

Not Just What, But When: Integrating Irregular Intervals to LLM for Sequential Recommendation

Wei-Wei Du
Sony Group Corporation
Tokyo, Japan
weiwei.du@sony.com

Takuma Udagawa
Sony Group Corporation
Tokyo, Japan
takuma.udagawa@sony.com

Kei Tateno
Sony Group Corporation
Tokyo, Japan
kei.tateno@sony.com

Abstract

Time intervals between purchasing items are a crucial factor in sequential recommendation tasks, whereas existing approaches focus on item sequences and often overlook by assuming the intervals between items are static. However, dynamic intervals serve as a dimension that describes user profiling on not only the history within a user but also different users with the same item history. In this work, we propose **IntervalLLM**, a novel framework that integrates interval information into LLM and incorporates the novel interval-infused attention to jointly consider information of items and intervals. Furthermore, unlike prior studies that address the cold-start scenario only from the perspectives of users and items, we introduce a new viewpoint: *the interval perspective* to serve as an additional metric for evaluating recommendation methods on the warm and cold scenarios. Extensive experiments on 3 benchmarks with both traditional- and LLM-based baselines demonstrate that our IntervalLLM achieves not only 4.4% improvements in average but also the best-performing warm and cold scenarios across all users, items, and the proposed interval perspectives. In addition, we observe that the cold scenario from the interval perspective experiences the most significant performance drop among all recommendation methods. This finding underscores the necessity of further research on interval-based cold challenges and our integration of interval information in the realm of sequential recommendation tasks. Our code is available here: <https://github.com/sony/ds-research-code/tree/master/recsys25-IntervalLLM>.

CCS Concepts

• **Information systems** → **Data mining**; • **Computing methodologies** → **Artificial intelligence**; • **Applied computing** → **Online shopping**.

Keywords

Irregular Interval, Sequential Recommendation, User Representation Learning

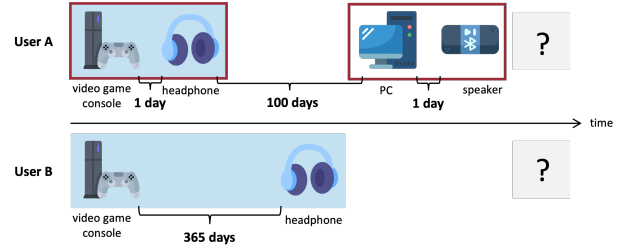


Figure 1: Examples of purchase histories for two users.

ACM Reference Format:

Wei-Wei Du, Takuma Udagawa, and Kei Tateno. 2025. Not Just What, But When: Integrating Irregular Intervals to LLM for Sequential Recommendation. In *Proceedings of August 04, 2025 (GenAIRecP@KDD 2025)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/xxxxxxx.xxxxxxx>

1 Introduction

In sequential recommendation, next-item prediction aims to forecast the next item a user is likely to select based on the past behavior. Recent state-of-the-art models have increasingly adopted LLM-based approaches [16, 17], which not only enhance textual understanding but also leverage reasoning ability and world knowledge. Prior to the emergence of LLMs, several sequential recommendation models [5, 15, 19] incorporated interval information to capture the temporal dynamics of user behavior. With the advent of powerful LLMs, some studies [3, 11, 26] integrated timestamps into LLM-based methods for time-series tasks, but none of these studies specifically focus on sequential recommendation.

As shown in Figure 1, existing models often consider the purchase item history as a sequence for sequential recommendation. They neglect the influence of temporal intervals in shaping user behavior, as depicted by the red squares. Even with identical item sequences, varying intervals (highlighted in blue for User A and User B) reveal differing item influences. In this work, we take the first step toward integrating interval information into LLM-based recommender systems, bridging the gap between sequential recommendation and the interval modeling capabilities of LLMs.

Furthermore, as the number of users and items increases, user-item interaction data becomes increasingly sparse, making it challenging to model user and item behaviors with limited interactions. This issue, commonly referred to as the cold-start problem, is frequently discussed in recommendation system research [23, 25]. However, in real-world recommendation systems, where interaction data accumulate over long periods, the cold-start problem from

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GenAIRecP@KDD 2025, Toronto, ON, Canada,

