# HindRec: Aligning User Preferences for Recommendation via Hindsight Fine-tuning

Yawen Zeng[*]
Hunan University
yawenzeng11@gmail.com

Huanwen Wang[*]
Hunan University
huanwenwang@hnu.edu.cn

Lingyu Chen
Nanjing University of Aeronautics and Astronautics
lingyucher@163.com

Wenshu Chen
Monash University
wche0165@student.monash.edu

Ran Chen
Hunan University
ccran@hnu.edu.cn

Hao Chen[†]
Hunan University
chenhao@hnu.edu.cn

## ABSTRACT

Given a user's historical interaction sequence, the recommendation model strives to understand the user's preferences and predict potential candidate items. Presently, the surging popularity of large language models (LLMs) has birthed an array of generative recommendation systems. However, the unfortunate drawback of merging LLMs into recommendation systems is that it cannot capture the true preferences of users (i.e., likes and dislikes).

In this paper, we venture to combine alignment techniques in LLMs to align interest preferences. Specifically, this paper first proposes the application of hindsight fine-tuning in generative recommendation model—referred to as HindRec—which includes three components: prompt construction, recommendation via LLM and hindsight fine-tuning. By constructing training data in the form of hindsight feedback, we fine-tune a LLM via a three-stage strategy to fully utilize positive and negative instances to align user preferences. Wide-ranging experimental corroboration of our HindRec has yielded truly significant outcomes.

## CCS CONCEPTS

• **Information systems → Recommendation Systems**.

## KEYWORDS

Recommendation Systems, Large Language Models, Chain of Hindsight

**ACM Reference Format:**
Yawen Zeng, Huanwen Wang, Lingyu Chen, Wenshu Chen, Ran Chen, and Hao Chen. 2024. HindRec: Aligning User Preferences for Recommendation via Hindsight Fine-tuning. In *Proceedings of the 30th SIGKDD Conference on Knowledge Discovery and Data Mining), August 29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

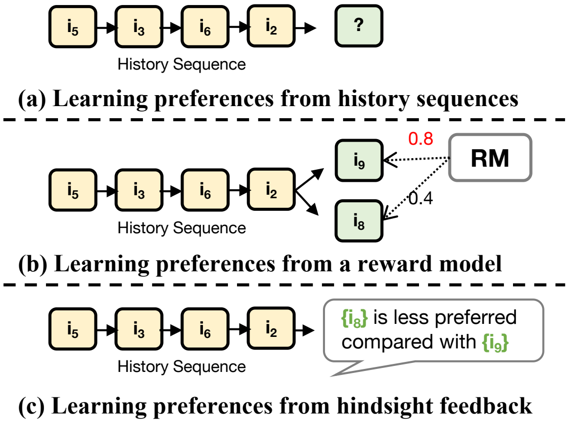
$i_5$
$i_3$
$i_6$
$i_2$
?
History Sequence
(a) Learning preferences from history sequences
$i_9$
0.8
RM
0.4
$i_8$
(b) Learning preferences from a reward model
{$i_8$} is less preferred compared with {$i_9$}
(c) Learning preferences from hindsight feedback

**Figure 1: Illustration of several preference learning strategies, a) from history interaction sequences, b) from a reward model, c) from hindsight feedback.**

## 1 INTRODUCTION

With the growing prominence of applications like TikTok and Temu, recommendation systems have garnered significant attention due to their robust capability to discern user preferences [13, 33]. From a scholarly perspective, taking into account a user's historical interaction sequence, a recommendation model endeavors to comprehend the user's preferences and predict potential candidate items [26–28], as depicted in Fig. 1(a).

