Enhancing User Interest Modeling with Knowledge-Enriched Itemsets for Sequential Recommendation

Chunyang Wang, Yanmin Zhu*, Haobing Liu, Wenze Ma, Tianzi Zang, Jiadi Yu
Shanghai Jiao Tong University,
{wangchy, yzhu, liuhaobing, mawenze991226, zangtianzi, jiadiyu}@sjtu.edu.cn

ABSTRACT

Sequential recommendation which aims to predict a user's next interaction based on his/her previous behaviors, has attracted great attention. Recent studies mainly employ deep recurrent neural networks or self-attention networks to capture dynamic user preferences. However, existing methods merely focus on modeling users' clear interests in interacted items. We argue that for an interaction, the user may also have ambiguous interests in items that are semantically related to the interacted one. For comprehensively capturing user preferences, it is beneficial to discover potential interests from historical interactions at a broader itemset level. Therefore, in this paper, we propose a knowledge graph enhanced sequential recommendation model namely KGIE, which focuses on enhancing user interest modeling with knowledge-enriched itemsets by incorporating the knowledge graph. Specifically, in addition to item-level interest modeling with interacted items, we further construct knowledge-enriched itemsets that are extracted via high-order knowledge associations with the interacted items. For capturing personalized itemset-level interests, we design an attentive aggregation unit to combine item embeddings considering both inherent and contextual personalization signals. Furthermore, to balance the contributions of both two levels of interest modeling, we adaptively learn high-level preference representations with a gating fusion unit. Extensive experiments on three real-world datasets demonstrate the superior performance beyond state-of-the-art methods and recommendation interpretability of our model.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Sequential Recommendation; Knowledge Graph; Interest Modeling

ACM Reference Format:

Chunyang Wang, Yanmin Zhu, Haobing Liu, Wenze Ma, Tianzi Zang, and Jiadi Yu. 2021. Enhancing User Interest Modeling with Knowledge-Enriched Itemsets for Sequential Recommendation. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482256

*Corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11...$15.00
https://doi.org/10.1145/3459637.3482256

1 INTRODUCTION

In recent years, the great success of recommender systems has been witnessed in many online applications, such as e-commerce, social media, entertainment services, etc. In many real-world scenarios, users' interests are intrinsically dynamic and evolve with their interactions on items. For example, users might be more interested in *AirPods* after just purchasing an *iPhone*. To accurately capture dynamic user preferences, sequential recommendation leverages the recent sequence of users' interacted items and recommends the next item. It is beneficial for providing accurate recommendations by considering the sequential order of historical interactions [23].

Existing methods for sequential recommendation could be mainly divided into three categories. The first category uses Markov Chains to capture sequential patterns from historical interaction behaviors [8, 26]. The models of this category only estimate short-term item transitions based on the most recent interaction, which leads to poor performance. The second category exploits Recurrent Neural Networks (RNNs) to capture long-term dependencies within the whole interaction sequence [1, 10]. These methods encode the sequence into a hidden representation step by step, which results in low computational efficiency. The third category of methods adopts self-attention mechanisms to capture pairwise dependencies among interacted items [16, 19, 36, 41, 45]. Due to good effectiveness and computation efficiency, self-attention networks based methods show considerable abilities to handle interaction sequences. However, existing works mainly capture user preferences based on the assumption that users always have *clear interests* in interacted items. We argue that there exist situations that users have *ambiguous interests* in multiple items simultaneously and finally interact with one of them arbitrarily. Such situations are quite common in real scenes but are largely neglected by existing methods when modeling user interests from historical interactions.

