In-context Learning for Addressing User Cold-start in Sequential Movie Recommenders

Xurong Liang
jxliang@amazon.com
Amazon Machine Learning
Brisbane, Australia

Vu Nguyen
vutngn@amazon.com
Amazon Machine Learning
Adelaide, Australia

Vuong Le
levuong@amazon.com
Amazon Machine Learning
Melbourne, Australia

Paul Albert
albrtpa@amazon.com
Amazon Machine Learning
Melbourne, Australia

Julien Monteil
jul@amazon.com
Amazon Machine Learning
Brisbane, Australia

ABSTRACT

The user cold-start problem remains a fundamental challenge for sequential recommender systems, particularly in large-scale video streaming services where a substantial portion of users have limited or no historical interaction data. In this work, we formulate an attempt at solving this issue by proposing a framework that leverages Large Language Models (LLMs) to enrich interaction histories using user metadata. Our approach generates a set of imaginary video items relevant to a given user's demographic, represented through structured item key-value attributes. The generated items are inserted into users' interaction sequences using early or late fusion strategies. We find that the generated user histories enable better initial user profiling for absolute cold users and enhanced preference modeling for nearly cold users. Experimental results on the public ML-1M dataset and an internal dataset from an Amazon streaming service demonstrate the effectiveness of our LLM-based augmentation method in mitigating cold-start challenges.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Movie Recommender System; Large Language Models

ACM Reference Format:
Xurong Liang, Vu Nguyen, Vuong Le, Paul Albert, and Julien Monteil. 2025. In-context Learning for Addressing User Cold-start in Sequential Movie Recommenders. In *Proceedings of the 19th ACM Conference on Recommender Systems (RecSys '25), Sep 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 4 pages. https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1 INTRODUCTION

Recommender systems are essential for delivering personalized content on video streaming services, enhancing user experience, and driving revenue. On the Amazon streaming service, for which we are researching and developing recommender systems, users with no watch history (absolute cold users) and those with only 1–5 watched videos (nearly cold users) account for a significant percentage of the user base. Due to these limited or non-existent historical data, traditional modeling systems struggle to represent these users accurately, resulting in suboptimal recommendations for a large majority of the user base.

With the rapid advancement of large language models (LLMs), recent work has explored their potential to address recommendation challenges by leveraging their language understanding capabilities [1, 6, 12, 15, 16]. For video recommendation, most recommender systems [5, 9, 11, 17] utilize item feature attributes (*i.e.*, item metadata) to describe historical interactions. Users' demographic-related attributes (*i.e.*, user metadata) are rarely exploited in the literature for inferring user preferences, due to the low availability of such data and to the low-level context it provides, which makes it challenging to deduce user preferences. Although these demographic attributes cannot identify individuals, they can reveal group-level preference trends valuable for recommendation, particularly in user cold-start scenarios where interaction data is sparse. Large Language Models (LLMs), trained on vast text corpora, implicitly encode extensive world knowledge, including correlations between described characteristics and associated interests or behaviors. Building on demonstrations of LLM capabilities for text-based sequential recommendation [2, 9] and zero-shot item cold-start [7], we propose to leverage the LLM embedded knowledge to tackle the *user* cold-start problem. Specifically, we investigate whether LLMs can interpret user metadata to predict relevant video items for new or near-new users, thereby enriching their initial interaction profiles and improving the accuracy of personalized recommendations.

This paper proposes an LLM-based imaginary item generator to mitigate the user cold-start problem in video recommendation. Given user metadata, the model generates a set of relevant imaginary video items, each described by standardized key-value attributes. We introduce three sampling strategies to guide the LLM using examples of first-user interactions. The generated items are inserted into the historical sequences of absolute and nearly cold users, effectively converting metadata into interaction signals. This enables initial user profiling for absolute cold users and enhances representation for nearly cold users. Two simple insertion strategies are discussed to integrate these items into sequential recommender

RecSys '25, Sep 22–26, 2025, Prague, Czech Republic

ACM ISBN xxxxx
https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

systems. We demonstrate the effectiveness of our approach on both the public ML-1M dataset[1] and an internal dataset from an Amazon streaming service.