The advancement of computer technology has consistently propelled this domain, with every technical innovation infusing it with fresh vigour, especially in aspects such as neural networks, graph learning, attention mechanism, and Transformer [1, 6–8, 12]. Recently, the technological maelstrom predominantly centers around large language models (LLMs), like ChatGPT, LLaMA [24], etc., which boast impressive generative capacities [18, 31, 32], thus instigating the emergence of generative recommendation systems such as Chat-REC [5]. Recommendation systems based on LLMs utilize LLMs to augment data, comprehend item content, and engage users via natural language, amongst other functions [12, 15]. Regrettably, the integration of LLMs into recommendation

[*] Both authors contributed equally to the paper.
[†] Corresponding author.

*KDD '24, August 29, 2024, Barcelona, Spain*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

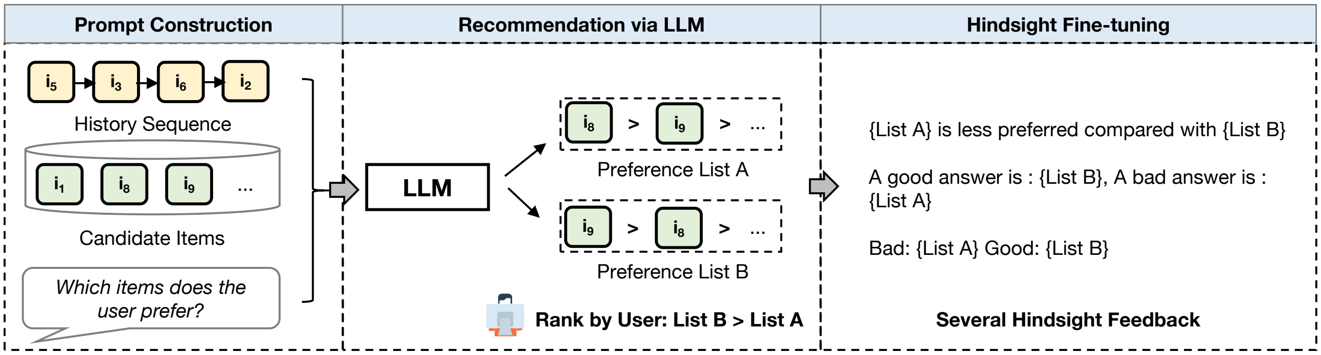
Prompt Construction
Recommendation via LLM
Hindsight Fine-tuning
i5 i3 i6 i2
History Sequence
i1 i8 i9 ...
Candidate Items
Which items does the user prefer?
LLM
i8 > i9 > ...
Preference List A
i9 > i8 > ...
Preference List B
Rank by User: List B > List A
{List A} is less preferred compared with {List B}
A good answer is : {List B}, A bad answer is : {List A}
Bad: {List A} Good: {List B}
Several Hindsight Feedback

**Figure 2: The overview of HindRec for recommendation, which consists of three components: a prompt construction module, a recommendation module via LLM, and a hindsight fine-tuning module.**

systems cannot capture the true preferences of users (i.e., likes and **dislikes**); in other words, LLMs learn possible likes but ignoring dislikes, which are one of the most commonly used negative samples in recommendation systems.

Furthermore, “alignment” may be the most fitting solution (i.e. RLHF) for recommendation systems among LLM-related technologies, evident in preference capture. Alignment techniques focus on align human morality to allow LLMs to generate less harmful responses [34]. In fact, both human ethical preferences and user interest preferences fall under the category of “preferences”. Thus, this technology can be seamlessly utilized to align users‘ “preferences”, especially to distinguish between liked and disliked items. As shown in Fig. 1(b), the trained reward model (RM) can be employed to partially rank potential items. Unfortunately, the exploration and integration of alignment techniques in the realm of recommendation systems still remain untouched by researchers.

Currently, a succession of alignment strategies are being introduced (e.g. PPO [22], DPO [20]), among which, chain of hindsight (CoH) [14, 35] has piqued our interest. Chain of hindsight eliminates the need for training an additional reward model, and instead, leverages the human capacity to derive insights from “ex post facto experiences”. This involves transforming all user feedback into input sequences dedicated to direct fine-tuning, as opposed to restricting it to the learning of partially ordered pairs. As depicted in Fig. 1(c), the model is expected to yield both good and bad outcomes simultaneously to facilitate alignment, thereby ensuring a solid grasp of both positive and negative data. This technique align with the user’s interest preferences more comprehensively from likes and dislikes.

