



Sequential Recommendation with Latent Relations based on Large Language Model

Shenghao Yang
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing, China
ysh21@mails.tsinghua.edu.cn

Weizhi Ma*
AIR, Tsinghua University
Beijing, China
mawz@tsinghua.edu.cn

Peijie Sun
DCST, Tsinghua University
Beijing, China
sun.hfut@gmail.com

Qingyao Ai
DCST, Tsinghua University
Beijing, China
aiqy@tsinghua.edu.cn

Yiqun Liu
DCST, Tsinghua University
Beijing, China
yiqunliu@tsinghua.edu.cn

Mingchen Cai
Meituan
Beijing, China
caimingchen@meituan.com

Min Zhang*
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing, China
z-m@tsinghua.edu.cn

ABSTRACT

Sequential recommender systems predict items that may interest users by modeling their preferences based on historical interactions. Traditional sequential recommendation methods rely on capturing implicit collaborative filtering signals among items. Recent relation-aware sequential recommendation models have achieved promising performance by explicitly incorporating item relations into the modeling of user historical sequences, where most relations are extracted from knowledge graphs. However, existing methods rely on manually predefined relations and suffer the sparsity issue, limiting the generalization ability in diverse scenarios with varied item relations.

In this paper, we propose a novel relation-aware sequential recommendation framework with **Latent Relation Discovery (LRD)**. Different from previous relation-aware models that rely on predefined rules, we propose to leverage the Large Language Model (LLM) to provide new types of relations and connections between items. The motivation is that LLM contains abundant world knowledge, which can be adopted to mine latent relations of items for recommendation. Specifically, inspired by that humans can describe relations between items using natural language, LRD harnesses the LLM that has demonstrated human-like knowledge to obtain language knowledge representations of items. These representations are fed into a latent relation discovery module based on the discrete state variational autoencoder (DVAE). Then the self-supervised relation discovery tasks and recommendation tasks are jointly optimized. Experimental results on multiple public datasets demonstrate our proposed latent relation discovery method can be incorporated with existing relation-aware sequential recommendation models

and significantly improve the performance. Further analysis experiments indicate the effectiveness and reliability of the discovered latent relations.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

Sequential recommendation, Large language model, Latent relation

ACM Reference Format:

Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. 2024. Sequential Recommendation with Latent Relations based on Large Language Model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, July 14–18, 2024, Washington, DC, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3626772.3657762>

1 INTRODUCTION

Sequential recommendation is a promising research topic in the community of recommender systems, aiming to predict the next item the user prefers based on his/her interaction history [40]. Various methods have been proposed to address the sequential recommendation task. Early studies focused on estimating the transition relations between items based on Markov chain assumptions [30]. In recent years, with the advancement of deep learning, various deep neural networks, such as Recurrent Neural Networks (RNNs) [11, 13, 17], Convolutional Neural Networks (CNNs) [34], and Transformers [10, 15, 21, 33], have been incorporated to better model user preferences reflected in their sequential historical interactions.

Although existing methods have achieved remarkable performance, they usually rely on item-based collaborative filtering algorithms [32] to calculate the implicit collaborative similarity between items while overlooking the explicit relations between items, which are prevalent and significantly influence user decisions in the real recommendation scenario. Recently, some relation-aware sequential recommendation methods [36, 37, 43] have been proposed to explicitly consider item relations during modeling user preferences

*Corresponding authors

This work is supported by the Natural Science Foundation of China (Grant No. U21B2026, 62372260) and Quan Cheng Laboratory (Grant No. QCLZD202301).



This work is licensed under a Creative Commons Attribution International 4.0 License.

and significantly improve the performance of the sequential recommendation. However, current approaches still face some challenges that limit the application of these models.

Specifically, for most existing relation-aware methods, item relation data is typically stored in a knowledge graph, which may suffer from *sparsity* issue of two aspects. Firstly, the *relation sparsity* on the edge set. The relational item modeling of existing methods is performed based on manually predefined relations. The attribute-based relations (e.g., “share category”) and co-occurrence-based relations (e.g., “also buy”) are usually used as these relations are relatively straightforward to define and can be derived from user interaction data and item metadata. Nevertheless, the relations between items are diverse in the real world, and manually defined relations are sparse compared to all latent relations. Relying on a restricted set of predefined relations limits the model’s capacity to generalize effectively across diverse recommendation scenarios. Secondly, the *item sparsity* on the vertex set. It is caused by the inherent data sparsity issue of recommender systems [31] and particularly affects the data collection of the co-occurrence-based relation since it requires substantial interaction data to collect item pairs that conform to the relation definition. To alleviate the above issue, we investigate to discover latent relations between items that contribute to the recommendation.

In this paper, we propose a language knowledge-based **Latent Relation Discovery (LRD)** method for the sequential recommendation. The motivation behind this approach is inspired by the fact that humans usually describe relations between items in natural language based on their knowledge. Observing the rich world knowledge and semantic representation capabilities exhibited by Large Language Models (LLMs) [3, 25, 35], we propose to leverage the abilities of LLMs to discover latent item relations. Specifically, we design a self-supervised learning framework to facilitate the process of discovering latent relations. We first leverage an LLM to obtain language knowledge representations of items. Subsequently, a relation extraction module is adopted to predict the latent relation between two items. Then we incorporate an item reconstruction module to reconstruct the representation of one item based on the representation of the predicted relation and the other item. Through this self-supervised learning process, the objective of reconstructing the original items forces the relation extraction module to predict relations with sufficient accuracy and generality. Furthermore, we incorporate the LRD into the existing relation-aware sequential recommendation frameworks and perform joint optimization.

The merits of our proposed framework are threefold. Firstly, LRD does not rely on manually defined relations, and can autonomously discover latent relations between items. This enhances the model’s ability to better capture diverse preferences reflected in user interaction history. Secondly, the optimal objective of the relation-aware sequential recommendation task serves as supervised signals to guide the relation discovery process, leading to the discovery of relations more beneficial to the recommendation. Last but not least, analyzing the predicted item relations by the LRD contributes to better interpretability of relation-aware sequential recommendation models.

We perform experiments on multiple public datasets to evaluate our proposed LRD approach. Leveraging latent relations derived from language knowledge-based item representation, the

Table 1: Notations.