2 METHOD

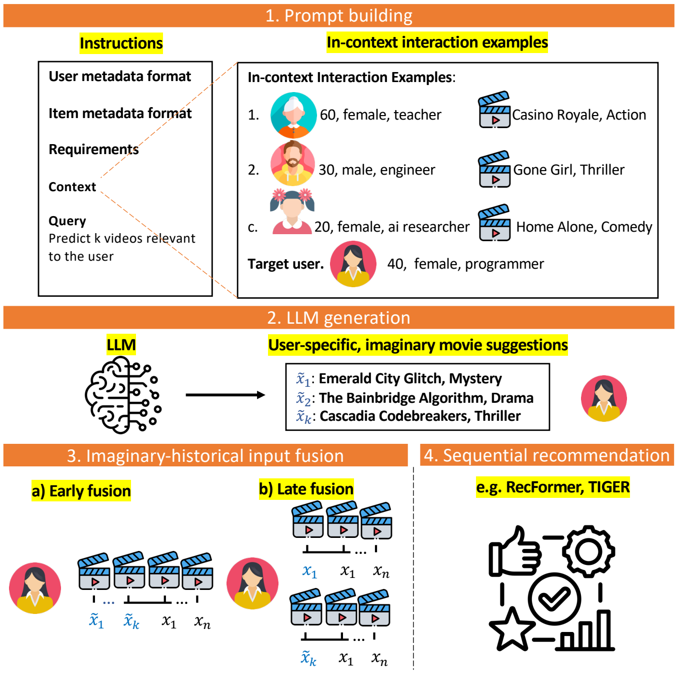
1. Prompt building
Instructions
In-context interaction examples
User metadata format
Item metadata format
Requirements
Context
Query
Predict k videos relevant to the user
In-context Interaction Examples:
1. 60, female, teacher
Casino Royale, Action
2. 30, male, engineer
Gone Girl, Thriller
c. 20, female, ai researcher
Home Alone, Comedy
Target user. 40, female, programmer
2. LLM generation
LLM
User-specific, imaginary movie suggestions
$\tilde{x}_1$: Emerald City Glitch, Mystery
$\tilde{x}_2$: The Bainbridge Algorithm, Drama
$\tilde{x}_k$: Cascadia Codebreakers, Thriller
3. Imaginary-historical input fusion
a) Early fusion
$\tilde{x}_1$ ... $\tilde{x}_k$ x_1 ... x_n
b) Late fusion
$\tilde{x}_1$ x_1 ... x_n
$\tilde{x}_k$ x_1 ... x_n
4. Sequential recommendation
e.g. RecFormer, TIGER

Figure 1: The proposed framework. We first design a comprehensive prompt, including selected in-context examples and target user metadata. The LLM then produces k user-specific imaginary video items that we insert into the user history. The augmented interaction sequence is then given to sequential recommenders to predict the next interaction.

This section presents our 4-step recommendation framework, illustrated in Fig 1. The framework receives as input a target user-metadata m with optional historical interactions $\mathbf{X} = \{x_1, ..., x_n\}$. It then forms the in-context prompt q to query the LLM to gather imaginary interactions $LLM : q \mapsto \tilde{\mathbf{X}} = \{\tilde{x}_1, ..., \tilde{x}_k\}$. $\mathbf{X}$ and $\tilde{\mathbf{X}}$ are then jointly fed into a backbone sequential recommender for the recommended item y.

1. Prompt building. We aim to build an in-context prompt that guides the LLM to generate possible relevant items given the target user's metadata. The Prompt commences with the description of the required key-value JSON format for user metadata and item features. It also provides a list of generic requirements of the task, including the length limit of the generated textual features and options for categorical features (such as genre).

The Prompt then continues with the Context section, which consists of c pairs of user metadata-relevant item examples $\mathbf{C} = \{(m^1, x_1^1), (m^2, x_1^2), ...(m^c, x_1^c)\}$. In forming the Context, we employ three strategies to select example (m^i, x_1^i) pairs: **a. Random**: The c pairs are simply randomly drawn from the training data, aiming at providing diverse examples to the Context; **b. Metadata matching**: Before random sampling, we filter the user pool to include only the ones who have one or more attribute fields that are identical to those of the target user. This strategy aims to give LLM the examples that are more relevant to the target. **c. Top-c Nearest Neighbors**: The strategy takes a step further into relevant context by selecting the closest examples to the target. The distance is measured on natural sentences formed by their item features through textual embeddings such as [3, 4, 10].