Therefore, this paper introduces for the first time the utilization of various alignment strategies, such as PPO, DPO and CoH, in recommendation systems. Notably, the hindsight fine-tuning approach, which we have named HindRec, demonstrated the highest level of effectiveness. A general overview is illustrated in Fig. 2. Our HindRec comprises three components aptly named: Prompt Construction, Recommendation via LLM, and Hindsight Fine-tuning. Initially, the Prompt Construction integrates the user’s historical sequence, candidate items, and textual prompts, to answer questions like “Which items is the user likely to prefer?”. Subsequently, a LLM module is utilized to execute the recommendation process; by resampling the generated outcomes of the LLM numerous times, a series of ranked lists are derived. These lists are then sorted based on real user interactions. Finally, these sorted lists are transformed into several hindsight feedback forms, which are used to fine-tune the LLM module via a three-stage strategy. The three-stage strategy can gradually enable our model to learn to recommend items, distinguish simple positive-negative pairs, and distinguish hard pairs. Comprehensive experiments on two datasets validate that our proposed HindRec delivers exceptionally noteworthy results.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, this is the first work that incorporates the chain of hindsight of alignment technique to address recommendation systems.
- We customize a generative recommendation model, HindRec, based on the large language model to capture user interest preferences.
- Extensive experiments are conducted on two real-world datasets, which demonstrate the effectiveness of our method.

## 2 OUR PROPOSED FRAMEWORK

The overall framework of HindRec is illustrated in Fig. 2. In general, it comprises three key components: 1) a prompt construction module, which serves to construct a sentence as input to a LLM, such as the historical sequence, candidate items, and textual prompts; and 2) a recommendation module via a LLM, which leverages the benefits of LLMs to offer more suitable items for users; 3) a hindsight fine-tuning module, which aims to train the LLM via the chain of hindsight.

### 2.1 Problem Formulation

Let $u_{(\cdot)}$ denotes a user whose historical interaction sequence is $I = \{i_5, i_3, i_6, i_2, ...\}$, where $i_{(\cdot)}$ represents different items. Formally, the goal of our HindRec is to predict the user’s next interaction based on the historical sequence, as the ground truth is like $i_9$.

### 2.2 Prompt Construction

To enable large language models (i.e. LLaMA) to generate recommended items, a comprehensive prompt $P$ will be constructed in this section. The prompt $P$ will consist of historical sequences $I$, candidate items $C$, and textual prompts, and will be formalized in the following format.

**Task Instruction**: Recommend 5 other items based on user's history from the candidate list, in list form, and the most likely item should be ranked at the top.
**Input**: The user has interacted with the following items in sequence, ("Pink Floyd - The Wall", "Canadian Bacon", ...). The candidate items available for selection are: ("The Blues Brothers", "Platoon", ...). Which items is the user likely to prefer?
**Output List A#**: ["Down by Law", "Almost Famous", "Full Metal Jacket", ...].
**Output List B#**: ["Almost Famous", "Platoon", "Down by Law", ...].
**Output List C#**: ["The Blues Brothers", "Platoon", "Almost Famous", ...].

Among them, all historical items $I$ will be replaced by their item names to enhance semantics. Following PALR [30], we use a small parameter model similar to it as the retrieval module to obtain candidate items $C$, the number of which is $m_1$. Notably, multiple outputs are generated by sampling at $m_2$ different temperatures, as we perform alignment techniques to capture user preferences.

## 2.3 Recommendation via LLM

The prompt $P$ acquired in the preceding section will be fed into LLaMA [24, 25] to anticipate potential items recommendation. In actuality, there are several viable prediction formats available [3, 4, 29], including pointwise for scoring, pairwise for comparison, and listwise for predicting sequence. In this paper, we ultimately opted for the listwise format for two reasons. Firstly, listwise is better suited for the long-range sequence generation of LLMs. Secondly, experimental results in Section 3.3 demonstrate the superiority of this approach over the other two options.