Notations	Descriptions
\mathcal{U}	The set of user
\mathcal{V}	The set of item
\mathcal{T}	The set of triplets
\mathcal{R}	The set of relations
\mathcal{R}_{def}	The set of predefined relations
\mathcal{R}_{latent}	The set of latent relations
S_u	The interaction sequence of user u
$\mathbf{u} \in \mathbb{R}^d$	The id embedding of user u
$\mathbf{m}_{u,v} \in \mathbb{R}^d$	The relation-aware user sequence representation of u
$\mathbf{v} \in \mathbb{R}^d$	The id embedding of item v
$\mathbf{e} \in \mathbb{R}^{d_L}$	The LLM-based embedding of item v
$\mathbf{r} \in \mathbb{R}^d$	The embedding of relation r

relation-aware sequential recommendation model captures more comprehensive user sequence representations. Experimental results demonstrate that compared to state-of-the-art (SOTA) relation-aware sequential recommendation models, the model enhanced by LRD achieves significantly better performance. Further analysis experiments reveal that the LRD module is indeed capable of discovering reasonable relations between items.

The main contributions of our work are summarized as follows:

- To the best of our knowledge, we first propose to discover latent relations based on LLM for relation-aware sequential recommender systems.
- We propose an LLM-based latent relation discovery framework, i.e., LRD, to harness the language knowledge to discover latent relations, which is a self-supervision learning method and flexible to work with existing relation-aware sequential recommenders through joint learning.
- Experimental results on multiple public datasets demonstrate that LRD significantly improves the performance of existing relation-aware sequential recommendation models by effectively discovering reliable relations between items.

2 PROBLEM STATEMENT

Let \mathcal{U} and \mathcal{V} denote the sets of users and items, respectively. For each user $u \in \mathcal{U}$, its chronologically-ordered interaction history is represented as $S_u = \{v_1, v_2, v_3, \dots, v_{N_u}\}$. For an item $v_i \in \mathcal{V}$, there may exist another related item v_{-i} with relation r , denoted as a triplet (v_i, v_{-i}, r) . \mathcal{R} denotes the set of relations, which is further divided into predefined relations set \mathcal{R}_{def} and latent relations set \mathcal{R}_{latent} . All relational item triplets associated with predefined relations \mathcal{T} can be stored in a knowledge graph \mathcal{G} , where the vertex set comprises all relational item pairs, and the edge set consists of all the predefined relations.

The objective of the sequential recommendation task is to provide a ranked list of items for the user u at the next interaction considering their interaction history S_u . The relation-aware sequential recommendation further considers the relation between each historic item $v_i \in S_u$ and the target item v_j . The existing methods only consider the relations in \mathcal{R}_{def} , while our method LRD further

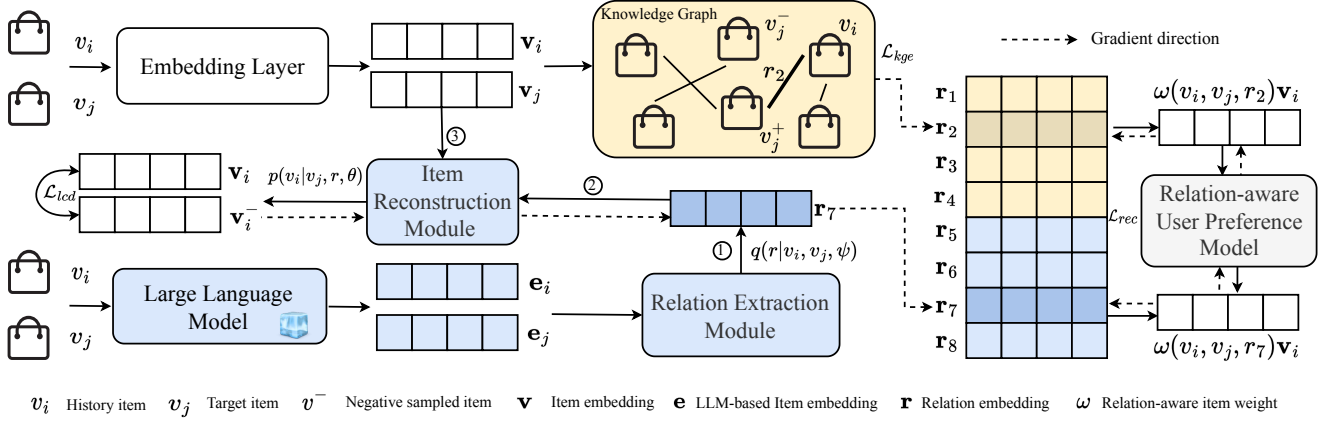


Figure 1: Overall Framework of relation-aware sequential recommendation with LRD. There are two main components of LRD: a relation extraction module to estimate the latent relation based on the language knowledge representation of two items obtained by LLM and an item reconstruction module to reconstruct the item based on the estimated relation and another item. The predefined relations from the knowledge graph and latent relations from LRD are both used to contribute to the relation-aware user preference modeling in the recommendation.

incorporates a latent relations set R_{latent} . The notations used in this paper are defined in Table 1.

3 METHOD

3.1 Framework Overview

The framework of our proposed relation-aware sequential recommendation based on latent relation discovery is illustrated in Figure 1.

The key component of the framework is the latent relation discovery module, which is designed as a self-supervised learning process inspired by the Discrete-state Variational Autoencoder (DVAE) [23]. The latent relation discovery module comprises two submodules: 1) A relation extraction module that utilizes an LLM to obtain language knowledge representations of items and predicts the latent relation between two items based on their language knowledge representations. 2) An item reconstruction module that reconstructs the representation of one item based on the representation of the predicted latent relation and the other item. Here, we incorporate the latent relation discovery module into the relation-aware sequential recommender. Specifically, we use the predicted latent relations to extend the predefined item relation embedding to better construct the user preference model. At the same time, we use the objectives of recommendation tasks to guide the discovery of more useful relations.

Next, we introduce the overall design of the latent relation discovery approach in Section 3.2, followed by the pipeline of the relation-aware sequential recommendation model based on latent relation discovery in Section 3.3.

3.2 Latent Relation Discovery (LRD)

3.2.1 Optimization Objective. Our objective is to predict the latent relation between two items. Since latent relations are those not covered by manually crafted relation datasets, we cannot train the model with supervised learning. Instead, we adopt a self-supervised

learning method inspired by DVAE. Following [23], we assume that all relations follow a uniform distribution $p_u(r)$, and the optimization objective is formalized as the following pseudo-likelihood:

$$\begin{aligned} \mathcal{L}(\theta) &= \log \sum_{r \in \mathcal{R}} p(v_i, v_{-i} | r, \theta) p_u(r) \\ &\approx \sum_{i=1}^2 \log \sum_{r \in \mathcal{R}} p(v_i | v_{-i}, r, \theta) p_u(r), \end{aligned} \quad (1)$$

where v_i and v_{-i} denote a pair of items, $i \in \{1, 2\}$. v_{-i} denotes $\{v_1, v_2\} \setminus \{v_i\}$ and $p(v_i | v_{-i}, r, \theta)$ denotes the probability of one item given another item and a relation, where θ denotes the parameters set.

