
Impact on users: The impact of this recommendation on me is positive.

Next, based on the results of the analysis above, please evaluate which recommender system performs better overall.
Conclude your evaluation with whether 'A wins', 'B wins', or 'Tie'.

Figure 6: An Example of Prompt for Pair-wise Evaluation of Recommendation System in Practice

GPT-4o Result:

Accuracy: Recommendation System A: This system suggests films from genres like Action ("Batman Returns," "A Perfect World," "The Abyss") and Drama ("A Perfect World") that align with my recent viewing history of action-packed, war-themed, and drama films. Recommendation System B: While B offers The Abyss and Little Boy Blue (Drama), the other suggestions—Babe (Children's, Comedy, Drama), Dear Jesse (Documentary), and Breathing Room (Romance)—are less aligned with my recent viewing history. Winner: A wins.

Satisfaction: Recommendation System A: Given that A suggests movies with a strong focus on action, war, and drama, including The Abyss and Batman Returns, I would likely find this list more satisfying. Recommendation System B: B includes films like Babe and Dear Jesse, which may not capture my attention as well as A's more action-oriented recommendations. Winner: A wins.

Inspiring Content: Recommendation System A: Films like A Perfect World (Action, Drama) and Farewell to Arms (Romance, War) may provoke thought due to their mature and complex themes, especially for a younger viewer who has been watching war and drama films. However, Pocahontas feels less inspiring for my age and interest level. Recommendation System B: Although Little Boy Blue (Drama) might inspire deeper reflection, other films like Babe and Dear Jesse are less likely to provoke thought based on my interests. The Abyss provides a sense of adventure, but overall, B's list feels less likely to encourage further exploration. Winner: A wins.

Content Quality: Recommendation System A: A offers high-quality films such as The Abyss, Batman Returns, and A Perfect World, all of which are well-regarded in their genres. Recommendation System B: B also includes The Abyss and Little Boy Blue, both of which are quality films. However, the presence of less universally acclaimed films like Breathing Room and Dear Jesse reduces the perceived quality of the list. Winner: A wins.

Explainability/Transparency: 'A wins' - Recommendation System A: The recommendations are relatively explainable given my recent viewing history. Recommendation System B: The recommendations are harder to explain. The Abyss is a good fit, and Little Boy Blue aligns with my interest in drama, but films like Babe, Dear Jesse, and Breathing Room do not clearly connect with my personal information or viewing history. Winner: A wins.

Positive Impact: Recommendation System A: A's list could lead to a more enjoyable and impactful viewing experience. Recommendation System B: B's recommendations could lead to a less positive impact because many of the suggested films do not align with my preferences. Winner: A wins.

Overall Winner: A wins.

Figure 7: An Example of Evaluation Results Provided by GPT-4o

A Case Study

To better understand our evaluation framework, we show a comprehensive and specific evaluation result generated by the large language model. Figure 6 presents a specific example of a prompt used for pair-wise evaluation of recommendation systems. Figure 7 shows the evaluation results returned by a large language model (i.e., GPT-4o) for this prompt. In summary, this example demonstrates that the pairwise evaluation method based on LLM can provide a reasonable and comprehensive evaluation report that includes both quantitative and qualitative analysis. Our codes are publicly available at `https://github.com/anonyProjects/RecSys-Arena`.

B Correlation Analysis

To analyze the correlation between the evaluation results derived from the LLM and the offline metrics, we calculated the Pearson Correlation (Cohen et al., 2009) between the AUC and the $\mathcal{Q}$, as shown in Table 5. Table 5 indicates that the evaluation results generated by GPT-4o and DeepSeek-V2.5 show a strong correlation with the offline metric AUC; however, the P-values are both greater than 0.01, suggesting that the correlation is not significant. In contrast, the evaluation results generated by Llama3.1-8B exhibit a moderate correlation with the offline metric AUC.

Table 5: The Pearson Correlation between the LLM-based pair-wise evaluation results and the AUC

LLM	MovieLens		MIND	
	Correlation Coefficient	P-value	Correlation Coefficient	P-value
GPT-4o	0.8972	0.1027	0.9001	0.0998
DeepSeek-V2.5	0.9436	0.0563	0.9530	0.0469
Llama3.1-8B	-0.2661	0.7338	0.7443	0.2556

C Alignment with User-Centric Explicit Evaluations

We also conducted an experiment on the MovieLens dataset, which includes users' explicit ratings,

Table 6: The performance of LLM-based pair-wise evaluation in terms of explicit feedback-based evaluation on MovieLens.

LLM	RS	explicit nDCG@5	Win(%)	Tie(%)	Lose(%)	Q	Rank
GPT-4o	NRMS	0.2496	61.1	14.7	24.2	1.9485	1
	LightGCN	0.2453	62.8	8.4	28.8	1.9139	2
	SASRec	0.2401	60.3	10.7	29.0	1.7884	3
	DeepFM	0.2387	59.1	11.4	29.5	1.7237	4
DeepSeek-V2.5	NRMS	0.2496	64.7	12.6	22.7	2.1898	1
	LightGCN	0.2453	62.2	9.7	28.1	1.9021	2
	SASRec	0.2401	61.3	9.2	29.5	1.8217	3
	DeepFM	0.2387	64.7	12.6	22.7	1.7680	4

* In the pair-wise evaluation, recommender system A R_A is one of NRMS, LightGCN, SASRec, or DeepFM, while recommender system B R_B is FM.

to investigate whether the evaluation results based on LLMs align with user explicit evaluations. The results in Table 6 demonstrate that the evaluation results from the LLM-based pair-wise evaluation method align with the trends observed in explicit metrics from user feedback. Therefore, the LLM-based pair-wise evaluation method can produce reliable evaluation results.