**Input of Pointwise**: ... The candidate items available for selection are: ("The Blues Brothers").
**Output of Pointwise**: Based on the above information, the user is predicted to score 3.

**Input of Pairwise**: ... The candidate items available for selection are: ("The Blues Brothers", "Platoon").
**Output of Pairwise**: The user prefers "The Blues Brothers" over "Platoon".

**Input of Listwise**: ... The candidate items available for selection are: ("The Blues Brothers", "Platoon", "Almost Famous").
**Output of Listwise**: The user's preference sequence is, ["The Blues Brothers", "Platoon", "Almost Famous", ...]

## 2.4 Hindsight Fine-tuning

At present, most of the LLM-based methods only learn the users' likes, but neglect the negative samples of dislikes. In this section, we suggest employing the chain of hindsight [14, 35], an alignment technique that functions as a substitute for reinforcement learning, to align user interest preferences[1].

Specifically, we collect multiple prediction results from the LLaMA (e.g. List A, List B, etc.) at varying temperature coefficients, and generate partial order pairs by comparing them with the ground truth. The List B that exhibits the highest similarity to the actual interaction is deemed as superior data. Using this approach, we developed a range of hindsight data formats for fine-tuning LLaMA, the number of which is $m_3$. Notably, the listwise data in section 2.3, along with the simple and hard samples mentioned above, will undergo training in three stages, progressing from easy to complex.

{List A} is less preferred compared with {List B}
A good answer is : {List B}, A bad answer is : {List A}
Bad: {List A} Good: {List B}

During the training phase, the model is expected to generate both positive and negative outcomes, with only the tokens enclosed within "$\{\cdot\}$" being considered for loss calculation, i.e. the log likelihood of token autoregressively: $\log \mathrm{p}(\mathbf{x}) = \log \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}(\mathrm{x_i}|\mathbf{x}_{<\mathrm{i}})$, where $\mathbf{x} = [x_1, \ldots, x_n]$ represents input tokens. This approach enables the model to encounter numerous negative examples. During the inference phase, the model is only required to produce positive tokens.

# 3 EXPERIMENTS

## 3.1 Experimental Settings

*3.1.1 Datasets.* We experiment with two datasets: Amazon Beauty and Movielens-1M. The Amazon Beauty[2] is a subset of the Amazon review datasets, comprising a comprehensive collection of user-item interactions on Amazon from May 1996 to July 2014. The Movielens-1M dataset[3] is a widely used benchmark dataset, consisting of one million movie ratings.

For dataset preprocessing, we adhere to standard practices. Numeric ratings and reviews are converted to "1", while all other values are converted to "0". Next, we eliminate duplicate interactions for each user and sort their historical items based on the time step of the interaction to generate the user's interaction sequence. In the case of the Beauty dataset, we obtained a total of 20,381 users, 11,145 items, and 196,492 interactions. Similarly, the Movielens-1M dataset yielded 5,892 users, 3,148 items, and 994,259 interactions.

*3.1.2 Evaluation Metric.* We evaluate the performance using two metrics [11]: hit rate (HR) and normalized discounted cumulative gain (NDCG). Among them, HR emphasizes the existence of positive items, while NDCG takes into account the ranking position information.

*3.1.3 Implementation Details.* All experiments are implemented on a server equipped with A100 GPUs. During training, the adam optimizer [10] is used to minimize our loss. Additionally, We employ a leave-one-out approach to assess the effectiveness of our proposed scheme and each baseline. Specifically, for each user, we designate the last interacted item as the test data and the previous historical

[1] The experimental results of PPO and DPO can be found in Section 3.3
[2] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/
[3] http://files.grouplens.org/datasets/movielens/ml-1m.zip