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM

<https://doi.org/xxxxxxx.xxxxxxx>

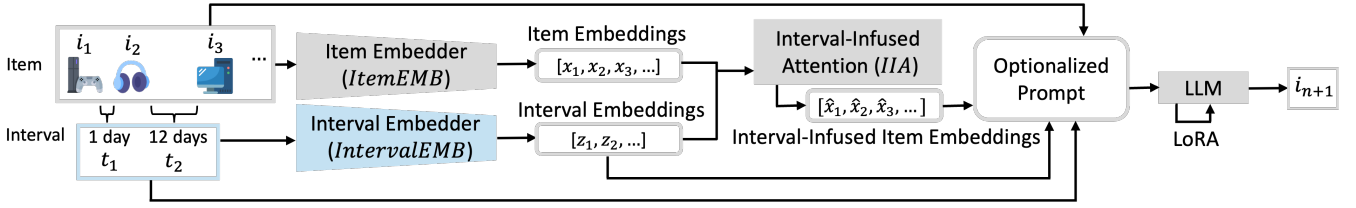


Figure 2: The proposed IntervalLLM. All LLM parameters except for those in LoRA are frozen.

a time-interval perspective, which refers to users whose interactions are characterized by relatively long time gaps, is another critical yet often overlooked issue. User preferences evolve over time, and item trends change dynamically. The temporal aspect of item sequences plays a significant role in distinguishing recent purchasing behaviors from historical ones. To address this gap, we aim to benchmark existing sequential recommendation methods not only for cold-start users and items but also from an interval cold-start perspective. Furthermore, we explore how leveraging LLM knowledge and interval-based information can enhance performance in scenarios where traditional methods struggle.

Our contributions are summarized as follows:

- To the best of our knowledge, IntervalLLM is the first work to integrate time intervals into LLMs for sequential recommendation, enabling the model to capture user behavior based on the intervals between consecutive items.
- Beyond separately incorporating item and interval embeddings, we propose Interval-Infused Attention to capture the temporal relevance between items and intervals.
- We introduce a novel perspective on the cold-start problem by considering time intervals and reveal that existing methods suffer from significant performance drops in interval cold-start scenarios. Experiments demonstrate that IntervalLLM achieves the best performance in overall, warm, and even cold-start scenarios.

2 Preliminaries

2.1 Related Work

Recent advancements in LLMs have demonstrated their remarkable effectiveness due to their extensive world knowledge and strong generalization capabilities. In the context of sequential recommendation tasks, two primary strategies have emerged for leveraging LLMs. The first treats LLMs as feature extractors, using their embeddings to initialize existing recommendation models [7, 22]. The second involves directly fine-tuning LLMs as recommenders, leveraging their vast pre-trained knowledge and advanced reasoning abilities [1, 2, 4, 7]. To further enrich recommendation representations, models such as MoRec [24], A-LLMRec [13], and LLM-SRec [14] incorporate user embeddings extracted from pre-trained collaborative filtering models. However, these approaches exclude timestamp or interval information, thereby neglecting the temporal dimension inherent in sequential data. To address this gap, we introduce a novel cold-start benchmark from the interval perspective and investigate effective strategies for integrating interval information into LLMs to enhance sequential recommendation performance.

2.2 Sequential Recommendation Task

In the context of sequential recommendation, a user interacts chronologically with a sequence of n item names $[i_1, i_2, \dots, i_n]$ and corresponding intervals $[t_1, t_2, \dots, t_{n-1}]$, where each t_k represents the time difference between consecutive items. To align with the generative capabilities of LLMs, a set of candidate items $[c_1, c_2, \dots, c_j]$ is randomly sampled from the item pool for each user. The goal of the sequential recommendation task can be formulated as:

$$i_{n+1} = f([i_1, i_2, \dots, i_n], [t_1, t_2, \dots, t_{n-1}], [c_1, c_2, \dots, c_j]) \quad (1)$$

where i_{n+1} denotes the next item to be recommended at the $(n+1)$ -th timestamp, and f represents the recommendation algorithm. Notably, the number of items in the sequence exceeds the number of time intervals by one, as each t_k captures the temporal gap between consecutive items i_k and i_{k+1} .

3 Methodology

The framework of IntervalLLM is illustrated in Figure 2, which first maps items and intervals into a shared latent space, then employs an interval-infused attention mechanism to model their interactions. Finally, we propose a novel optionalized prompt and leverage the LLM as the recommender.

3.1 Behavior Representation Learning

Item Embedder. To apply LLMs as recommenders, following the previous work [16], an item embedder is required to map each item into the language space, transforming the textual natural language of item names into embeddings. Each item name associated with a user is converted into an embedding x , denoted as follows:

$$x_k = \text{ItemEMB}(i_k), \quad (2)$$

where $\text{ItemEMB}()$ represents the tokenizer and embedding layer. Here, k denotes the k -th item in the sequence.

