User Knowledge Prompt for Sequential Recommendation

Yuuki Tachioka
tachioka.yuki@core.d-itlab.co.jp
Denso IT Laboratory
Minato, Tokyo, Japan

Abstract

The large language model (LLM) based recommendation system is effective for sequential recommendation, because general knowledge of popular items is included in the LLM. To add domain knowledge of items, the conventional method uses a knowledge prompt obtained from the item knowledge graphs and has achieved SOTA performance. However, for personalized recommendation, it is necessary to consider user knowledge, which the conventional method does not fully consider because user knowledge is not included in the item knowledge graphs; thus, we propose a user knowledge prompt, which converts a user knowledge graph into a prompt using the relationship template. The existing prompt denoising framework is extended to prevent hallucination caused by undesirable interactions between knowledge graph prompts. We propose user knowledge prompts of user traits and user preferences and associate relevant items. Experiments on three types of dataset (movie, music, and book) show the significant and consistent improvement of our proposed user knowledge prompt.

CCS Concepts

• **Information systems** → **Recommender systems**; **Personalization**.

Keywords

sequential recommendation, personalization, user knowledge graph, LLM, collaborative filtering

ACM Reference Format:
Yuuki Tachioka. 2024. User Knowledge Prompt for Sequential Recommendation. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3640457.3691714

1 Introduction

Sequential recommendation can consider the user-item interaction history to recommend the next items which user may be interested in, and is important to capture dynamic changes in user preferences. As in collaborative filtering (CF) [7], it is important to capture the relationship between items and users, and attention-based methods between items and users have been proposed [10, 26, 28], but the use of item knowledge is insufficient. For example, it is important to consider the category of the item [3]. In particular, a recently improved large language model (LLM) can effectively recommend items based on the traits of the item by taking advantage of the knowledge of the item [4, 5, 12, 18], because the general knowledge of popular items is included in the text data for pretraining. The pre-train, personalized prompt, and predict paradigm (P5) [5] is one of the most widely used methods and uses a personalized prompt to convert the recommendation task into a natural language processing task [13, 22]. To recommend various items, including unpopular items, domain knowledge is needed, but the general knowledge included in the LLM is insufficient. To complement insufficient knowledge, recommendation models that use external sources of knowledge of items have improved the performance of the recommendation [8, 9, 21].

To introduce external knowledge into P5, the knowledge prompt tuning for sequential recommendation (KP4SR) has been proposed. The masked personalized prompt (MPP) is used to recommend the next items and the item knowledge graph (IKG) is inputted into LLM using the item knowledge prompt (IKP), which converts IKG into a prompt using relation templates[1]. KP4SR has achieved SOTA performance compared to other LLM-based recommendation methods in three recommendation tasks, including book, music, and movie [24]. When the amount of inputted KGs increases, unrelated KG can cause hallucinations, and it is necessary to avoid introducing noisy knowledge [1, 20, 23, 25]. KP4SR can address this problem by prompt denoising, which can control the attention of the LLM between the KGs and focus the attention between only the related KGs.

KP4SR using IKP can effectively consider item traits, but cannot consider user traits such as user attributes and preferences. Recommendation that fits user traits is desirable [2, 19]; thus, we propose a user knowledge prompt (UKP) that inputs user knowledge obtained from the user knowledge graphs (UKGs) into LLM using the relation templates in a similar way to the IKP of KP4SR. The proposed recommendation model can simultaneously consider both item knowledge and user knowledge, by exploiting the prompt denoising framework.

Furthermore, for effective recommendation, the conversion, combination, and inference of UKGs can extract useful information. For example, the relationship between the item and the users, as CF considers, is also important [3, 9, 15]. Our method can use the UKP of association of items by user preference, which produces CF-like effects, obtained from the inference of multiple UKGs of users whose preferences are similar. Experiments on three types of dataset show that our proposed method significantly improves the performance of KP4SR consistently on different tasks.

[1] In our paper, knowledge graph and knowledge prompt used in KP4SR are described as IKG and IKP to distinguish them used in our proposed method.