**Table 1: Performance comparison of various baselines on Amazon Beauty dataset.**

| | Amazon Beauty | | | | | |
|---|---|---|---|---|---|---|
| Method | H@3 | H@5 | H@10 | N@3 | N@5 | N@10 |
| BPR-MF | 0.0134 | 0.0232 | 0.0299 | 0.0110 | 0.0188 | 0.0194 |
| NCF | 0.0141 | 0.0246 | 0.0293 | 0.0137 | 0.0192 | 0.0202 |
| GRU4Rec | 0.0120 | 0.0215 | 0.0234 | 0.0105 | 0.0179 | 0.0191 |
| Caser | 0.0147 | 0.0251 | 0.0282 | 0.0125 | 0.0186 | 0.0210 |
| SASRec | 0.0257 | 0.0288 | 0.0417 | 0.0232 | 0.0237 | 0.0283 |
| PALR | 0.0684 | 0.0724 | 0.0781 | 0.0632 | 0.0694 | 0.0726 |
| RecRanker | 0.0715 | 0.0748 | 0.0806 | 0.0696 | 0.0730 | 0.0755 |
| RankVicuna | 0.0823 | 0.0854 | 0.0941 | 0.0748 | 0.0766 | 0.0808 |
| **HindRec** | **0.0985** | **0.1005** | **0.1074** | **0.0913** | **0.0958** | **0.1011** |

**Table 2: Performance comparison of various baselines on Movielens-1M dataset.**

| | Movielens-1M | | | | | |
|---|---|---|---|---|---|---|
| Method | H@3 | H@5 | H@10 | N@3 | N@5 | N@10 |
| BPR-MF | 0.0245 | 0.0401 | 0.0540 | 0.0173 | 0.0245 | 0.0361 |
| NCF | 0.0266 | 0.0437 | 0.0519 | 0.0190 | 0.0252 | 0.0448 |
| GRU4Rec | 0.0141 | 0.0228 | 0.0414 | 0.0106 | 0.0144 | 0.0272 |
| Caser | 0.0283 | 0.0435 | 0.0525 | 0.0208 | 0.0264 | 0.0507 |
| SASRec | 0.0267 | 0.0503 | 0.0568 | 0.0169 | 0.0261 | 0.0496 |
| PALR | 0.0793 | 0.0939 | 0.1014 | 0.0676 | 0.0802 | 0.0976 |
| RecRanker | 0.0899 | 0.1002 | 0.1105 | 0.0783 | 0.0911 | 0.1025 |
| RankVicuna | 0.0923 | 0.1055 | 0.1201 | 0.0855 | 0.0969 | 0.1094 |
| **HindRec** | **0.1049** | **0.1195** | **0.1314** | **0.0983** | **0.1085** | **0.1176** |

items as the validation data, while the remaining items are utilized for model training.

## 3.2 Overall Performance

To compare our solution with competitors, traditional algorithms, and generative methods are considered as baselines. Among them, traditional algorithms, BPR-MF [21], NCF [7], GRU4Rec [8], Caser [23], SASRec [9], are varied techniques for creating personalized recommendations based on different algorithms and methods, like matrix factorization, GRUs, CNN, etc. Meanwhile, generative methods, PALR [30], RecRanker [16], RankVicuna [19] are directly predict items via LLMs.

The results are presented in Table 1 and Table 2, we have the following observations: 1) the performance of generative models via LLM surpasses that of most other methods, thereby demonstrating the superiority of larger models. Additionally, the input format of prompts plays a crucial role, and different strategies can elicit varying abilities of LLMs. 2) the fine-tuned model outperforms direct API calls, and almost all trained models reflect this outcome. This highlights the significance of customized models in comprehending user behavior. 3) our HindRec outperforms all other methods on both datasets, indicating the value of leveraging alignment frameworks and human feedback to capture user preferences.