Interval Embedder. Previous work [6] has proposed some regularization methods for processing numerical inputs. In sequential recommendation, the scale of the interval is particularly crucial. Therefore, we propose an interval embedder to encode the temporal scale z as follows:

$$z_{k-1} = \text{IntervalEMB}(t_{k-1}), \quad (3)$$

where $\text{IntervalEMB}()$ denotes the embedding layer for the interval, which we implement as a simple MLP layer in our experiments.

Interval-Infused Attention (IIA). To jointly model the relations between interval and items, an intuitive method is to directly add interval embeddings into the embedding sequence produced by the LLM. However, this fails to effectively integrate interval

Optionalized Next Item Recommendation Prompt

Input: This user has purchased: i_1 [ITEM] \hat{x}_1 [/ITEM], and after t_1 [INTERVAL] \hat{z}_1 [/INTERVAL] days purchased i_2 [ITEM] \hat{x}_2 [/ITEM], and after t_2 [INTERVAL] \hat{z}_2 [/INTERVAL] days purchased i_3 [ITEM] \hat{x}_3 [/ITEM]... . Based on this history, recommend the next product that the user is most likely to purchase from the following twenty game options: A: c_1 \n B: c_2 \n C: c_3 \n ... \n T: c_{20} \n The answer with the option's letter only is

Output: B

Figure 3: The optionalized prompt template for the next item recommendation task. i represents an item, t denotes the corresponding interval, c is the candidate item, \hat{x} is the attention weight, and \hat{z} is the interval embedding. The special tokens [ITEM] and [/ITEM] mark the location of attention which is used for item embedding, while [INTERVAL] and [/INTERVAL] specify the position of the interval embedding.

information into item embeddings. Therefore, we applied scaled dot-product attention [21] to separately characterize the importance of both items and intervals and then aggregate corresponding weights as interval-infused item embeddings. Given a user as an example, we represent the item sequence with item textual embeddings $X = (x_1, \dots, x_n)$ and the corresponding interval embeddings $Z = (0, z_1, \dots, z_{n-1})$. To ensure alignment between the two sequences, a zero vector is padded at the beginning of the interval embeddings, making the lengths of X and Z identical. The formula of the interval-infused attention $IIA()$ is derived as follows:

$$Q_z = ZW^{Q_z}, K_x = XW^{K_x}, V_x = XW^{V_x}, \quad (4)$$

$$IIA(X, Z) = \text{softmax}\left(\frac{Q_z K_x^T}{\sqrt{d_q}} + M\right)V_x, \quad (5)$$

where Q_z is the query matrix of Z , K_x and V_x denote the key and value matrices of X , and M denotes the casual attention matrix to prevent information leakage from future tokens. The projection matrices $W^{Q_z}, W^{K_x}, W^{V_x} \in \mathbb{R}^{d_{\text{lm}} \times d_q}$ are learnable parameters.

To consider user behaviors from different perspectives, we extend interval-infused attention with h heads to learn interval-infused item embeddings:

$$[\hat{x}_1, \dots, \hat{x}_n] = \text{Concat}(IIA_1, \dots, IIA_h)W^O, \quad (6)$$

where $W^O \in \mathbb{R}^{hn \cdot d_{\text{lm}} \times n \cdot d_{\text{lm}}}$ is a linear layer.

Optionalized Prompt. To structure the full sequence in a format compatible with LLM, we design a text prompt that effectively conveys the relevant information for LLM instruction tuning. Most existing works consider sequential recommendation as a generation task, evaluating whether the ground truth is included in the generated sequence [16, 18]. However, since the ground truth is often an item name that may be a full sentence rather than a single word, evaluating the generation task becomes more challenging. For example, there are cases where the ground truth appears within the generated output but is accompanied by other item names, leading to ambiguity. Additionally, the ground truth may be present in the generated output but with spelling errors or incorrect details, further complicating evaluation. To address these challenges, we propose an option-based generation task, where alphabetic options (e.g., A, B, etc.) are added before each candidate item name as shown in Figure 3. This optional problem setting significantly reduces ambiguity and enables a more reliable evaluation of model

Table 1: Dataset characteristics. Density is defined as the number of interactions per user (#Interactions/#Users).