The pseudo-likelihood $\mathcal{L}(\theta)$ can be lower-bounded based on Jensen's inequality through a variational posterior $q(r | v_i, v_{-i}, \psi)$:

$$\begin{aligned} \mathcal{L}(\theta) > \mathcal{L}(\theta, \psi) &= \sum_{i=1}^2 \sum_{r \in \mathcal{R}} q(r | v_i, v_{-i}, \psi) \log p(v_i | v_{-i}, r, \theta) \\ &\quad + \alpha H[q(r | v_i, v_{-i}, \psi)], \end{aligned} \quad (2)$$

where $q(r | v_i, v_{-i}, \psi)$ is the relation extraction module used to predict the relation between a pair of items and $p(v_i | v_{-i}, r, \theta)$ is the item reconstruction module that reconstructs the representation of the item given the predicted relation and another item. ψ and θ denote the parameters of the two modules, respectively. Intuitively, maximizing the probability of reconstructing the original item force $q(r | v_i, v_{-i}, \psi)$ to provide sufficiently accurate and highly generalizable relations. H is an entropy term used to regularize the probabilities predicted by the relation extraction module, ensuring more uniform predictions. α is the hyper-parameter to balance the regularization strength. Next, we further present the design of the relation extraction module $q(r | v_i, v_{-i}, \psi)$ and the item reconstruction module $p(v_i | v_{-i}, r, \theta)$ separately.

3.2.2 Relation Extraction. In the relation extraction module, we aim to predict the latent relation between two given items. Intuitively inspired by the ability of humans to describe relations between two items with natural language based on their knowledge, we investigate discovering latent relations between items from the perspective of language knowledge. Considering that LLM has exhibited human-like world knowledge and effective semantic representations, we leverage LLM to obtain item representations that potentially embed important information for discovering item relations not covered in the manually predefined relation set. We refer to these representations as language knowledge item representations and feed them into the relation extraction module.

Specifically, given an item $v = \{w_1, w_2, w_3, \dots, w_{N_v}\}$, where w_i denotes each token of the item's text. We feed the token sequence into the LLM to obtain the language knowledge representation of the item, as shown in Equation (3).

$$\mathbf{e} = W_1(LLM([w_1, w_2, w_3, \dots, w_{N_v}])) + b_1, \quad (3)$$

where $LLM(\cdot)$ denotes a specific pooling strategy on the hidden states of the LLM's last layer to obtain the output item representation. Different LLMs may use different pooling strategies, such as CLS-pooling, mean-pooling, etc [16]. $W_1 \in \mathbb{R}^{d_L \times d}$ and $b_1 \in \mathbb{R}^d$ denote the weight and bias of a projection layer, respectively, which is used to reduce the dimensionality of the LLM's output to match the input dimensions of the recommendation model.

Next, the relation extraction module $q(r|v_i, v_{-i}, \psi)$ in Equation (2) predicts the relation between two given items on the relation set \mathcal{R} based on their language knowledge representations, i.e., $\mathbf{e}_i \in \mathbb{R}^d$ and $\mathbf{e}_{-i} \in \mathbb{R}^d$. In practice, we can adopt any classifier that allows gradient backpropagation. Without loss of generality, we use a lightweight linear classifier:

$$q(r|v_i, v_{-i}, \psi) = \text{SoftMax}(W_2[\mathbf{e}_i; \mathbf{e}_{-i}] + b_2), \quad (4)$$

where $W_2 \in \mathbb{R}^{2d \times |\mathcal{R}|}$ and $b_2 \in \mathbb{R}^{|\mathcal{R}|}$ are the weight and bias of the linear classifier, respectively, and $[\cdot; \cdot]$ denotes the concatenation operation.

3.2.3 Relational Item Reconstruction. Through the relation extraction module, we estimate the latent relation between two items v_i and v_{-i} . Given the estimated relation and one of the items, the item reconstruction module aims to reconstruct the other item. The specific definition is as follows:

$$p(v_i|v_{-i}, r, \theta) = \frac{\exp(\phi(v_i, v_{-i}, r))}{\sum_{v'_i \in \mathcal{V}} \exp(\phi(v'_i, v_{-i}, r))}, \quad (5)$$

where $\phi(v_i, v_{-i}, r, \theta)$ is a scoring function for two items and a relation, and any triplet scoring function can be used. Without loss of generality, we use DistMult [44] as scoring functions:

$$\phi(v_i, v_{-i}, r) = \mathbf{v}_i^T \text{diag}(\mathbf{r}) \mathbf{v}_{-i}, \quad (6)$$

where $\text{diag}(\mathbf{r})$ is a diagonal matrix with the relation embedding \mathbf{r} as its diagonal elements. Note that we use the representation of item IDs \mathbf{v}_i for reconstruction instead of language knowledge representation \mathbf{e}_i . This design is to align the representation space of the predicted relation r and the relation-aware recommendation model.

Since Equation (5) involves calculations across the entire set of items \mathcal{V} , resulting in high computational complexity, we approximate $\log p(v_i|v_{-i}, r, \theta)$ using negative sampling as follows:

$$\log p(v_i|v_{-i}, r, \theta) = \log \sigma(\phi(v_i, v_{-i}, r, \theta)) + \log \sigma(-\phi(v_i^-, v_{-i}, r, \theta)), \quad (7)$$

where σ is the sigmoid activation function, and v_i^- is a randomly sampled negative item. Ultimately, the optimization objective in Equation (2) becomes:

$$\mathcal{L}(\theta, \psi) = \sum_{i=1}^2 \sum_{r \in \mathcal{R}} q(r|v_i, v_{-i}, \psi) [\log \sigma(\phi(v_i, v_{-i}, r, \theta)) + \log \sigma(-\phi(v_i^-, v_{-i}, r, \theta))] + \alpha H[q(r|v_i, v_{-i}, \psi)]. \quad (8)$$

3.3 LRD-based Sequential Recommendation

In this section, we present how to apply latent relations extracted by LRD into a relation-aware sequential recommender. We first introduce the pipeline of the sequential recommendation explicitly considering item relations in Section 3.3.1, followed by the joint optimization framework of the latent relation discovery and relation-aware sequential recommendation in Section 3.3.2.