Considering the potential ambiguity of user interests, we emphasize the importance of expanding user interests modeling beyond interacted items, i.e., the **item level**, to sets of other related items, i.e., the **itemset level**. Although we could explicitly observe an interaction between a user and a certain item, it is inappropriate to assert that the user only has clear interests in the item. There are circumstances that users are likely to have ambiguous interests over a set of related items instead of focusing on the interacted one. Give an illustrative example of the movie domain in Figure 1. According to the user's real intentions, he/she is simultaneously interested in a set of movies of the *Romantic* genre and therefore arbitrarily interacts with two movies *One Day* and *Titanic* successively. In

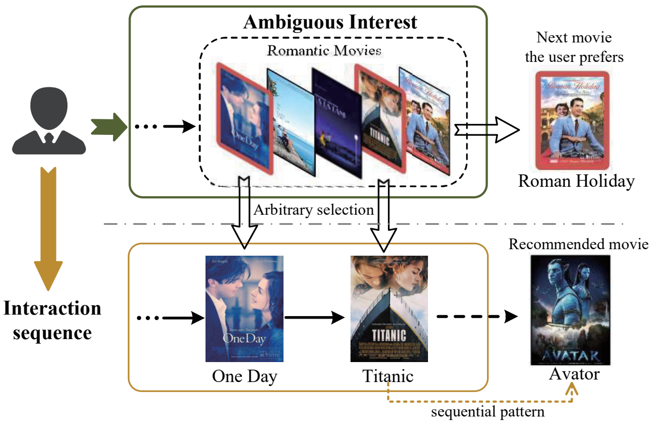


Figure 1: An illustrative example showing ambiguous user interests in the movie domain. In the user's intentions, he/she might have ambiguous interests in a group of movies instead of clear interests in a specific movie.

such situations, paying too much attention to the interacted items may result in recommendations closely related to those items themselves. For example, *Avator* might be recommended according to item-level sequential patterns with the watched *Titanic*. But in fact, another romantic movie *Roman Holiday* which exists in the same movie set could satisfy the user's current interests better. Therefore, for effectively discovering ambiguous interests hidden from historical interactions, it is beneficial to expand user interest modeling over broader itemsets beyond interacted items.

Based on the insights above, we aim to explore user interest modeling over itemsets consisting of semantically related items of interacted items. To realize this, we incorporate knowledge graphs as auxiliary information to provide item associations. However, there are still two major challenges to be addressed. First, for an interacted item, *how to properly highlight semantically related items to perform itemset-level interest modeling remains a challenge.* Recent studies on knowledge graph based recommendation simply regard items and attributes as homogeneous entities and enrich item representations with embeddings of all neighbors [33, 35, 37]. This is insufficient to concentrate on discovering potential interests in the semantically related items of the interacted items. Thus, itemset-level interest modeling should be conducted in a more explicit manner. Second, *how to effectively infer personalized itemset-level interests for different interactions is also challenging*. For the same itemset, different users are likely to have personalized preferences for items in it. Even for the same user, his/her preferences for the same itemset are usually dynamic in different contexts. Therefore it necessitates a design to infer personalized itemset-level interests by considering the personalization signals of each interaction.

To tackle the aforementioned challenges, we propose a Knowledge Graph based Interest modeling Expansion model namely KGIE for sequential recommendation. KGIE aims to expand user interest modeling with knowledge-enriched itemsets that are relevant to interacted items. For capturing comprehensive user preferences, both item-level and itemset-level interest modeling are conducted based on historical interactions. Specifically, at the item level, we adopt knowledge graph convolution layers for enhancing the representations of interacted items with knowledge information. At the itemset level, firstly, we explicitly construct knowledge-enriched itemsets for different items by leveraging high-order connectivities in the knowledge graph. Then we design a personalization-aware attentive aggregation unit to learn personalized itemset-level interest representations of different interactions. Two types of personalization signals including inherent and contextual user preferences are extracted for each interaction. To adaptively combine two levels of interest modeling, we devise a gating fusion unit to learn a high-level preference representation for each interaction. Finally, based on the sequence of preference representations, we apply multi-head self-attention layers to capture the evolution of user preferences.

The contributions of our work are summarized as follows:

- We propose a novel knowledge graph based sequential recommendation model namely KGIE, which focuses on modeling user interests at the item-level as well as the itemset-level for comprehensively capturing user preferences.
- We propose to expand user interest modeling from interacted items to knowledge-enriched itemsets. We design a personalization-aware attentive aggregation unit to capture personalized itemset-level interests and a gating fusion unit to adaptively combine both item-level and itemset-level interest representations.
- We conduct extensive experiments on three real-world datasets, which show that our model consistently outperforms state-of-the-art methods under different metrics. We also present the recommendation interpretability of our model.