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
RecSys '24, October 14–18, 2024, Bari, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3691714

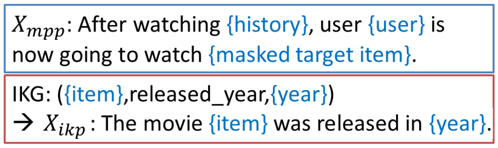

Figure 1: Example of MPP and IKP from IKG.

2 Item Knowledge Prompt (IKP) by KP4SR

This section describes KP4SR [24], which is the baseline of the proposed method. KP4SR uses two types of prompts: MPP and IKP, with prompt denoising, which prevents hallucinations when IKP is introduced.

MPP converts recommendation tasks into natural language processing tasks, as does P5 [5]. Fig. 1 shows an example of MPP[2]. MPP $X_{mpp}(u, \mathcal{H}_m)$ is constructed by the user u and the user's history of previously watched m items $\mathcal{H}_m = \{i_1, \ldots, i_m\}$ with the target item masked. We can obtain the next item, which users can be interested in by predicting the next $m+1$-th target item masked, i_{m+1}, by LLM.

IKP is a prompt converted from IKG using relation templates. KGs are composed of triples (head / relation / tail) as (h, r, t). There are K IKGs about N items and they are formed in the tree structure $\mathcal{K}_i(\{i_1, \ldots, i_N\}) = \{\kappa_1, \ldots, \kappa_K\}$ as shown in Fig. 2. To input the knowledge of IKGs into LLM, IKP can convert an IKG using a template $\mathcal{T}$ about the relation r. For example, the IKG $\kappa = (i, \text{genre}, \textit{genre})$ of the item i is converted by the relation template about $r = \text{genre}$, $\mathcal{T}(\kappa) =$ "The genre of i is *genre*". The construction of IKP can generate a prompt from the IKGs obtained from $\mathcal{K}_i$ with an interaction history as $X_{ikp}(\mathcal{H}_m, \mathcal{K}_i) = \bigcup_{\kappa \in \mathcal{K}_i(\mathcal{H}_m)} \mathcal{T}(\kappa)$, where $\bigcup$ concatenates the sentences generated using templates.

The inputted prompt to LLM is

$$X_p(u, \mathcal{H}_m, \mathcal{K}_i) = [SP]X_{mpp}(u, \mathcal{H}_m)[SP]X_{ikp}(\mathcal{H}_m, \mathcal{K}_i)[SP], \quad (1)$$

where $[SP]$ is a special token. Loss function is a maximization of the likelihood of P_θ by using model θ as

$$\mathcal{L}_\theta = -\sum_{u \in \mathcal{U}} \sum_{m=1}^{M} \log P_\theta(i_{m+1} | X_p(u, \mathcal{H}_m, \mathcal{K}_i)), \quad (2)$$

where $\mathcal{U}$ is a set of users. For inference, by using beam search, the next item lists are predicted.

Prompt denoising prevents hallucination and confusion when using multiple KGs. For example, there are two similar KGs: "Item i_1 has a relationship r_1 with α_1" and "Item i_2 has a relationship r_1 with β_1". When both KGs are inputted into LLM, it may wrongly infer "Item i_1 has a relationship r_1 with β_1" [23]. This also occurs when KG is hopped. To solve this problem, a knowledge tree mask is used. After searching the knowledge tree as shown in Fig. 2, prompt denoising makes only the self-, parent-, child-, and sibling[3]-KGs visible and the other KGs invisible. To control visibility, the attention of the transformer between the target tokens t and t' is modified to $\mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d} + \mu_{t,t'})\boldsymbol{V}$, where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are the query, key, and value matrix of the transformer, respectively, and d is a query dimension. The offset μ is

$$\mu_{t,t'} = \begin{cases} 0 & k = k', p(\kappa_k) = \kappa_{k'}, p(\kappa_{k'}) = \kappa_k, p(\kappa_k) = p(\kappa_{k'}) \\ -\infty & \text{otherwise} \end{cases}, \quad (3)$$

where k and k' are indices of KGs before conversion into prompt ($t \in \mathcal{T}(\kappa_k)$, $t' \in \mathcal{T}(\kappa_{k'})$) and $p(\kappa)$ is a function that extracts the parent node of κ from KG tree.