3.3.1 Relation-aware Sequential Recommendation. Given a user's interaction history $S_u = \{v_1, v_2, v_3, \dots, v_{N_u}\}$ and a target item v_j , the user's preference can be reflected in the history of interacted items. We further explicitly consider the relations between each historical item and the target item for sufficient user preference modeling. Formally, the preference score of user u for target item v_j is defined as:

$$y_{u,j} = (\mathbf{u} + \mathbf{m}_{u,j}) \mathbf{v}_j^T + b_j, \quad (9)$$

where $\mathbf{u} \in \mathbb{R}^d$ and $\mathbf{v}_j \in \mathbb{R}^d$ are representations of the user and target item, $\mathbf{m}_{u,j} \in \mathbb{R}^d$ is the user's historical sequence representation considering the relations between historical items and the target item. $b_j \in \mathbb{R}^d$ is the bias term.

Calculating $\mathbf{m}_{u,j}$ is a crucial step in modeling relation-aware user preferences. Specifically, it is the aggregation of multiple user sequence representations considering different relations types, which is defined as:

$$\mathbf{m}_{u,j} = \text{AGG}([\mathbf{s}_{u,j,r_1}; \mathbf{s}_{u,j,r_2}; \dots; \mathbf{s}_{u,j,r_{|\mathcal{R}|}}]), \quad (10)$$

where AGG is an aggregation function that can adopt various aggregation methods such as mean-pooling, max-pooling, and attention-pooling [36, 43]. \mathcal{R} is the relations set, including predefined relations in the knowledge graph and latent relations discovered by the LRD. $\mathbf{s}_{u,j,r}$ is the historical sequence representation of user u given a relation r and target item v_j , defined as:

$$\mathbf{s}_{u,j,r} = \sum_{v_i \in S_u} \omega(v_i, v_j, r) \mathbf{v}_i, \quad (11)$$

where $\omega(v_i, v_j, r)$ is the relation intensity between historical item v_i and target item v_j under relation r . It is a normalized weight across historical items, defined as:

$$\omega(v_i, v_j, r) = \frac{\exp(\phi(v_i, v_j, r))}{\sum_{v'_i \in S_u} \exp(\phi(v'_i, v_j, r))}, \quad (12)$$

where $\phi(v_i, v_j, r)$ is the triplet scoring function given two items and a relation. The scoring function used here is consistent with

the one used in the item reconstruction module in Section 3.2.3, i.e., Equation (5). This alignment facilitates joint optimization in subsequent steps.

3.3.2 Joint Learning. For the relation-aware sequential recommendation task, we adopt the BPR pairwise loss [29] to define its optimization objective:

$$\mathcal{L}_{rec} = - \sum_{u \in \mathcal{U}} \sum_{j=2}^{N_u} \log \sigma(y_{u,j} - y_{u,j^-}). \quad (13)$$

To leverage discovered latent relations for the recommendation task and simultaneously let user interaction data guide the relation discovery process, we jointly optimize the objectives of the latent relation discovery task in Equation (8) and the recommendation task in Equation (13). For this purpose, we make little modifications to the Equation (8) and explicitly represent item pairs as historical items and target items, as shown below:

$$\begin{aligned} \mathcal{L}_{lrd} = - \sum_{u \in \mathcal{U}} \sum_{j=2}^{N_u} \sum_{i=1}^{j-1} \sum_{r \in \mathcal{R}} q(r|v_i, v_j, \psi) [& \log \sigma(\phi(v_i, v_j, r, \theta)) \\ & + \log \sigma(-\phi(v_i^-, v_j, r, \theta))] \\ & + \alpha H[q(r|v_i, v_j, \psi)]. \end{aligned} \quad (14)$$

Additionally, to ensure the model's capability to model predefined relations. We adopt a widely used knowledge graph embedding method to optimize the representations of items and relations in the knowledge graph, and the optimization objective is shown as:

$$\mathcal{L}_{kge} = - \sum_{(v_i, v_{-i}, r) \in \mathcal{T}} \log \sigma(\phi(v_i, v_{-i}, r) - \phi(v_i^-, v_{-i}^-, r)). \quad (15)$$

The knowledge graph embedding task is also included in the joint optimization framework. Ultimately, the joint optimization objective is defined as:

$$\mathcal{L} = \mathcal{L}_{rec} + \gamma \mathcal{L}_{kge} + \lambda \mathcal{L}_{lrd}, \quad (16)$$

where γ and λ are the coefficients of the knowledge graph embedding task and the latent relation discovery task, respectively.

4 EXPERIMENTS

In this section, we first introduce the experimental setting and then present experimental results and analyses to answer the following research questions:

- **RQ1:** How is the effectiveness of LRD-enhanced relation-aware sequential recommendation models?
- **RQ2:** Does each component of LRD-enhanced relation-aware sequential recommender contribute to the recommendation performance?
- **RQ3:** Does LRD discover reliable and vital relations between items?

4.1 Experimental Settings

4.1.1 Datasets. We conduct experiments on three publicly available datasets from diverse domains to validate the model's capability to discover latent relations in various recommendation scenarios.

Table 2: Statistics of the datasets after preprocessing.

	Datasets	MovieLens	Offices	Electronics
User-Item Interactions	#user	943	4,905	192,403
	#item	1,349	2,420	63,001
	#inter.	99,287	53,258	1,682,498
	density	7.805%	0.448%	0.014%
Item Relations	#relation	2	4	4
	#triplets	886K	778K	2,148M

- **MovieLens¹:** This dataset is widely used for movie recommendations and consists of user ratings and attribute information for movies. We utilized the MovieLens-100k version for our experiments. Two predefined relations, namely “release year” and “genre”, are extracted from the dataset. To discover latent relations between items, in addition to the text of movie title, release year, and genre available in the dataset, we crawl movie information from IMDB², including director, actor, and brief based on movie titles and release year.
- **Amazon Office Products and Electronics [9, 24]:** These two datasets are subsets from the Amazon e-commerce dataset, representing two distinct domains. They contain user ratings, reviews, and rich item metadata. We utilize two attributes (i.e., category and brand) and co-occurrence information (i.e., “also buy” and “also view”) as predefined item relations. Texts of the item title, category, and brand are used to discover latent relations between items.

For all datasets, following previous work [36], we filter users and items with fewer than 5 interactions. The statistical information for each dataset after preprocessing is presented in Table 2.