2 RELATED WORK

2.1 Sequential Recommendation

The sequential recommendation task aims to predict subsequent interactions of a user given the sequence of his/her past accessed items. Early Markov Chain (MC) based methods mainly capture short-term sequential patterns by estimating the item transition probability matrix [8, 26]. For instance, FPMC [26] applies personalized MCs to learn single-step item transitions. Further, Fossil [8] combines an item similarity model with high-order MCs that consider multiple previous interactions. However, these methods only model transitions among a few consecutive interactions but fail to capture long-term preferences from the whole sequence. Recently, deep learning methods for sequence learning such as recurrent neural networks (RNNs) and attention-based methods show great effectiveness in handling long-range sequences [21, 32, 40, 43]. As a simple and effective variant of RNN, the gated recurrent unit (GRU) [4] and its various extensions have been introduced for sequential recommendation [10, 18, 24, 30, 42].

Besides, the self-attention networks (SANs) [31] based methods also achieve promising performance as well as efficiency improvement [3, 14–16, 19, 20, 22, 28, 41]. For example, SASRec [16] firstly applies the self-attention mechanism in sequential recommendation model to capture pairwise dependencies in the item sequence. Later, GC-SAN [41] strengthens the self-attention network with graph neural network (GNN) to capture local dependencies that exist over adjacent interactions. BERT4Rec [28] utilizes bidirectional self-attention architecture to encode historical interaction sequences. TiSASRec [19] additionally models time intervals between sequential interactions to learn attention weights. To reveal sequential patterns between features of interacted items, FDSA [45] applies parallel self-attention blocks on both item sequences and feature sequences. However, these methods neglect the situations that users are likely to have broader interests beyond these interacted items. Differently, our work additionally models users' potential interests based on knowledge-enriched itemsets which are semantically related to historical interactions.

2.2 Knowledge Graph based Recommendation

In recent years, knowledge graphs (KGs) have been introduced into recommender systems for providing rich item side information [5]. As we know, Zhang et.al firstly [44] propose to combine collaborative filtering and knowledge graph embedding methods (e.g., TransE [2]) to enhance recommendation performance. Recent studies mainly make efforts to incorporate item knowledge via designed paths [11, 13, 29, 38, 47] or embedding propagation [33, 35, 37, 39]. For example, KARN [47] exploits knowledge graphs to search extra connectivity paths between users and candidate items. To enrich user preference representations, Ripplenet [33] proposes to perform preference propagation with KG starting from users' interaction history. Following the idea of embedding propagation, Wang et.al propose KGCN [35] and its follow-up approach KGCN-LS [34] to iteratively aggregate semantic information from multi-hop neighbors of candidate items. By integrating the user-item interaction relation into the original KG, KGAT [37] explores the propagation mechanism in a collaborative KG. Similarly, CKAN [39] combines collaboration signals and knowledge associations through two separate steps of embedding propagation. These methods mainly deal with conventional recommendation tasks which overlook the sequential order of interactions, while our work focuses on sequential recommendation to capture dynamic user preferences.

For knowledge graph enhanced sequential recommendation, Huang et.al [13] propose to extract semantic paths from knowledge graph between each user-item pair and encode paths to enrich interaction representations. However, it's hard to guarantee that each user-item pair can be connected through paths within a limited length in practice. Another state-of-the-art approach, KSR [12], utilizes external memory networks to refine attribute-level preference representations for enhancing a GRU-based sequential recommender. This work extracts relevant attributes of interacted items from knowledge graphs, but neglects high-order knowledge associations among items. As we discussed above, potential interests in semantically related items should be emphasized to some extent. In our work, we explicitly construct knowledge-enriched itemsets that are searched via high-order knowledge graph connections with the interacted items and capture personalized itemset-level interests for each interaction.

3 PROBLEM FORMULATION

In this section, we formulate the knowledge graph based sequential recommendation problem as follows.