Dataset	#User	#Item	#Interaction	Density
Video Games	94,762	25,612	814,586	8.59
CDs and Vinyl	123,876	89,370	1,552,764	12.53
Books	776,370	495,063	9,488,297	12.22

performance. In summary, the prompt includes task definition, user behavior sequence incorporating both items and intervals, and candidate options. To align interval embeddings and interval-infused item embeddings with LLM (i.e., d_{lm}), all textual components are first processed through the initial layer of the LLM and then concatenated with embeddings.

Instruction Fine-tuning by LoRA. To mitigate the gap between the capabilities of LLMs in general text generation and sequential recommendation tasks, LoRA [10] is applied to fine-tune LLMs for sequential recommendation tasks. The loss function of IntervallLM is formulated as:

$$L((i_{1..n}, t_{1..n-1}), i_{n+1}) = -\log(P_\theta(i_{n+1} | (i_{1..n}, t_{1..n-1}))), \quad (7)$$

where θ includes the parameters of the item embedder, the interval embedder, the interval-infused attention and the LoRA within the LLM's parameters.

4 Experiments

In this section, we conduct comprehensive experiments to answer the following key research questions:

- RQ1: How does IntervallLM performs in three sequential recommendation datasets?
- RQ2: How do different components of IntervallLM affect model performance?
- RQ3: How does IntervallLM perform in the cold scenario from various perspectives?

4.1 Experimental Setup

Datasets. We conduct experiments on three Amazon Reviews datasets [9]: Video Games, CDs and Vinyl, and Books. The characteristics of each dataset are summarized in Table 1. Following prior

Table 2: Overall performance with Hit Rate@1 (\uparrow) on three datasets. The best result in each column is in boldface, while the second-best result is underlined. * means that some of the predictions are not valid (e.g., misspelling).

Category	Method	Video Games	CDs and Vinyl	Books
2*Traditional	GRU4Rec	49.0%	46.5%	35.7%
	SASRec	50.8%	50.9%	38.0%
Traditional + Interval	TiSASRec	52.9%	<u>54.1%</u>	58.6%
2*LLM-based	LLaMA	56.0%	45.2%*	60.2%
	LLaRA	50.5%	49.6%*	60.0%*
2*LLM-based + Interval	LLaMA + Interval	<u>56.3%</u>	48.7%	<u>61.1%</u>
	IntervallLM (Ours)	61.7%	55.4%	61.9%

Table 3: Ablation study of IntervallLM on the Video Games dataset. Δ indicates applying timestamp as the text prompt. *IntervalEMB* indicates adding the interval embedding. *IIA* indicates adding the interval-infused embeddings.

LLaMA	Interval	<i>IntervalEMB</i>	<i>IIA</i>	Hit Rate@1 (\uparrow)
✓	✗	✗	✗	56.0%
✓	Δ	✗	✗	54.2%
✓	✓	✗	✗	56.3%
✓	✓	✓	✗	56.8%
✓	✓	✓	✓	61.7%

work [12], we use five-core datasets in which both users and items have at least five interactions each.

Implementation Details. All LLM-based methods are trained for up to 5 epochs with a batch size of 64 and 20 candidates [16], using LLaMA-2 (7B) as the backbone with an optionalized prompt. For IntervallLM, the LLM input dimension (d_{llm}) is 4096, embedding dimension (d_q) is 256, and the number of attention heads (h) is 2.

Evaluation Metric. We follow previous work [16] to use Hit Ratio@1. Notably, [16] further adds the validation ratio as an additional metric, as certain models may generate invalid responses, such as word mismatches with the candidate set; however, in our method, the validation ratio is consistently 100% attributed to our optionalized prompt design. We adopt a leave-one-out evaluation strategy following previous work [12], where the last item in each user’s sequence is reserved for testing, the second-to-last item for validation, and the remaining items are used for training.

Baselines. We compare four groups of models as our baselines:

- Traditional methods: GRU4Rec [8] and SASRec [12].
- Traditional method with interval: TiSASRec [15].
- LLM-based methods: LLaMA [20] and LLaRA [16].
- LLM-based method with interval: LLaMA + Interval. As no existing method incorporates interval information into LLMs, we build a naive baseline by directly adding interval information into the prompt for LLaMA.