4.1.2 Baselines. We selected multiple sequential recommendation models as baselines for our experiments:

- **Caser [34]** adopts convolutional filters to capture sequential patterns of embed user sequences.
- **GRU4Rec [11]** utilizes Gated Recurrent Units (GRU) to model user representation by capturing patterns in the historical interaction sequence.
- **SASRec [15]** incorporates self-attention mechanisms to aggregate the historical item representations to obtain the user representation.
- **TiSASRec [18]** further considers time intervals between historical interactions based on the SASRec.
- **RCF [43]** adopts a two-level attention network to integrate item relations into the modeling of the user sequence representation.
- **KDA [36]** incorporates a Fourier-based temporal evolution module to capture the dynamic changes in item relations over time.

Among these methods, **Caser**, **GRU4Rec**, **SASRec**, and **TiSASRec** belong to item-based collaborative filtering models, while **RCF** and **KDA** are models that explicitly incorporate item relations. Our

¹<https://grouplens.org/datasets/movielens/100k/>

²<https://www.imdb.com/>

Table 3: Overall performance of different models. The best performances are denoted in bold fonts. “H@K” is short for “HR@K” and “N@K” is short for “NDCG@K”, respectively. The subscript “LRD” denotes the model is enhanced by LRD. “Improv.” means the relative improvement of the LRD-based model over the corresponding vanilla model. The superscripts [†] and [‡] indicate $p \leq 0.05$ and $p \leq 0.01$ for the paired t-test of the LRD-based model vs. vanilla model.

Datasets	MovieLens				Office				Electronics			
Metrics	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10
Caser	0.5217	0.6872	0.3571	0.4107	0.3095	0.4762	0.1993	0.2530	0.4620	0.5865	0.3435	0.3838
GRU4Rec	0.5101	0.6723	0.3451	0.3976	0.3295	0.4856	0.2164	0.2670	0.4699	0.5994	0.3487	0.3906
SASRec	0.5186	0.6829	0.3712	0.4242	0.4027	0.5439	0.2751	0.3210	0.4805	0.6083	0.3587	0.4000
TiSASRec	0.5313	0.6882	0.3812	0.4322	0.4014	0.5433	0.2745	0.3209	0.5114	0.6329	0.3860	0.4253
RCF	0.5101	0.6660	0.3635	0.4137	0.4145	0.5696	0.2911	0.3413	0.5790	0.7004	0.4475	0.4868
RCF _{LRD}	0.5398 [‡]	0.6882 [‡]	0.3886 [‡]	0.4365 [‡]	0.4381 [‡]	0.5761 [‡]	0.3127 [‡]	0.3573 [‡]	0.5828 [†]	0.7035 [†]	0.4510 [†]	0.4901 [†]
Improv.	+5.82%	+3.33%	+6.91%	+5.51%	+5.69%	+1.14%	+7.42%	+4.69%	+0.66%	+0.44%	+0.78%	+0.68%
KDA	0.5748	0.7381	0.4182	0.4711	0.4453	0.6145	0.3127	0.3676	0.6008	0.7194	0.4665	0.5049
KDA _{LRD}	0.6066[‡]	0.7434[‡]	0.4420[‡]	0.4867[‡]	0.4826[‡]	0.6302[‡]	0.3403[‡]	0.3881[‡]	0.6111[‡]	0.7295[‡]	0.4760[‡]	0.5143[‡]
Improv.	+5.53%	+0.72%	+5.69%	+3.31%	+8.38%	+2.55%	+8.83%	+5.58%	+1.71%	+1.40%	+2.04%	+1.86%

proposed LRD method can discover latent item relations beneficial for recommendations to enhance existing relation-aware sequential recommendation models. Therefore, we compare the performance of LRD-enhanced RCF and KDA, i.e., RCF_{LRD} and KDA_{LRD}, with the baselines above.

4.1.3 Evaluation Metrics. We use two evaluation metrics, HR@K and nDCG@K, to evaluate the performance of the models, where K is set to 5 and 10. We adopt the leave-one-out method to construct the dataset. Specifically, for a user’s interaction history sequence, we use the last item for testing, the second-to-last item for validation, and the remaining items for training. When predicting the next item, following previous work [36], we rank the ground-truth next item against 99 randomly sampled negative items.

4.1.4 Implementation Details. We implement LRD and baseline models with the ReChorus³ library. We harness a widely used LLM, i.e., GPT-3⁴, to obtain language knowledge representations of items. The max length of the history sequence is set to 20. For all models, we use the Adam optimizer and carefully search for hyperparameters, with a batch size of 256 and embedding dimension of 64. The early stop is adopted if the nDCG@5 does not improve for 10 epochs. We tune the learning rate in {1e-2, 1e-3, 1e-4} and the l2-normalization coefficients in {1e-4, 1e-5, 1e-6, 0}. The coefficients of the knowledge graph embedding task and the latent relation discovery task are tuned within {0.1, 1, 5, 10}. The number of latent relations to discover is tuned between {5, 6, 7, 8, 9, 10}. The regularization coefficient of the relation extraction module of LRD is set to 0.1. The code of our implementation is available at: <https://github.com/ysh-1998/LRD>.

4.2 Performance Comparison (RQ1)

We compare two LRD-enhanced relation-aware sequential recommendation models, namely RCF_{LRD} and KDA_{LRD}, with baseline methods. The overall experimental results are presented in Table 3.

³<https://github.com/THUwangcy/ReChorus/tree/master>

⁴We use a version tailored for text embedding, i.e., text-embedding-ada-002

For baseline methods, several observations can be made. Firstly, traditional methods perform significantly worse than relation-aware methods in highly sparse datasets, i.e., Amazon. While the performance gap is not significant in the relatively dense dataset, i.e., MovieLens. This demonstrates that traditional methods heavily rely on rich interaction data. Secondly, RCF and KDA achieve significantly better performance than non-relation-aware sequential recommendation models by incorporating item relations into the sequential recommendation. This indicates that explicitly modeling relations between historical items and the target item contributes to the user sequence modeling by capturing sufficient user preference. Thirdly, TiSASRec and KDA give superior performance by utilizing modules that model temporal information based on SASRec and RCF, respectively. This demonstrates that considering temporal factors in modeling user interaction sequences can effectively enhance model performance.

The RCF_{LRD} and KDA_{LRD} achieve significant performance improvements over the vanilla models and outperform almost all non-relation-aware methods. We attribute these improvements to the fact that previous relation-aware sequential recommendation methods rely on predefined relations, limiting the model’s ability to capture diverse item relations. Especially on datasets with fewer predefined relations, i.e., MovieLens, the performance of RCF is not significantly better than traditional methods. While the capability of RCF effectively improves by leveraging our proposed LRD method. Moreover, the performance improvement of KDA_{LRD} suggests that the discovered latent relations by LRD also exhibit temporal evolution characteristics.