Interaction Sequence. In a typical sequential recommendation scenario, let $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ denotes the set of all users and $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ denotes the set of all items. The historical intercation sequence of user u is defined as $S^u = \{s_1^u, s_2^u, ..., s_n^u\}$, where n is the number of items that the user u has interacted with. The items in S^u are sorted by time in ascending order, i.e. s_t^u represents the t-th interacted item of user u.

Knowledge Graph. Besides historical interaction records of users, a knowledge graph $\mathcal{G}$ is introduced for providing item side information. The knowledge graph organizes the relationships among real-world entities as triples denoted as $\{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ is the set of entities and $\mathcal{R}$ is the set of diverse relations. In different recommendation scenarios, knowledge graphs reflect well-known facts about items of corresponding fields. The items for recommendation are aligned according to $\mathcal{A} = \{(v, e)|v \in \mathcal{V}, e \in \mathcal{E}\}$, where e could be either head or tail entity in the triples.

Problem Statement. Given the interaction sequence $S^u = \{s_1^u, s_2^u, ..., s_n^u\}$ as well as knowledge graph $\mathcal{G}$, the knowledge graph based sequential recommendation task is to predict which item the user u will interact with (e.g., click, buy) next time, i.e., s_{n+1}^u. Specifically, the goal is to predict the probability that the user u will interact with each candidate item v. Further, we generate the recommendation list containing top-K items by ranking the interaction probabilities of all candidate items in descending order.

4 KNOWLEDGE GRAPH BASED INTEREST MODELING EXPASION

In this section, we introduce the proposed Knowledge Graph based Interest modeling Expansion model (KGIE) shown in Figure 2. In what follows, we elaborate on the main components of our model, including: 1) knowledge-enhanced item-level interest modeling, which enriches inherent item representations with attribute information encoded in the knowledge graph; 2) itemset-level interest modeling, which explores potential interests on itemsets that are semantically related to interacted items; 3) sequential interactions modeling, which adaptively combines representations of two levels of user interests and captures the evolution of user preferences.

4.1 Item-level Interest Modeling

In addition to the identity of an item, associated attributes could also appeal to users' interests in the item. In other words, attribute information could reflect the user's fine-grained preferences for his/her interacted items. Therefore, for enhancing the ability to model user interests in interacted items, i.e., at the item level, we enrich item representations of all historical interactions by extracting attribute information encoded in the knowledge graph.

For enriching the representation of an interacted item s_i, we aggregate entity embeddings of its local neighbors $\mathcal{N}_{e_i}$ in a knowledge graph. The $\mathcal{N}_{e_i}$ contains neighboring entities of the aligned item entity e_i in the knowledge graph, where $(s_i, e_i) \in \mathcal{A}$. We can obtain $\mathcal{N}_{e_i} = \{t|(e_i, r, t) \in \mathcal{G}\}$ along with knowledge graph connections. Following Wang et.al [35], to keep computation efficient when performing embedding aggregation, we uniformly sample a fixed number L of neighbors, i.e., $|\mathcal{N}_{e_i}| = L$.

Considering the diversity of relations, we perform relation-aware embedding aggregation to incorporate attribute information into the item entity embedding $\mathbf{e}_i$. Specifically, similar to knowledge graph convolution operation in [35], we firstly refine a neighborhood representation vector $\boldsymbol{e}_{\mathcal{N}_{e_i}}$ using embeddings of neighbors