[2] We assume examples are movie, but if interaction can be changed as read or listen, KP4SR and our proposed method can be easily applied to other types of items such as music or book.

[3] Their parent nodes are the same.

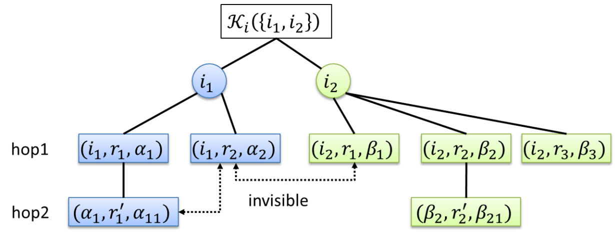

Figure 2: Example of IKG tree $\mathcal{K}_i$ with prompt denoising by using knowledge tree mask where self-, parent-, child-, and sibling-KGs are visible, e.g., item i_1 has a relationship r_1 with α_1 and (i_1, r_2, α_2) is invisible to (i_2, r_1, β_1) and vice versa.

3 User Knowledge Prompt (UKP)

We propose using UKP to consider user knowledge as in Sec. 3.2, 3.3, and 3.4. After UKP X_{ukp} is obtained from the UKG tree $\mathcal{K}_u$ using a relation template, $X_{ukp}(u, \mathcal{H}_m, \mathcal{K}_u) = \bigcup_{\kappa \in \mathcal{K}_u(u, \mathcal{H}_m)} \mathcal{T}(\kappa)$ is inputted to LLM after X_{mpp} and X_{ikp} as

$$X'_p(u, \mathcal{H}, \mathcal{K}_i, \mathcal{K}_u) = X_p(u, \mathcal{H}, \mathcal{K}_i)X_{ukp}(u, \mathcal{H}, \mathcal{K}_u)[SP]. \quad (4)$$

Loss function is also a maximization of likelihood of P_θ as

$$\mathcal{L}_\theta = -\sum_{u \in \mathcal{U}} \sum_{m=1}^{M} \log P_\theta(i_{m+1} | X'_p(u, \mathcal{H}_m, \mathcal{K}_i, \mathcal{K}_u)). \quad (5)$$

3.1 Prompt denoising for UKP

Prompt denoising can also be applied to UKP. Fig. 3 shows that the UKG consists of two parts: user trait parts (e.g., $(u, \text{age}, \textit{age})$), which are complete within the UKG $\mathcal{K}_u$, and user-item interaction parts (e.g., $(u, \text{rate-}v, i)$), which are related to IKG $\mathcal{K}_i$. This is different from IKG, and the application of prompt denoising is also different for user-item interaction parts. For parts of the user trait, prompt denoising is also performed as KP4SR does. For parts of the user-item interaction, to reflect the knowledge of the item, when the tail of the UKG is an item i, the IKGs of the item i, which belong to $\mathcal{K}_i$, become visible by modifying the offset $\mu = 0$. When the token t is from $\mathcal{K}_u$ ($t \in \mathcal{T}(\kappa_k)$, $\kappa_k \in \mathcal{K}_u$) and t' is from $\mathcal{K}_i$ ($t' \in \mathcal{T}(\kappa_{k'})$, $\kappa_{k'} \in \mathcal{K}_i$), the condition for μ is added to (3) as

$$\mu_{t,t'} = 0 \text{ if } \mathrm{tail}(\kappa_k) = \mathrm{head}(\kappa_{k'}), \quad (6)$$

where tail and head are functions that return tail and head of KG. In the figure example, because the tail of (u, R_3, i_1) is an item i_1, this KG is visible to KGs $(i_1, *, *)$ and vice versa, where $*$ is an arbitrary element. Because the attention between X_{ikp} and X_{ukp} is zero except in the above case, our method can consider both the item traits and the user traits to prevent hallucination.