4.3 Ablation study (RQ2)

In this section, we investigate the contributions of each component of our proposed LDR-based relation-aware sequential recommendation model to the recommendation performance. For this purpose, we design two variant models: (1) w/o LLM, where item ID representations replace the language knowledge representation obtained through LLM to investigate whether language knowledge is crucial

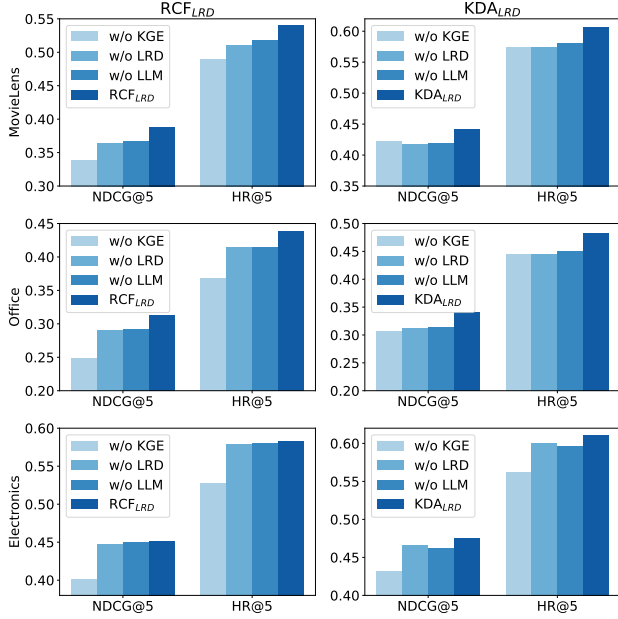


Figure 2: Ablation Study on variant models

in the relation discovery process. (2) w/o KGE, which removes the knowledge graph embedding task, meaning the model cannot optimize the representation of predefined relations utilizing supervised signals in the knowledge graph. The vanilla model without the LRD can also be seen as a variant model, i.e., w/o LRD.

Figure 2 presents the comparative results of the variant models, revealing several findings. Firstly, removing the LLM for obtaining item language knowledge representations leads to a noticeable performance decline. This indicates that ID representations lacking rich semantic information have limited effectiveness in the process of discovering latent relations, emphasizing the crucial role of LLM’s rich world knowledge. Secondly, the variant model without the KGE task shows a significant performance decrease, highlighting the necessity of maintaining the modeling of predefined relations while utilizing latent relations.

4.4 Latent Relation Analyses (RQ3)

To further validate the effectiveness and reliability of the latent item relations discovered by LRD, we perform additional analysis experiments.

4.4.1 Relation Embeddings. To investigate the distribution of the learned relation embeddings, we select the model with the best performance on the Office dataset, i.e., KDA_L_{CD}, and calculate the pair-wise cosine similarity of the relation embeddings. This model has 12 relations, including predefined relations (1-4) and latent relations (5-12). The similarity matrix is shown in Figure 3. We can find that there is a notable difference between the embeddings of predefined relations and latent relations. This is reasonable as the learning of predefined relation embeddings primarily relies on supervised signals from the knowledge graph, while the learning of

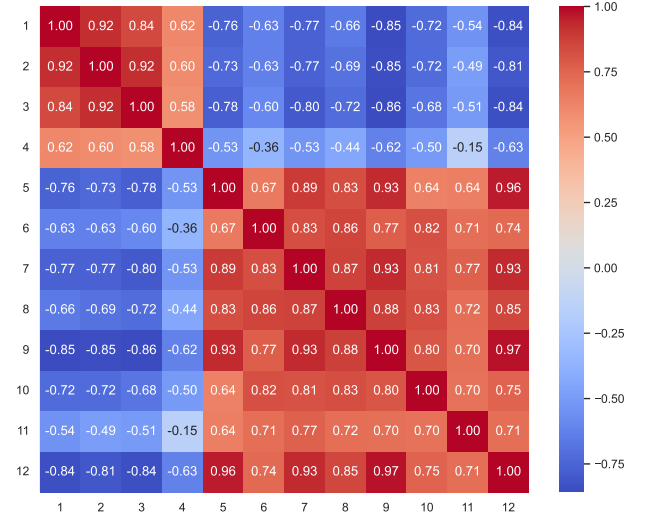


Figure 3: The pair-wise cosine similarity of the relation embeddings of KDA_L_{CD} on Office dataset. The labels on the horizontal and vertical axes denote the relation IDs, where 1-4 are predefined relations and 5-12 are latent relations learned by LRD.

latent relation embeddings depends on the latent relation discovery task. It is worth noting that the relation extraction module of LRD also predicts predefined relations for the reconstruction of relational items. This indicates that the joint optimization of the latent relation discovery task and the knowledge graph embedding task helps the model effectively distinguish predefined relations from latent relations during the learning process.

4.4.2 Latent Relational Items. To provide a clearer illustration of the learned item relations, we present representative item pairs of several relation types in Figure 4. These pairs are obtained by sorting the model’s scores for these item pairs on these relations, i.e., the selected item pairs are samples with higher scores under each relation. To facilitate readability, we have simplified the item descriptions. Several observations can be made from Figure 4. Firstly, items in relation #1 generally exhibit complementary functionalities and are frequently purchased together. This indicates that the model effectively learns predefined relations from the knowledge graph, i.e., the “also buy” relation. Secondly, the model groups item pairs with common characteristics into the same latent relation type. These relations are more complex than predefined relations, possibly involving multi-hop connections and demonstrating characteristics tailored to specific scenarios and tasks. The relation #5 reflects the relations between items in document processing, editing, and storage scenarios. Items like “Inkjet Printers” and “Expanding File Jackets & Pockets”, though not directly related, can establish a two-hop relation through “printing paper”, completing a document processing procedure. Relation #8 focuses on tasks related to transportation. Items like “Transparent Tape” and “Cart, Chrome Shelving” are related to the task of packing and transporting goods.

These examples further demonstrate the reliability of the latent relations discovered by LRD and confirm the conclusion drawn in

#1	Item1	Desktop Staplers	Inkjet Printer	Copy & Multipurpose Paper
	Item2	Staples	Inkjet Printer Ink	Ballpoint Pens
#5	Item1	Inkjet Printers	Laser Printers	Porous-Point Pens
	Item2	Expanding File Jackets & Pockets	Ballpoint Pens	Hanging Folders
#8	Item1	Transparent Tape	Shipping Labels	Label Makers
	Item2	Cart, Chrome Shelving	Security Lock Boxes	Envelope

Figure 4: Descriptions of representative item pairs belong to relation #1, #5 and #8 of KDA_{LCD} on Office dataset, where relation #1 is “also buy”, relation #5 and #8 are learned latent relations.