## 3.3 Ablation Study

We implement ablation study from the following aspects, 1) variants of recommendation strategy; 2) variants of training strategy; 3) variants of LLM module. First of all, "Pointwise", "Pairwise" and "Listwise" denote the three recommendation strategies. As mentioned in Section 2.3, where "Pointwise" for scoring, "Pairwise" for comparison, and "Listwise" for predicting sequence. Thereafter, for training strategy, PPO [17] and DPO [20] are adopted to train our model. Finally, for variants of LLM module, different symbols represent the use of different base models [2, 24, 25] for training.

**Table 3: Ablation study of several variants on Amazon Beauty dataset.**

| | Amazon Beauty | | | | | |
|---|---|---|---|---|---|---|
| Method | H@3 | H@5 | H@10 | N@3 | N@5 | N@10 |
| Pointwise | 0.0753 | 0.0874 | 0.0898 | 0.0734 | 0.0877 | 0.0883 |
| Pairwise | 0.0853 | 0.0921 | 0.0957 | 0.0814 | 0.0903 | 0.0943 |
| Listwise | 0.0985 | 0.1005 | 0.1074 | 0.0913 | 0.0958 | 0.1011 |
| PPO | 0.0811 | 0.0873 | 0.0883 | 0.0775 | 0.0817 | 0.0866 |
| DPO | 0.0910 | 0.971 | 0.0985 | 0.0869 | 0.0912 | 0.0960 |
| Vicuna-7B | 0.0924 | 0.0994 | 0.1002 | 0.0904 | 0.0942 | 0.0985 |
| Qwen-14B | 0.0963 | 0.0997 | 0.1064 | 0.0909 | 0.0945 | 0.1002 |
| LLaMA-13B | 0.0985 | 0.1005 | 0.1074 | 0.0913 | 0.0958 | 0.1011 |

**Table 4: Ablation study of several variants on Movielens-1M dataset.**

| | Movielens-1M | | | | | |
|---|---|---|---|---|---|---|
| Method | H@3 | H@5 | H@10 | N@3 | N@5 | N@10 |
| Pointwise | 0.0852 | 0.0940 | 0.1184 | 0.0842 | 0.0917 | 0.0945 |
| Pairwise | 0.0924 | 0.1053 | 0.1230 | 0.0879 | 0.0965 | 0.1030 |
| Listwise | 0.1049 | 0.1195 | 0.1314 | 0.0983 | 0.1085 | 0.1176 |
| PPO | 0.0880 | 0.0905 | 0.1089 | 0.0813 | 0.0885 | 0.0918 |
| DPO | 0.0983 | 0.1014 | 0.1203 | 0.0922 | 0.1001 | 0.1075 |
| Vicuna-7B | 0.0969 | 0.1035 | 0.1268 | 0.0954 | 0.1043 | 0.1105 |
| Qwen-14B | 0.0953 | 0.1016 | 0.1258 | 0.0944 | 0.1030 | 0.1083 |
| LLaMA-13B | 0.1049 | 0.1195 | 0.1314 | 0.0983 | 0.1085 | 0.1176 |

As presented in Table 3 and Table 4, we have the following observations: 1) "Pointwise" performs worse than "Pairwise", which illustrates the instability of directly predicting scores. And "Pairwise" is not as good as "Listwise", which shows that Listwise may be a more suitable strategy for recommendation systems. 2) After employing PPO and DPO, the model's efficacy has reached a remarkably high level, indicating the affirmative impact of alignment technology on the recommendation system. 3) Different base models have an impact on the final results. LLMs with larger parameters and better understanding capabilities will have better recommendation results under the same amount of training data.

# 4 CONCLUSIONS

In this paper, we aim to introduce the chain of hindsight, an alignment technique that replaces reinforcement learning, into recommendation systems to capture interest preferences. Specifically, we sample the output of the LLM at varying temperatures, construct hindsight training data that includes positive and negative examples, and finally fine-tune the LLM. Our experimental results demonstrate the effectiveness of our HindRec.

In the future, we intend to extend our work by incorporating more human feedback. This is a natural line of exploration: given the efficacy of limited human annotation, what benefits might be derived from gathering a greater volume of human feedback?