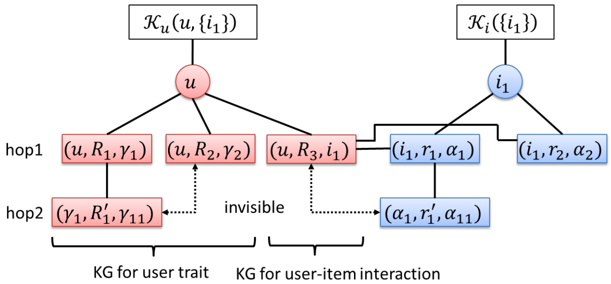

Figure 3: Example of UKG tree $\mathcal{K}_u$ in conjunction with IKG $\mathcal{K}_i$ e.g., user u has a relationship R_1 with γ_1. prompt denoising is also applied to UKG, but only for user-item interaction, related IKGs are visible, e.g., $(i_1,*,*)$ is visible to (u,R_3,i_1) and vice versa.

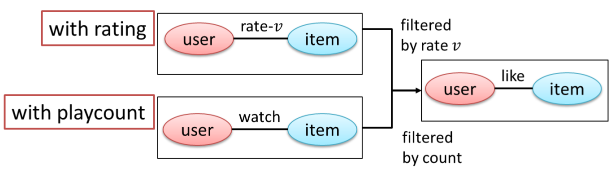

Figure 4: Conversion of user-item relationship into user preference UKG.

3.2 UKP of user traits

3.2.1 User attributes. User attributes can be expressed as UKG. For example, the age of the user u is (u, age, age). This can be converted into UKP "The age of the user u is *age*". UKG $(u, \text{from}, country)$ can be converted into UKP "The user u is from *country*".

3.2.2 User-item interaction characteristics. UKP can consider user-item interaction characteristics. The selection of items depends on the types of users. If UKG $(u, \text{distinct-films}, \text{high})$ is obtained[4], u likes to explore various genres of films and tends to accept the recommendation of items of the newly encountered genre.

3.3 UKP of item rating by interacted users

From the user-item interaction UKG $(u, \text{rate-}v, i)$, average rating KG of the item i is obtained as $\text{ave}_{u\in\mathcal{U}_i}((u, \text{rate-}v, i))$ by taking an average (ave) for all users who interact with the item i, $\mathcal{U}_i$. KG $(i, \text{ave-rate}, arate)$ can be converted into "The average rate of the item i by the interacted users is *arate*" using the template. This can be used for popularity-based recommendation, which is effective for whom with high agreeableness [19].

3.4 UKP of association of items by user preferences

3.4.1 User preference. User preference can be expressed as UKG. If user-annotated UKG (u, like, i) can be obtained, it is easy to convert it to UKP. If these UKGs cannot be obtained, the user-item interaction UKGs can be converted as follows.

Ratings are effective when items have reliable ratings and are not played repeatedly, such as movies. For UKG where the user u rates item i $(u, \text{rate-}v, i)$ [5] is filtered by the threshold φ satisfying $v > \varphi$ as (u, like, i) if $v > \varphi$ in $(u, \text{rate-}v, i)$.

Playcount is effective when items are played repeatedly. Especially for music, favorite tracks are played repeatedly. The UKG of the user-item interaction (u, listen, i) is produced event by event, and these are filtered by playcount as (u, like, i) if $|(u, \text{listen}, i)| > \eta$, where $|\cdot|$ is the number of UKGs.

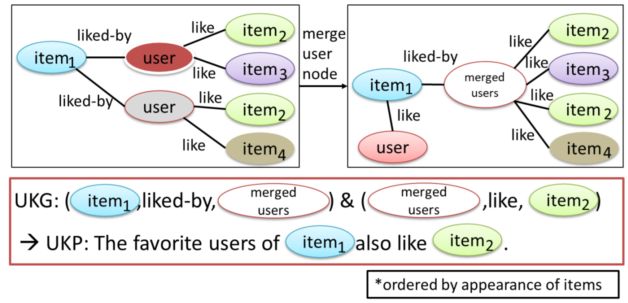

Figure 5: Association of relevant items from user-item interaction obtained from UKG and its conversion into UKP.