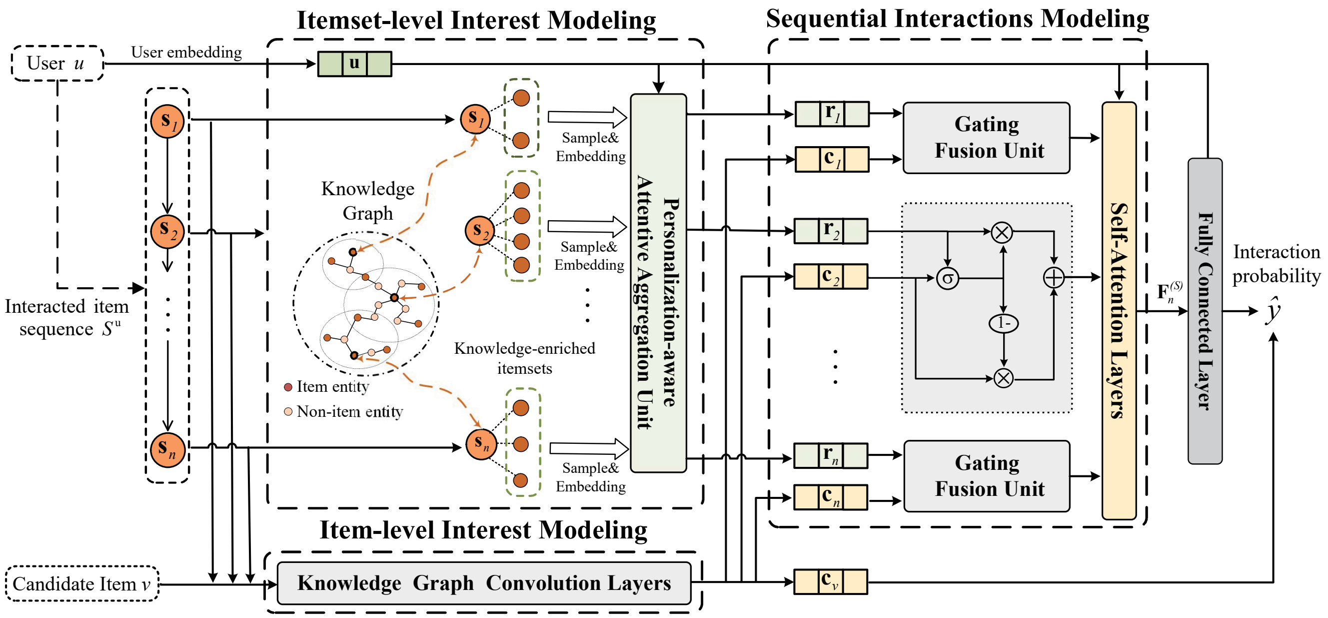


Figure 2: The overall architecture of the proposed KGIE model. Both item-level and itemset-level user interest modeling are conducted based on the interaction sequence. Personalized itemset-level interest representations are learned from knowledge-enriched itemsets of interacted items. Two levels of user interest modeling are adaptively combined with a gating fusion unit. Finally, the interaction sequence is handled with self-attention layers to capture the evolution of user preferences.

$\mathcal{N}_{e_i}$ according to following equations:

$$\mathbf{e}_{\mathcal{N}_{e_i}} = \sum_{e_j \in \mathcal{N}_{e_i}} p(\mathbf{u}, \mathbf{r}_j)\mathbf{e}_j, \tag{1}$$

$$p(\mathbf{u}, \mathbf{r}_j) = \frac{\exp(\sigma(\mathbf{u}\mathbf{W}_p\mathbf{r}_j))}{\sum_{e_k \in \mathcal{N}_{e_i}} \exp(\sigma(\mathbf{u}\mathbf{W}_p\mathbf{r}_k))}, \tag{2}$$

where $\mathbf{W}_p \in \mathbb{R}^{d\times d}$ are learnable parameters. The $\mathbf{e}_j$ and $\mathbf{r}_j \in \mathbb{R}^d$ are embeddings of one neighboring entity and its relation with e_i. By integrating the inherent user embedding $\mathbf{u} \in \mathbb{R}^d$, a relation-aware preference score $p(\mathbf{u}, \mathbf{r}_j)$ is calculated with an attention mechanism to indicate the user's preference for the specific relation. Next, the $\mathbf{e}_{\mathcal{N}_{e_i}}$ is utilized to update the entity representation $\mathbf{e}_i$ as follows:

$$\hat{\mathbf{e}}_i = \mathrm{ReLU}(\mathbf{W}_{agg}(\mathbf{e}_{\mathcal{N}_{e_i}} + \mathbf{e}_i) + \mathbf{b}_{agg}). \tag{3}$$

where $\mathbf{W}_{agg} \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_{agg} \in \mathbb{R}^d$ are learnable parameters of the aggregation function. Then $\hat{\mathbf{e}}_i$ is utilized as the enhanced item entity representation, i.e, $\mathbf{e}_i \leftarrow \hat{\mathbf{e}}_i$.