Section 4.4.1 that latent relations differ significantly from predefined relations. Leveraging these more complex and diverse relations, LRD-based relation-aware sequential recommendation models effectively capture more intricate user preferences, resulting in improved recommendation performance.

4.4.3 Case Study. To further demonstrate how the model leverages latent relations to achieve improved recommendation performance, we present a case in Figure 5. In this case, we record the triplet scores between each historical item and the target item on all relations. The relation with the highest score is considered as the predicted relation between items. As shown in Figure 5, only one predefined relation exists between historical items and the target item, i.e., “also buy.” Relying solely on this predefined relation, KDA fails to rank the target item at a high position. While LRD-based model KDA_{LRD} identifies latent relations #5 and #12 between items 1 and 3, and the target item, respectively. Considering the findings from Sections 4.4.1 and Section 4.4.2, latent relations #5 and #12 exhibit similar embeddings and both reflect item relations in the document processing scenario. Specifically, in this case, the printout from the “Wireless All-in-One Printer” and the document recorded with the “Pen-Grip Fine Point Pen” both can be stored in the “Folder”. Benefiting from the discovered latent relations by LRD, KDA_{LRD} ranks the target item in the 4th position, significantly outperforming the vanilla model KDA.

4.5 Hyper-parameter Sensitivity

In this section, we perform a sensitivity analysis on two crucial hyperparameters of LRD: the number of latent relations the model aims to discover, i.e., Num_latent, and the coefficient of the latent relation discovery task in the joint optimization, i.e., λ . We aim to analyze the effect of hyperparameter selection on model performance. Firstly, we examine the sensitivity of the model performance to different values of num_latent. Note that to isolate the impact of the two hyperparameters, we present the average performance

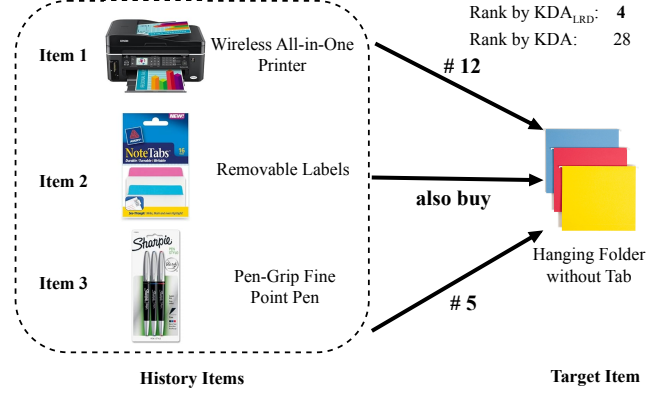


Figure 5: The interaction history of user A3N4VTNFPMTHEF on Office dataset. KDA_{LCD} ranks the target at a high position leveraging the learned latent relation between historical items and target item, i.e., relation #5 and #12, which outperforms KDA solely utilizing predefined relation, i.e. “also buy”.

under the same num_latent value across all λ values. Figure 6 illustrates the model performance under various num_latent values and several observations can be made. Firstly, the model exhibits significant performance differences under different num_latent values, indicating that both learning an insufficient number of latent relations and learning redundant or useless relations impair the model performance. Secondly, the overall trends of performance changes with num_latent are generally consistent across different models on the same dataset. While the optimal num_latent value varies for different datasets. Specifically, the optimal value for MovieLens is 6, while for the Office dataset is 5. This aligns with the assumption that there are differences in item relations across diverse recommendation scenarios. Next, we analyze the hyperparameter λ . The performance shown in Figure 6 represents the average metrics under the same λ value across all num_latent values. We observe a stable improvement in model performance until the optimal value is reached, which is generally consistent on the same dataset for two models. That demonstrates the effectiveness of the latent relation discovery task.

5 RELATED WORK

5.1 Sequential Recommendation

In the literature of recommender systems, sequential recommendation is a widely researched task, aiming to recommend items that interest users based on their historical interactions [40]. Earlier works adopt the Markov chain to model item transition relations in the interaction history of users [8, 30]. In recent years, with the development of deep learning methods, various deep neural networks have been proposed to capture user preferences in historical interaction sequences. It includes Recurrent Neural Networks (e.g., GRU [11], LSTM [41] and HRNN [27]), Convolutional Neural Networks [34, 45], Attention-based Network [15, 17, 33], and Graph Neural Networks [5, 42]. The core idea of these methods is to

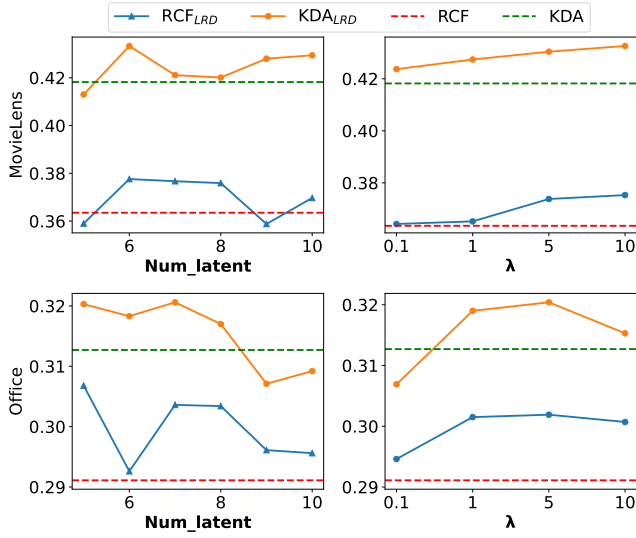


Figure 6: nDCG@5 comparison w.r.t. the number of latent relations and the coefficient of the latent relation discovery task, i.e., λ .

capture item-based collaborative filtering information in historical interactions, overlooking explicit relations between items, which is crucial for understanding user behavior and extracting user preferences. The proposed approach in this paper can effectively uncover relations among items, thus providing a more accurate modeling of user preferences.