3.4.2 Item association. The inference of UKGs can associate items by preferences. First, the user's preference UKG (u, like, i_1) can be obtained as in Sec. 3.4.1. Its head and tail are replaced to obtain KG $(i_1, \text{liked-by}, u)$ as in Fig. 5. The merging of the user nodes results in KGs $(i_1, \text{liked-by}, \text{merged users})$ and $(\text{merged users}, \text{like}, i_*)$. After reordering items in the descending order of $|(\text{merged users}, \text{like}, i_*)|$, KG $(\text{merged users}, \text{like}, i_2)$ is obtained in the figure example, because the count of item$_2$ (i_2) is greater than those of item$_3$ and item$_4$. The combination of these KGs can be converted to UKP "The favorite users of i_1 also like i_2". This can provide a list of relevant items suggested by users whose preferences are similar. Because this type of information can be obtained by CF, this prompt can produce CF-like effects. It is especially effective when the number of user-interacted items is small [11, 27].

3.5 Quantization of continuous variables

LLM is not good at dealing with numerical variables and it is difficult to automatically categorize continuous variables when inputting continuous variables directly into LLM. For example, the LLM tokenizer encodes continuous variables using different characters [17]. To encourage grouping of continuous variables, the continuous variables should be quantized before input. Ages are grouped by 10 years, e.g., $(u, \text{age}, 43)$ is quantized as $(u, \text{age}, 40\text{s})$. Ratings are quantized in 0.25 steps, e.g., $(i, \text{ave-rate}, 4.15)$ is quantized as $(i, \text{ave-rate}, 4.25)$. Numerical variables are converted into the expression of the extent (very high / high / middle / low / very low) for each quintile. For example of playcount, the original KG $(u, \text{playcount}, 12450)$ is converted to $(u, \text{playcount}, \text{high})$.

4 Experiment

4.1 Experimental setup

As in the KP4SR paper [24], we prepared three types of dataset (lfm-1b [16], movielens [6], and amazon-books [14]) to validate the effectiveness of our proposed method. Compared with 11 methods,

[4]Numerical variables such as the number of distinct tracks is converted to the expression of the extent such as "high" as shown in Sec. 3.5.

[5]For example, when the user u rates the item i as 4.0, UKG is $(u, \text{rate-}4.0, i)$.

Table 1: Statistics of the datasets

Stats	lfm-1b (track)	movielens	amazon-books
# users	120,280	69,878	651,598
# items	6,911,817	9,708	396,702
# interactions	1,026,252,572	9,995,471	17,518,845

Table 2: NDCG and Hit Rate [%] (lfm-1b). A statistical test confirmed that all improvements are significant at $\alpha = 0.01$ level (**).

	NDCG			Hit Rate		
K	5	10	50	5	10	50
KP4SR	14.07	14.96	15.59	16.99	19.76	22.22
UKP(u)	14.19**	15.09**	15.74**	17.14**	19.91**	22.46**
UKP(cf)	**15.09****	**16.17****	**16.86****	**18.75****	**22.07****	**24.79****

including advanced methods based on attention and LLM [5, 10, 18, 28], KP4SR has achieved SOTA performance. Table 1 shows the statistics of the datasets[6]. movielens has a smaller number of items than the number of users, whereas lfm-1b has a smaller number of users than the number of items. amazon-book has similar orders.

To remove noises as in previous studies[24], for lfm-1b and movielens, items that appeared less than ten times were removed and users who interacted less than five times were removed. For amazon books, items that appeared less than 20 times were removed, and users who interacted less than ten times were removed. Users whose playlist length was more than 10 were reliable users who were used to generate a list of relevant items. For lfm-1b, when the playcount was greater than the third quantile φ, the item was the favorite. For movielens, when the rating was more than $\eta = 3$, the item was favorite.

Based on preliminary studies, the maximum number of sequential interactions was five, the maximum degree, which is the number of tails for the identical head and relation, was eight, and the maximum hop number of KGs was two. The maximum degree of relevant items obtained from UKG was five. The model dimension was 512 and eight multihead attentions were used. AdamW optimizer was used with a maximum learning rate of 0.001. The batch size was 64 and the number of epochs was 100 for lfm-1b and movielens and 60 for amazon-books.

Performance was evaluated in terms of the normalized discounted cumulative gain (NDCG)@K and Hit Rate@K with $K = \{5, 10, 50\}$ and in a leave-one-out manner, where, keeping the last two items for testing, we trained the models using other data. The items kept were masked and performance was evaluated for them. A paired t-test with Bonferroni correction between KP4SR and UKP was used to test significance.