Since a single graph convolution layer which unites equation (1-3) gathers one-hop neighborhood information, we allow stacking K layers to recursively aggregate richer semantic information from multi-hop neighbors. It should be noted that incorporating embeddings of high hop neighborhoods has the risk of bringing huge noise, which may disturb the original item representation. We will discuss the influence of settings for K in section 5.4. Finally, the knowledge enhanced representations of all interacted items is obtained as $\{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}$, where $\mathbf{c}_i = \mathbf{e}_i^{(K)}$ and $\mathbf{e}_i^{(K)}$ is the updated representation of item s_i obtained with the K-th shared knowledge graph convolution layer.

4.2 Itemset-level Interest Modeling

As we mentioned before, it is deficient to understand users' preferences by merely focusing on the interacted items. For an interaction, the user might pay similar attentions to other related items instead of the interacted one. Therefore, besides item-level interests in interacted items, potential interests in semantically related items should also be considered. To address this factor, we propose to additionally expand user interest modeling to the itemset level for comprehensively capturing user preferences.

4.2.1 Knowledge-enriched Itemsets Construction. Built with real-world facts about items, knowledge graphs can objectively describe semantic associations among items. Therefore, we aim to search for semantically related items of interacted items to construct knowledge-enriched itemsets. Specifically, an knowledge-enriched itemset $\mathcal{I}_i$ for an interacted item s_i is constructed by searching along with knowledge graph connections in the following two steps.

First, we search for all item entities which connect to e_i with no more than M hops in the knowledge graph as a candidate pool. For instance, assume that there exists a path $[e_i, e_{a_1}, e_{i_1}, e_{a_2}, e_{i_2}]$ where e_{i_*} and e_{a_*} denotes item entities and non-item entities, respectively. In such situations, both e_{i_1} and e_{i_2} will be reached through graph-search algorithms, e.g., breath-first search (BFS), and served as candidates. We set a path truncation threshold M for filtering weakly related items to avoid noise. Second, due to the varying size of candidate pools for different items, we sample an itemset $\mathcal{I}_i = \{e_{i,1}, e_{i,2}, ..., e_{i,H}\}$ with a fixed size of H to guarantee the learning process to be efficient and scalable. For simplicity, we use the uniform sampling strategy. We will leave the design of itemset selection strategy in future work.

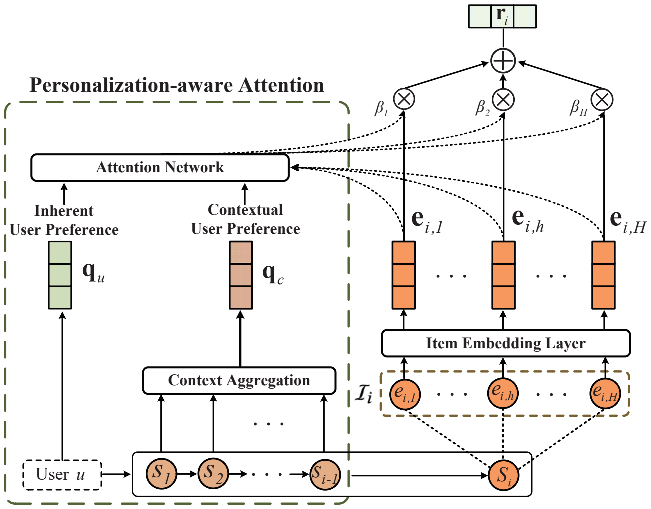


Figure 3: Illustration of the personalization-aware attentive aggregation unit, which adaptively captures itemset-level interest considering two types of personalization signals.

4.2.2 Personalization-aware Attentive Aggregation. As shown in Figure 3, we design an attentive aggregation module to learn personalized itemset-level interest representation for each interaction. Instead of simply adding or concatenating item embeddings, we perform personalized attention considering both inherent and contextual user preferences as two types of personalization signals to combine item embeddings of an itemset.