The Prompt concludes with the Query, which contains the metadata of target user m. It is then served to the LLM generation step, which is detailed in the next section.

2. LLM generation. We feed the formulated prompt into the LLM and receives k recommended imaginary items $\tilde{X} = \{\tilde{x}_1, ..., \tilde{x}_k\}$. These items are represented by their generated textual, categorical attributes and timestamps, which do not necessarily match those of actual items in the dataset. This representation is well-suited for content-based recommendation backbones [9, 11, 13], which are used in our experiments. Potentially, they can also be mapped into actual items by embedding matching for ID-based sequential recommender systems [8, 14].

3. Imaginary-historical input fusion. In this stage, the LLM-generated imaginary items $\tilde{\mathbf{X}}$ are joined with available historical interaction sequence $\mathbf{X}$ to form the enhanced input sequence $\hat{\mathbf{X}}$ to the recommender. We design two fusion strategies:

a. Early Fusion: The imaginary items are first sorted by their generated timestamps to form a proper temporal sequence. Then we concatenate that imaginary sequence with the historical interaction sequence to form the input for the backbone recommender: $\hat{\mathbf{X}} = [\tilde{\mathbf{X}}, \mathbf{X}]$. In the absolute cold-start case, the historical sequence is empty and the input is solely made of the imaginative interactions. This technique takes on the intuition that the static user intention inferred from metadata precedes any actual interactions.

b. Late Fusion: Different from early fusion, in this strategy we separate k variants of imagined LLM suggestions $\tilde{X}$ and concatenate each of them with the historical sequence X independently. The variants are joined later at the latent embeddings of the recommender: $\hat{X} = \{\hat{x}_i := [\tilde{x}_i, X]\}_{i=1,..,k}$. This strategy is aimed at exploring the stochastic property of generated items, where the variants represent the distribution of possible interacted items given the metadata.

4. Sequential recommendation. The combined imaginary-historical interaction sequence is passed as input to the sequential recommender backbone, which generates the output recommended item y given the input sequence $\hat{\mathbf{X}}$. The sequential recommenders commonly can be dissected into an encoder $\mathcal{E}$ which processes the input sequence into a hidden latent vector $\mathbf{h} = \mathcal{E}(\hat{\mathbf{X}})$, and decoder $\mathcal{D}$ which generate output y from h: $y = \mathcal{D}(\mathbf{h})$.

In *early fusion*, a single pass of the encoder and decoder is done as in a regular inference case. In *late fusion*, we call the encoder of the recommenders k times on each member of $\hat{\mathbf{X}}$: $\mathbf{h}_i = \mathcal{E}(\hat{x}_i)$, then join the encoded embeddings up by average pooling into a single embedding: $\mathbf{h} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{h}_i$. This joint embedding is then used to decode the output $y = \mathcal{D}(\mathbf{h})$.

[1] https://grouplens.org/datasets/movielens/1m/

Table 1: Recformer [9] performance with and without our framework. The chunk with 0 historical interaction denotes the group of absolute cold users. We compare "no generator": no imaginary items inserted, "random": random in-context examples are given to the LLM, "match attr": in-context examples have one user attribute match with the target, and "*c*-NN": the *c* in-context examples are drawn from nearest neighbors of the target user. Best results are bolded, second best underlined.

ML-1M																	
#historical interaction		0		1-5		6-10		11-15		16-20		21-30		31-100		100+	
setting	fusion	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20
no generator	—	0.145	0.382	0.159	0.300	0.202	0.500	**0.329**	0.550	0.427	0.737	0.427	0.766	**0.317**	**0.753**	0.368	0.690
random	early	0.220	0.505	**0.219**	**0.600**	0.166	0.500	0.311	**0.600**	**0.445**	**0.842**	0.393	0.735	0.288	0.614	**0.374**	**0.704**
	late	0.136	0.326	0.169	0.344	**0.220**	0.469	0.190	0.469	0.429	0.813	0.385	0.745	0.315	0.738	0.318	0.662
match attr	early	0.210	0.463	0.143	0.450	0.213	**0.600**	0.237	0.400	0.403	0.737	**0.434**	**0.786**	0.309	0.718	0.345	0.663
	late	0.136	0.326	0.168	0.400	0.210	0.550	0.232	0.450	0.377	0.684	0.406	0.755	0.251	0.530	0.306	0.629
c-NN	early	**0.228**	**0.516**	0.167	0.550	0.165	0.450	0.326	**0.600**	0.396	0.737	0.390	0.735	0.283	0.596	0.350	0.700
	late	0.142	0.343	0.102	0.250	0.121	0.350	0.155	0.550	0.364	0.632	0.320	0.621	0.200	0.488	0.261	0.573