5.2 Relation-aware Recommendation

In contrast to traditional recommendation methods that only consider item-based collaborative filtering similarity, relation-aware recommendation methods explicitly incorporate relations between items into the recommendation model. One line of methods constructs a knowledge graph containing relations between items and relations between items and attributes [1, 4, 22, 38, 39, 46]. These methods enhance item representations through knowledge graph embedding tasks. CKE [46] constructs a knowledge graph containing items and attributes and optimizes graph embedding tasks and recommendation tasks jointly. CFKG [1] further incorporates users into the knowledge graph. It defines a special “purchase” relation as the proxy of the interactions between users and items to transform the recommendation task into the knowledge graph embedding task. Another research direction focuses on sequential recommendation scenarios and explicitly models relations between historical items and the target item [36, 37, 43]. RCF [43] proposes a two-level attention framework to compute the user’s attention to relation types and the relation intensity between historical items and the target item separately. Furthermore, considering the evolution of item relations over time, KDA [36] proposes a Fourier-based temporal evolution module to incorporate time information into the modeling of relational items. Nevertheless, existing methods rely on manually predefined relations and suffer from relation sparsity and item sparsity issues. In this paper, we propose to uncover latent

relations among items, enabling the model to adapt to diverse and complex recommendation scenarios.

5.3 Large Language Model for Recommendation

Recently, the emergence of Large Language Models (LLMs) [3, 25, 26, 35, 47] has revolutionized the field of natural language processing. The performance of various natural language processing tasks has significantly improved with the reasoning abilities and world knowledge of LLMs [14, 48]. The application of LLMs to recommendation tasks has been widely researched in two main research directions [6]. One line of work leverages LLMs’ reasoning capabilities and knowledge to solve various recommendation tasks. Liu et al. [19] utilize ChatGPT [25] to perform five recommendation tasks in zero-shot and few-shot settings, achieving impressive results in explanation generation tasks while performing poorly in the direct recommendation and sequential recommendation tasks. P5 [7] transform various recommendation tasks into natural language format and feed them into T5 [28]. Then the model is fine-tuned with the language modeling training objective and exhibits performance improvement on five recommendation tasks. Another line of research investigates combining recommendation models with LLMs. TALLRec [2] constructs instruction-tuning samples based on the user rating data and fine-tune LLaMA [35] with parameter-efficient-tuning, i.e., LoRA [12], achieving promising performance on few-shot recommendation scenario. ONCE [20] utilizes ChatGPT as a data augementer to acquire knowledge-enhanced representations of users and news, improving the performance of the news recommendation. However, existing recommendation methods with LLMs have significant limitations in terms of performance and efficiency. In this paper, we leverage the rich world knowledge of the LLM to acquire language knowledge representations of items, which is used to discover item relations that contribute to recommendations. We fully utilize the knowledge of the LLM and ensure the efficiency of the overall framework.

6 CONCLUSION

In this paper, we propose a novel method for discovering latent item relations based on the Large Language Model, namely LRD. Leveraging the rich world knowledge of LLM and a self-supervised learning approach, LRD effectively extracts latent item relations. We jointly optimize LRD with existing relation-aware sequential recommender systems. On the one hand, the latent relations discovered by LRD provide more sophisticated item associations, contributing to the sufficient modeling of intricate user preference. On the other hand, the supervision signals from user interactions guide the relation discovery process effectively. Experimental results on multiple public datasets demonstrate that LRD significantly improves the performance of existing relation-aware sequential recommendation methods. Further analyses demonstrate the reliability of the latent relations. Note that the crucial LLM in our method is not meticulously selected in the current implementation. It leaves us a future work to explore the performance of our method utilizing more advanced LLMs.

REFERENCES

- [1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*

- 11, 9 (2018), 137.
- [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 *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. ACM. <https://doi.org/10.1145/3604915.3608857>
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [4] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*. 151–161.
- [5] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 378–387.
- [6] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2023. Recommender Systems in the Era of Large Language Models (LLMs). [arXiv:2307.02046](https://arxiv.org/abs/2307.02046) [cs.IR]
- [7] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). [arXiv:2203.13366](https://arxiv.org/abs/2203.13366) [cs.IR]
- [8] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.
- [9] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee. <https://doi.org/10.1145/2872427.2883037>
- [10] Zhankui He, Handong Zhao, Zhe Lin, Zhaowen Wang, Ajinkya Kale, and Julian McAuley. 2021. Locker: Locally constrained self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3088–3092.
- [11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. [arXiv:2106.09685](https://arxiv.org/abs/2106.09685) [cs.CL]
- [13] Seonwon Jang, Hoyeop Lee, Hyunsook Cho, and Sehee Chung. 2020. Cities: Contextual inference of tail-item embeddings for sequential recommendation. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 202–211.
- [14] Ehsan Kamalloo, Nouha Dziri, Charles LA Clarke, and Davood Rafiei. 2023. Evaluating Open-Domain Question Answering in the Era of Large Language Models. *arXiv preprint arXiv:2305.06984* (2023).
- [15] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [16] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. [arXiv:2011.05864](https://arxiv.org/abs/2011.05864) [cs.CL]
- [17] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 1419–1428.
- [18] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.
- [19] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149* (2023).
- [20] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. ONCE: Boosting Content-based Recommendation with Both Open- and Closed-source Large Language Models. [arXiv:2305.06566](https://arxiv.org/abs/2305.06566) [cs.IR]
- [21] Zhiding Liu, Mingyue Cheng, Zhi Li, Qi Liu, and Enhong Chen. 2022. One Person, One Model—Learning Compound Router for Sequential Recommendation. In *2022 IEEE International Conference on Data Mining (ICDM)*. 289–298. <https://doi.org/10.1109/ICDM54844.2022.00039>
- [22] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The world wide web conference*. 1210–1221.
- [23] Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics* 4 (2016), 231–244.
- [24] Julian McAuley, Christopher Targett, Qin Feng Shi, and Anton van den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. [arXiv:1506.04757](https://arxiv.org/abs/1506.04757) [cs.CV]
- [25] OpenAI. 2023. GPT-4 Technical Report. [arXiv:2303.08774](https://arxiv.org/abs/2303.08774) [cs.CL]
- [26] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.
- [27] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*. 130–137.
- [28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. [arXiv:1910.10683](https://arxiv.org/abs/1910.10683) [cs.LG]
- [29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [30] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
- [31] Deepjyoti Roy and Mala Dutta. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data* 9, 1 (2022), 59.
- [32] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.
- [33] 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*. 1441–1450.
- [34] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.
- [35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [36] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2020. Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2020), 1–33.
- [37] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 109–118.
- [38] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.
- [39] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference*. 2000–2010.
- [40] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).
- [41] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.
- [42] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.
- [43] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 125–134.
- [44] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. [arXiv:1412.6575](https://arxiv.org/abs/1412.6575) [cs.CL]
- [45] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2018. A Simple Convolutional Generative Network for Next Item Recommendation. [arXiv:1808.05163](https://arxiv.org/abs/1808.05163) [cs.IR]
- [46] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.
- [47] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).
- [48] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848* (2023).