On the one hand, different users could have their own inherent preferences for items in the same itemset. We leverage the learnable embedding of user identity as the representation of inherent user preference $\mathbf{q}_u = \mathbf{u} \in \mathbb{R}^d$, which is utilized to provide personalization signals for different users. It's also feasible to enrich the inherent user preference representation with user profile information, e.g., *age*, *gender* and *occupation*.

On the other hand, even for the same user, his/her preferences for the same itemset are usually dynamic in different contexts. Users' preferences are usually affected by their recent or historical interactions. Therefore, for an interaction s_i, we extract contextual user preference representation $\mathbf{q}_c \in \mathbb{R}^d$ from previous interactions $\{s_1, ..., s_{i-1}\}$ to provide personalization signals of different contexts. We implement three ways of context aggregation, including: 1) short-term context, where $\mathbf{q}_c = \mathbf{e}_{i-1}$. The last interacted item is treated as the recent or short-term context. 2) long-term context, where $\mathbf{q}_c = avg(\mathbf{e}_1, .., \mathbf{e}_{i-1})$. The average pooling of embeddings of all previously interacted items are utilized to indicate long-term context. 3) attentive context, which is obtained as follows:

$$\mathbf{q}_c = \sum_{j=1,\dots,i-1} \alpha_j \mathbf{e}_j, \tag{4}$$

$$\alpha_j = \text{softmax}(\mathbf{v}_c \sigma(\mathbf{W}_{c_1}\mathbf{e}_j + \mathbf{W}_{c_2}\mathbf{e}_{i-1})), \tag{5}$$

where $\mathbf{W}_{c_1}, \mathbf{W}_{c_2} \in \mathbb{R}^{d\times d}$ and $\mathbf{v}_c \in \mathbb{R}^d$. An attention mechanism is employed to adaptively capture the main purpose of previous interactions with the most recent interaction as query signals. We discuss the effect of different personalization signals in section 5.3.

After obtaining two types of user preference representations as personalized queries, an attention network is utilized to compute the attention weights of different items in an itemset. The attention weight of the h-th item in the itemset $\mathcal{I}_i$ is calculated as follows:

$$\beta_h = \text{softmax}\left(\mathbf{v}_h \sigma(\mathbf{W}_{h_1}\mathbf{q}_u + \mathbf{W}_{h_2}\mathbf{q}_c + \mathbf{W}_{h_3}\mathbf{e}_{i,h})\right), \tag{6}$$

where $\mathbf{W}_{h_1}, \mathbf{W}_{h_2}, \mathbf{W}_{h_3} \in \mathbb{R}^{d\times d}$ and $\mathbf{v}_h \in \mathbb{R}^d$ are parameters to be learned. The final itemset-level interest representation of the i-th interaction is the summation of weighted item embeddings:

$$\mathbf{r}_i = \sum_{h=1}^{H} \beta_h \mathbf{e}_{i,h}. \tag{7}$$

4.3 Sequential Interactions Modeling

Considering the chronological order of users' historical interactions, the evolution of user preferences should be well captured from the interaction sequence. To realize this, we focus on how to effectively fuse two levels of interest modeling to infer a comprehensive preference representation of each interaction and how to capture the evolution of user preferences from the whole sequence.

4.3.1 Gated Interest Fusion. To incorporate both item-level and itemset-level interest representations of each interaction, we first design an interest fusion module to generate high-level preference representations. Considering the unavailability of users' real intention of interactions, we balance the contributions of two levels of interest modeling with a learnable gating fusion unit.