4.2 RQ: Does the UKP of user traits improve performance?

Table 2 shows the results of lfm-1b. Compared to KP4SR that considers the traits of the items, the proposed UKP(u) that can consider user traits in Sec. 3.2 significantly improved performance. This indicates that the consideration of user traits is important for recommendation.

[6]We reproduce the datasets and the statistics of the datasets are different from that of [24].

Table 3: NDCG and Hit Rate [%] (movielens). A statistical test confirmed that all improvements are significant at $\alpha = 0.01$ level (**).

	NDCG			Hit Rate		
K	5	10	50	5	10	50
KP4SR	6.98	8.61	10.25	10.30	15.35	21.88
UKP(cf)	8.20**	10.06**	11.98**	12.16**	17.94**	25.60**
UKP(r)	**8.30****	**10.23****	**12.19****	**12.30****	**18.29****	**26.08****

Table 4: NDCG and Hit Rate [%] (amazon-books). A statistical test confirmed that all improvements are significant at $\alpha = 0.01$ level (**).

	NDCG			Hit Rate		
K	5	10	50	5	10	50
KP4SR	4.16	4.64	5.07	5.32	6.82	8.52
UKP(cf)	**5.12****	**5.61****	**6.04****	**6.49****	**8.03****	**9.73****

4.3 RQ: Does the UKP of the association of items improve performance?

The UKP of the association of items for CF (UKP(cf)) which can consider the relevant items suggested by similar users in Sec. 3.4 improved the performance and was better than UKP(u). This indicates that for the lfm-1b dataset, in addition to the item traits obtained from IKG, the association of the relevant items from UKG was helpful. Table 3 shows the results of the movielens. UKP(cf) also improved performance significantly compared to KP4SR.

4.4 RQ: Does the UKP of the item rating improve performance?

Because movielens provides rating information, UKP(r), which inputs ratings by prompt in Sec. 3.3 was also validated. In this case, UKP(r) was better than UKP(cf), which indicates that the rating information is effective when provided.

4.5 RQ: Is UKP effective for other datasets?

To confirm that our proposed method is also effective for other datasets, UKP(cf) was validated on the amazon-books dataset as shown in Table 4. Our proposed method also significantly improved performance compared to KP4SR, which confirmed that our proposed method is effective for different types of dataset. The UKG used in UKP(cf) can be derived from the user-item interaction history; this method can be widely applicable.

5 Conclusion

To consider the traits of the users for sequential recommendation, we modify KP4SR that uses IKP obtained from the IKG, and we propose a UKP of the user traits by extracting UKG, which is compatible with IKP. We propose UKP of user attribute, user rating, and association of items by similar users that produces CF-like effects. The prompt denoising focuses on attention between related KGs and prevents hallucination, when both IKP and UKP are inputted into LLM. Experiments on three sequential recommendation tasks show that our proposed UKPs significantly improved the performance of KP4SR on different datasets, indicating the consistent effectiveness of our proposed method.

References

[1] Giacomo Balloccu, Ludovico Boratto, Christian Cancedda, Gianni Fenu, and Mirko Marras. 2024. Faithful Path Language Modeling for Explainable Recommendation over Knowledge Graph. arXiv:2310.16452 [cs.IR] https://arxiv.org/abs/2310.16452

[2] Laura Burbach, Johannes Nakayama, Nils Plettenberg, Martina Ziefle, and André Calero Valdez. 2018. User Preferences in Recommendation Algorithms: The Influence of User Diversity, Trust, and Product Category on Privacy Perceptions in Recommender Algorithms. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 306–310. https://doi.org/10.1145/3240323.3240393

[3] Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. 2021. Category-aware Collaborative Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 388–397. https://doi.org/10.1145/3404835.3462832

[4] Yuwei Cao, Nikhil Mehta, Xinyang Yi, Raghunandan Hulikal Keshavan, Lukasz Heldt, Lichan Hong, Ed Chi, and Maheswaran Sathiamoorthy. 2024. Aligning Large Language Models with Recommendation Knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 1051–1066. https://doi.org/10.18653/v1/2024.findings-naacl.67

[5] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) *(RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 299–315. https://doi.org/10.1145/3523227.3546767