4.2 RQ1: Overall Performance Comparison

Table 2 shows the recommendation performance of IntervallLM compared with traditional sequential recommendation models and

LLM approaches across three datasets, which draws the following observations: (1) LLM-based methods outperform traditional models, demonstrating that leveraging pre-trained LLM knowledge can enhance recommendation performance. This improvement is particularly notable in Books, where item names often carry rich semantic meaning, leading to substantial performance gains. (2) LLaMA + Interval surpasses LLaMA across all three datasets, showcasing the effectiveness of adding interval information. (3) IntervallLM consistently achieves superior performance compared to all other methods on all datasets, highlighting not only the importance of incorporating interval information into LLMs but also the efficacy of the proposed interval-infused attention. Overall, these findings emphasize the effectiveness of utilizing LLMs to capture the semantic meaning of items, incorporating interval information, designing attention to leverage item and interval, and employing optionalized prompts tailored for the recommendation task.

4.3 RQ2: Ablation Study

An extensive ablation study is conducted to validate the design of IntervallLM, as presented in Table 3. The comparison between the first row and second as well as third rows reveals that directly representing timestamp as the text hinders model performance, while using interval text helps understand temporal relations between items. Furthermore, adding the interval embeddings from the interval embedder enables the model to learn the scale of intervals more effectively (row 4). The boosted performance in row 5 indicates applying interval-infused attention enhances the model’s ability to capture relations between items and their corresponding intervals. In summary, these results suggest that all proposed components contribute positively to the overall performance.

4.4 RQ3: Study on Warm and Cold Scenarios

Most previous studies have focused on warm and cold scenarios for user or/and item [13, 14]. However, interval provides a perspective different from users and items, for evaluating whether interactions occur frequently within a short period. The interval-based viewpoint can reflect the activity level of each user.

Following the setup of A-LLMRec [13], a user or item is categorized as "Warm" if it falls within the top 35% of interactions and as "Cold" if it falls within the bottom 35%. For intervals, we first compute the average interval between interactions for each user. If a user’s average interval falls within the top 35% of interactions,

Table 4: Performance on Warm/Cold scenarios evaluated by Hit Rate@1 (\uparrow) on the Video Games dataset. *Warm* represents the warm scenario, *Cold* represents the cold-start scenario, and *Diff.* = $((Cold - Warm)/Warm)$.

Method	User Perspective			Item Perspective			Interval Perspective			Overall (\uparrow)
	<i>Warm</i> (\uparrow)	<i>Cold</i> (\uparrow)	<i>Diff.</i>	<i>Warm</i> (\uparrow)	<i>Cold</i> (\uparrow)	<i>Diff.</i>	<i>Warm</i> (\uparrow)	<i>Cold</i> (\uparrow)	<i>Diff.</i>	
GRU4Rec	49.3%	48.8%	-1.0%	53.3%	45.1%	-15.4%	55.2%	43.7%	-20.8%	49.0%
SASRec	52.2%	50.1%	-4.0%	54.7%	47.4%	-13.3%	57.4%	45.2%	-21.2%	50.8%
TiSASRec	55.0%	52.3%	-4.9%	54.3%	51.0%	-6.1%	54.6%	49.1%	-10.1%	52.9%
LLaMA	56.5%	55.8%	-1.2%	58.8%	53.7%	-8.7%	61.1%	51.8%	-15.2%	56.0%
LLaRA	51.6%	50.0%	-3.1%	54.0%	46.9%	-13.1%	60.3%	43.3%	-28.0%	50.5%
LLaMA + Interval	56.1%	56.2%	+0.2%	59.1%	53.8%	-9.0%	60.6%	53.3%	-12.0%	56.3%
IntervalLLM (Ours)	61.6%	61.8%	+0.3%	64.4%	59.5%	-7.6%	65.6%	59.1%	-9.9%	61.7%

they are classified as "Warm", and those in the bottom 35% as "Cold". After training the model on the full training dataset, we distinguish warm and cold scenarios based on this categorization.

4.4.1 Superiority of IntervalLLM. IntervalLLM outperforms all baselines across all warm and cold scenarios, and all perspectives as shown in Table 4. This result highlights that integrating intervals and leveraging their relationships with items enables the LLM to better capture temporal knowledge. Consequently, our approach not only alleviates the cold-start issue from the interval perspective but also improves performance in user and item cold-start scenarios.

4.4.2 Performance Drop Analysis. *Diff.* is used as an additional metric to evaluate the performance drop between warm and cold scenarios. When comparing methods that incorporate interval information (TiSASRec, LLaMA + Interval, and IntervalLLM) with those that do not, the smaller performance drops in the interval perspective highlight the advantage of leveraging interval information to mitigate the cold-start issue. Overall, IntervalLLM exhibits the smallest performance drop across both the user and interval perspectives, demonstrating its effectiveness in addressing the cold-start problem.