Amazon Proprietary dataset																	
#historical interaction		0		1		2		3		4		5		6-10		11+	
setting	fusion	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20
no generator	—	0.452	0.730	0.356	0.646	0.479	0.800	0.414	0.758	0.409	0.795	**0.396**	0.727	0.447	**0.853**	0.449	0.762
random	early	0.452	0.718	0.412	0.693	0.488	0.810	**0.480**	**0.832**	0.429	0.811	0.358	0.727	**0.460**	0.836	0.434	**0.857**
	late	0.481	0.759	**0.448**	**0.745**	**0.526**	0.839	0.443	0.777	0.416	0.795	0.352	0.727	0.429	0.818	0.408	0.714
match attr	early	0.301	0.606	0.310	0.630	0.352	0.727	0.405	0.811	0.374	0.764	0.344	0.682	0.405	**0.853**	0.434	0.810
	late	0.477	0.765	0.436	0.729	0.519	**0.846**	0.447	0.814	**0.438**	**0.819**	0.384	**0.773**	0.445	0.836	0.412	0.762
c-NN	early	0.433	0.736	0.395	0.682	0.463	0.792	0.462	0.780	0.409	0.780	0.384	**0.773**	0.454	0.836	**0.484**	0.810
	late	**0.487**	**0.777**	0.427	**0.745**	0.481	0.815	0.467	0.814	0.401	0.780	0.358	0.727	0.445	0.851	0.426	0.810

3 RESULTS AND DISCUSSION

Settings. We experiment with our framework on two datasets: *a. the MovieLens 1M (ML-1M)* public dataset and *b. the Amazon Proprietary (AP) dataset.* The AP dataset naturally consists of a significant proportion of cold and near-cold users. For ML-1M, we randomly select a set of users' historical interactions and trim them down to simulate those targeted scenarios. For the imaginary item generator LLM, here we report results for the Llama-3.3-70B-Instruct[2], which is not fine-tuned to focus on building high-quality in-context prompts. For the sequential recommender backbone, here we employ Recformer [9] for its high performance, good efficiency and reliable implementation.

Hyperparameter settings include in-context size $c = 10$, imaginary video items $k = 5$. For evaluation, we use the common leave-one-out strategy [8] and rank the ground truth item among the other 100 sampled negative items.

We conduct experiments on the three strategies of context building and the two methods for interaction fusion. We also compare to a baseline method called *no generator*, which simply samples a movie from the dataset to be the interaction input for absolute cold users, and leaves the other users' interactions as is.

Results. We report the NDCG@20 and Recall@20 values evaluated under each setting for sampled test users grouped into different historical interaction chunks in Tab 1. The performance clearly indicates that imaginary item augmentation with LLM by our method consistently boosts recommendation accuracy. This improvement is generally stronger in cold (0 historical interaction) or very nearly cold (1-10 in ML-1M; 1-4 in AP) users. As the number of historical interactions increases, the effect of imaginary items wanes and eventually approaches the "no generator" baseline.

Insights. We observe that random example sampling for in-context learning achieves strong performance for users with at least one historical interaction, while *c*-NN sampling is preferable for cold users (0 interactions).

We suggest that these findings are the result of the differing movie exploration behaviors within the movie search space: conservative for *c*-NN versus exploratory for random sampling. Notably, increased exploration appears less risky and thus beneficial given at least one user interaction, whereas it degrades performance for cold users, for whom more conservative in-context examples are more effective.

Finally, we report that for ML-1M, the early fusion injection yields better performance, whereas for the Amazon internal dataset, late fusion is more informative. This is most likely a consequence of the lower content diversity of AP, which renders interpolated movie representations more meaningful as the representation space is more compact.