[6] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2016), 19 pages. https://doi.org/10.1145/2827872

[7] 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* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[8] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 505–514. https://doi.org/10.1145/3209978.3210017

[9] Xiaowen Huang, Quan Fang, Shengsheng Qian, Jitao Sang, Yan Li, and Changsheng Xu. 2019. Explainable Interaction-driven User Modeling over Knowledge Graph for Sequential Recommendation. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19)*. Association for Computing Machinery, New York, NY, USA, 548–556. https://doi.org/10.1145/3343031.3350893

[10] W. Kang and J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE Computer Society, Los Alamitos, CA, USA, 197–206. https://doi.org/10.1109/ICDM.2018.00035

[11] Pengyang Li, Rong Chen, Quan Liu, Jian Xu, and Bo Zheng. 2022. Transform Cold-Start Users into Warm via Fused Behaviors in Large-Scale Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2013–2017. https://doi.org/10.1145/3477495.3531797

[12] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, Prompt, and Recommendation: A Comprehensive Survey of Language Modeling Paradigm Adaptations in Recommender Systems. *Transactions of the Association for Computational Linguistics* 11 (12 2023), 1553–1571. https://doi.org/10.1162/tacl_a_00619

[13] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. LLM-Rec: Personalized Recommendation via Prompting Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 583–612. https://doi.org/10.18653/v1/2024.findings-naacl.39

[14] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 188–197. https://doi.org/10.18653/v1/D19-1018

[15] Jiarui Qin, Kan Ren, Yuchen Fang, Weinan Zhang, and Yong Yu. 2020. Sequential Recommendation with Dual Side Neighbor-based Collaborative Relation Modeling. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 465–473. https://doi.org/10.1145/3336191.3371842

[16] Markus Schedl. 2016. The LFM-1b Dataset for Music Retrieval and Recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval* (New York, New York, USA) *(ICMR '16)*. Association for Computing Machinery, New York, NY, USA, 103–110. https://doi.org/10.1145/2911996.2912004

[17] Aaditya K. Singh and DJ Strouse. 2024. Tokenization counts: the impact of tokenization on arithmetic in frontier LLMs. arXiv:2402.14903 [cs.CL] https://arxiv.org/abs/2402.14903

[18] 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 *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. ACM, New York, NY, USA, 1441–1450. https://doi.org/10.1145/3357384.3357895

[19] Yuuki Tachioka. 2023. Conditioning of variational autoencoder by user traits for item recommendation. In *Proceedings of the Music recommender systems workshop @ 17th ACM Conference on Recommender Systems (RecSys 2023)* (Singapore). 1–7. https://doi.org/10.5281/zenodo.8372443

[20] Shuyao Wang, Yongduo Sui, Chao Wang, and Hui Xiong. 2024. Unleashing the Power of Knowledge Graph for Recommendation via Invariant Learning. In *Proceedings of the ACM on Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 3745–3755. https://doi.org/10.1145/3589334.3645576

[21] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1929–1937. https://doi.org/10.1145/3534678.3539382

[22] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. 2024. Personalized Prompt for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3376–3389. https://doi.org/10.1109/TKDE.2024.3357498

[23] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1611–1621. https://doi.org/10.1145/3477495.3531963

[24] Jianyang Zhai, Xiawu Zheng, Chang-Dong Wang, Hui Li, and Yonghong Tian. 2023. Knowledge Prompt-tuning for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 6451–6461. https://doi.org/10.1145/3581783.3612252

[25] Haonan Zhang, Dongxia Wang, Zhu Sun, Yanhui Li, Youcheng Sun, Huizhi Liang, and Wenhai Wang. 2024. Does Knowledge Graph Really Matter for Recommender Systems? arXiv:2404.03164 [cs.IR] https://arxiv.org/abs/2404.03164

[26] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level deeper self-attention network for sequential recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (Macao, China) *(IJCAI'19)*. AAAI Press, 43204326. https://doi.org/10.24963/ijcai.2019/600

[27] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2020. Cold-start Sequential Recommendation via Meta Learner. In *AAAI Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:228083905

[28] 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 *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 1893–1902. https://doi.org/10.1145/3340531.3411954