4.4.3 Performance Drop in the Interval Perspective is more Larger than in User and Item Perspectives. A comparative analysis of intervals, users, and items reveals that the interval perspective experiences the most significant performance drop across all methods. This finding suggests that the interval aspect has been largely overlooked in prior research. Our study underscores the importance of addressing the cold-start issue specifically from the interval perspective.

5 Conclusion

This paper introduces a novel perspective by benchmarking the cold-start scenario from the interval viewpoint, going beyond traditional user and item dimensions, and highlighting a significant performance drop in previous works. Our proposed IntervalLLM not only encodes dynamic intervals, but also learns the extent to which interval influences item recommendations through interval-infused attention. The experiments show that empowering LLM by interval yields effectiveness improvements for recommendation models across various benchmarks. In addition, we analyze the

performance from the cold scenario of user, item, and interval, respectively, to identify both well-supported and underperforming user groups, which consistently show the superior performance of IntervalLLM in both warm and cold scenarios. For future research, we plan to explore other possibilities of integrating interval into LLM and further solving the cold-start interval issue.

References

- [1] Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Fuli Feng, Xiangnan He, and Qi Tian. 2023. A Bi-Step Grounding Paradigm for Large Language Models in Recommendation Systems. *CoRR* abs/2308.08434 (2023).
- [2] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *RecSys*. ACM, 1007–1014.
- [3] Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2025. LLM4TS: Aligning Pre-Trained LLMs as Data-Efficient Time-Series Forecasters. *ACM Trans. Intell. Syst. Technol.* (Feb. 2025). <https://doi.org/10.1145/3719207>
- [4] Zheng Chen. 2023. PALR: Personalization Aware LLMs for Recommendation. *CoRR* abs/2305.07622 (2023).
- [5] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. 2023. Uniform Sequence Better: Time Interval Aware Data Augmentation for Sequential Recommendation. In *AAAI*. AAAI Press, 4225–4232.
- [6] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large Language Models Are Zero-Shot Time Series Forecasters. *CoRR* abs/2310.07820 (2023).
- [7] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging Large Language Models for Sequential Recommendation. In *RecSys*. ACM, 1096–1102.
- [8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. *arXiv:1511.06939* [cs.LG] <https://arxiv.org/abs/1511.06939>
- [9] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiuxi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).
- [10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*. OpenReview.net.
- [11] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *ICLR*. OpenReview.net.
- [12] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. *CoRR* abs/1808.09781 (2018). [arXiv:1808.09781](https://arxiv.org/abs/1808.09781)
- [13] Sein Kim, Hongseok Kang, Seungyeon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1395–1406.
- [14] Sein Kim, Hongseok Kang, Kibum Kim, Jiwan Kim, Donghyun Kim, Minchul Yang, Kwangjin Oh, Julian McAuley, and Chanyoung Park. 2025. Lost in Sequence: Do Large Language Models Understand Sequential Recommendation? *arXiv:2502.13909* [cs.LG] <https://arxiv.org/abs/2502.13909>

- [15] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM*. ACM, 322–330.
- [16] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. LLaRA: Large Language-Recommendation Assistant. arXiv:2312.02445 [cs.LG] <https://arxiv.org/abs/2312.02445>
- [17] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Hui Feng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *CoRR* abs/2306.05817 (2023).
- [18] Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024. LLM-ESR: Large Language Models Enhancement for Long-tailed Sequential Recommendation. In *NeurIPS*.
- [19] Jinseok Seol, Youngrok Ko, and Sang-goo Lee. 2022. Exploiting Session Information in BERT-based Session-aware Sequential Recommendation. *CoRR* abs/2204.10851 (2022).
- [20] 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, Madsen 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, Rishi 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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* abs/2307.09288 (2023).
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017).
- [22] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. *CoRR* abs/2104.07413 (2021).
- [23] Hongli Yuan and Alexander A. Hernandez. 2023. User Cold Start Problem in Recommendation Systems: A Systematic Review. *IEEE Access* 11 (2023), 136958–136977.
- [24] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id-vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2639–2649.
- [25] Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, Feiran Huang, Sheng Zhou, Jiajun Bu, Allen Lin, James Caverlee, Fakhri Karray, Irwin King, and Philip S. Yu. 2025. Cold-Start Recommendation towards the Era of Large Language Models (LLMs): A Comprehensive Survey and Roadmap. *CoRR* abs/2501.01945 (2025).
- [26] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *NeurIPS*.