4 PRACTICAL CONSIDERATIONS

Our proposed framework was shown to be effective in generating personalized recommendations for absolute and nearly cold users. However, it requires an average imaginary video item inference time per user of 72s and 175s for the ML-1M and AP datasets, respectively. For this reason, the LLM generation process needs to be implemented offline, which is possible due to the limited set of available user metadata. The imaginary examples can be stored with user metadata attributes as keys, and looked up in real-time. In this way, the strict latency constraints of the service are met.

Another technical consideration is that sometimes the LLM failed to correctly select the categorical value from a pre-defined attribute value set, which is solved by re-prompting to generate a valid set of imaginary video items for a test user. We also note that the absence of a fine-tuning stage may limit the achieved performance on the video semantic generation task under consideration, and we will address this in future work.

[2] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct. This model is used in the context of this paper only and not for our production system.

5 SPEAKER BIO

Xurong Liang is an Applied Scientist Intern at Amazon Australia. He is currently pursuing his PhD degree from the University of Queensland, with a focus on lightweight recommender systems.

Vu Nguyen is a Senior Applied Scientist at Amazon Australia, specializing in data-efficient machine learning for customer-impact applications. He received his PhD from Deakin University in 2015. He has been working at Deakin, Credit AI, University of Oxford before joining Amazon in late 2020.

Vuong Le is a Senior Applied Scientist at Amazon Australia with research interests in Neural reasoning, Recommendation systems and Video understanding. He received his PhD in ECE at University of Illinois at Urbana-Champaign in 2014 and joined Amazon thereafter. He served as a Senior research lecturer at Deakin University between 2017 and 2022 and returned to Amazon since then.

Paul Albert is an Applied Scientist at Amazon Australia, where his research interests encompass parameter-efficient finetuning, Large Language Models, unsupervised learning, and recommendation systems. He received his PhD from Dublin City University, Ireland, in 2023. Subsequently, he joined the Center for Augmented Reasoning in Adelaide, Australia, before commencing his role at Amazon Australia in 2025.

Julien Monteil is leading the Machine Learning group at Amazon International Machine Learning Australia, which primarily focuses on the Research and Development of recommender systems for Amazon customers globally. He is also an Adjunct Senior Lecturer at the University of Queensland. He has 12 years of post-PhD experience in the Research and Development of ML systems for customer-facing applications in retail, automotive, networking and healthcare.

REFERENCES

[1] Geetha Sai Aluri, Siddharth Sharma, Tarun Sharma, and Joaquin Delgado. 2024. Playlist search reinvented: LLMs behind the curtain. In *RecSys*. 813–815.

[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*. 1007–1014.

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*. 4171–4186.

[5] Hao Ding, Anoop Deoras, Bernie Wang, and Hao Wang. 2021. Zero-Shot Recommender Systems. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*.

[6] Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints* 3 (2023).

[7] Feiran Huang, Yuanchen Bei, Zhenghang Yang, Junyi Jiang, Hao Chen, Qijie Shen, Senzhang Wang, Fakhri Karray, and Philip S Yu. 2025. Large Language Model Simulator for Cold-Start Recommendation. In *WSDM*. 261–270.

[8] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[9] 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. In *KDD*. 1258–1267.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[11] Yang Liu, Yitong Wang, and Chenyue Feng. 2024. Unirec: A dual enhancement of uniformity and frequency in sequential recommendations. In *CIKM*. 1483–1492.

[12] Guangtao Nie, Rong Zhi, Xiaofan Yan, Yufan Du, Xiangyang Zhang, Jianwei Chen, Mi Zhou, Hongshen Chen, Tianhao Li, Ziguang Cheng, et al. 2024. A hybrid multi-agent conversational recommender system with llm and search engine in e-commerce. In *RecSys*. 745–747.

[13] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *NIPS* 36 (2023), 10299–10315.

[14] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.

[15] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *WWW* 27, 5 (2024), 60.

[16] Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, et al. 2025. Cold-Start Recommendation towards the Era of Large Language Models (LLMs): A Comprehensive Survey and Roadmap. *arXiv preprint arXiv:2501.01945* (2025).

[17] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*. 1893–1902.