## REFERENCES

[1] Shiyu Chen, Yawen Zeng, Da Cao, and Shaofei Lu. 2022. Video-guided machine translation via dual-level back-translation. *Knowledge-Based Systems* 245 (2022), 108598.

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

[3] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM.

[4] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, Brian Chu, Zexi Chen, and Manoj Tiwari. 2023. Leveraging Large Language Models in Conversational Recommender Systems. arXiv:2305.07961 [cs.IR]

[5] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. arXiv:2303.14524 [cs.IR]

[6] Ning Han, Yawen Zeng, Chuhao Shi, Guangyi Xiao, Hao Chen, and Jingjing Chen. 2023. BiC-Net: Learning Efficient Spatio-temporal Relation for Text-Video Retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 3, Article 86 (dec 2023), 21 pages.

[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. arXiv:1708.05031 [cs.IR]

[8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. arXiv:1511.06939 [cs.LG]

[9] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. arXiv:1808.09781 [cs.IR]

[10] P.Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*. ACM.

[11] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD 2020*.

[12] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. arXiv:2305.13731 [cs.IR]

[13] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023. Large Language Models for Generative Recommendation: A Survey and Visionary Discussions. arXiv:2309.01157 [cs.IR]

[14] Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023. Chain of Hindsight Aligns Language Models with Feedback. arXiv:2302.02676 [cs.LG]

[15] Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. arXiv:2304.10149 [cs.IR]

[16] Sichun Luo, Bowei He, Haohan Zhao, Yinya Huang, Aojun Zhou, Zongpeng Li, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2024. RecRanker: Instruction Tuning Large Language Model as Ranker for Top-k Recommendation. arXiv:2312.16018 [cs.IR]

[17] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL]

[18] Keyu Pan and Yawen Zeng. 2023. Do LLMs Possess a Personality? Making the MBTI Test an Amazing Evaluation for Large Language Models. arXiv:2307.16180 [cs.CL]

[19] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. arXiv:2309.15088 [cs.IR]

[20] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290 [cs.LG]

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. arXiv:1205.2618 [cs.IR]

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347* (2017).

[23] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. arXiv:1809.07426 [cs.IR]

[24] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[25] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]

[26] Huanwen Wang, Yawen Zeng, Jianguo Chen, Ning Han, and Hao Chen. 2023. Interval-enhanced Graph Transformer solution for session-based recommendation. *Expert Systems with Applications* 213 (2023), 118970.

[27] Huanwen Wang, Yawen Zeng, Jianguo Chen, Zhouting Zhao, and Hao Chen. 2022. A Spatiotemporal Graph Neural Network for session-based recommendation. *Expert Systems with Applications* 202 (2022), 117114.

[28] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative Recommendation: Towards Next-generation Recommender Paradigm. arXiv:2304.03516 [cs.IR]

[29] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. RecMind: Large Language Model Powered Agent For Recommendation. arXiv:2308.14296 [cs.IR]

[30] Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. 2023. PALR: Personalization Aware LLMs for Recommendation. arXiv:2305.07622 [cs.IR]

[31] Yawen Zeng. 2022. Point Prompt Tuning for Temporally Language Grounding. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2003–2007.

[32] Yawen Zeng, Da Cao, Xiaochi Wei, Meng Liu, Zhou Zhao, and Zheng Qin. 2021. Multi-Modal Relational Graph for Cross-Modal Video Moment Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2215–2224.

[33] Yawen Zeng, Qin Jin, Tengfei Bao, and Wenfeng Li. 2023. Multi-Modal Knowledge Hypergraph for Diverse Image Retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 3 (Jun. 2023), 3376–3383.

[34] Yawen Zeng, Keyu Pan, and Ning Han. 2023. RewardTLG: Learning to Temporally Language Grounding from Flexible Reward. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 2344–2348.

[35] Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E. Gonzalez. 2023. The Wisdom of Hindsight Makes Language Models Better Instruction Followers. arXiv:2302.05206 [cs.CL]