Specifically, inspired by the design of GRU [4] that learns gating signals to control the update of hidden states, we propose to learn a fusion gate to adaptively control the combination of two levels of interest representations. For the i-th interaction, given knowledge enhanced item-level interest representation, i.e., $\mathbf{c}_i$ and itemset-level interest representation, i.e., $\mathbf{r}_i$, we combine them as follows:

$$\mathbf{g}_i = \sigma(\mathbf{W}_{g_1}\mathbf{c}_i + \mathbf{W}_{g_2}\mathbf{r}_i), \tag{8}$$

$$\mathbf{p}_i = \mathbf{g}_i \cdot \mathbf{c}_i + (1 - \mathbf{g}_i) \cdot \mathbf{r}_i, \tag{9}$$

where $\mathbf{W}_{g_1}$ and $\mathbf{W}_{g_2} \in \mathbb{R}^{d\times d}$ are learnable transformation parameters and σ is the sigmoid function. The $\mathbf{g_i} \in \mathbb{R}^d$ denotes the learned gate signals to balance the contributions of the item-level and itemset-level interest modeling. The low values of $\mathbf{g_i}$ indicate that the currently interacted item may be insufficient for reflecting user preferences, i.e., interest modeling expansion towards the relevant itemset should be taken into account. In the end, we finally obtain the sequence of refined preference representations for all historical interactions, which is denoted as $P^u = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$.

4.3.2 Multi-head Self-attention Layers. Next, we focus on capturing the sequential evolution of user preferences via state-of-the-art self-attention networks [31]. Compared with other self-attention network based methods which mainly capture sequential dependencies with item sequences [16, 28], we apply the self-attention layers on the sequence of high-level preference representations P^u.

The preference representations are combined as a matrix $\mathbf{P} \in \mathbb{R}^{n\times d}$ along with the interaction order. To indicate relative positions, we assign learnable positional embeddings $\mathbf{Z} \in \mathbb{R}^{n\times d}$ and inject them into preference representations as $\tilde{\mathbf{P}} = \mathbf{P} + \mathbf{Z}$. We build stacked self-attention layers, which are mainly composed of multi-head self-attention blocks and point-wise feed-forward networks. Single multi-head self-attention block adopts h parallel modules and each module performs scaled dot-product attention over $\tilde{\mathbf{P}}$ as follows:

$$\mathbf{H_i} = \mathrm{softmax}\left(\frac{(\tilde{\mathbf{P}}\mathbf{W}_i^Q)(\tilde{\mathbf{P}}\mathbf{W}_i^K)^\top}{\sqrt{d}}\right)(\tilde{\mathbf{P}}\mathbf{W}_i^V), \tag{10}$$

where $\mathbf{W}_i^Q$,$\mathbf{W}_i^K$,and $\mathbf{W}_i^E \in \mathbb{R}^{d\times\frac{d}{h}}$ are projection matrics for generating queries, keys and values, respectively. The attention weight calculated with j-th query and k-th key represents the dependency strength between user preferences at the j-th and k-th interaction. Considering the temporal order of interactions, we mask the weights of subsequent preferences for each interaction to guarantee causality. Thus we can obtain an attentionally weighted sum of previous preference representations to capture sequential evolution.

After concatenating the outputs of parallel self-attention modules, the outputs $\mathbf{O} \in \mathbb{R}^{d\times d}$ are fed into feed-forward networks (FFNs) to endow the nonlinear transformation:

$$\mathbf{O} = \mathrm{concat}(\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_h), \tag{11}$$

$$\mathbf{F} = \mathrm{ReLU}(\mathbf{O}\mathbf{W}_{o1} + \mathbf{b}_1)\mathbf{W}_{o2} + \mathbf{b}_2 + \mathbf{O}, \tag{12}$$

where the residual connection is applied for low-layer feature aggregation. We unite the calculations of a single self-attention layer in equation (10-12) as $\mathbf{F} = \mathrm{SA}(\tilde{\mathbf{P}})$. To learn more complex dependencies among preferences, we stacked multiple self-attention layers to hierarchically extract information from the sequence of preference representations. The outputs of i-th layer is calculated as:

$$\mathbf{F}^{(i)} = \mathrm{SA_i}(\mathbf{F}^{(i-1)}), i = 1, ..., S, \tag{13}$$

where S is the number of self-attention layers and $\mathbf{F}^{(0)} = \tilde{\mathbf{P}}$.

4.4 Prediction and Training

To predict the next item that a user will interact with, we generate final representations for the user and candidate items and calculate the interaction probabilities of user-item pairs. We first obtain $\mathbf{F}_n^{(S)} \in \mathbb{R}^d$, i.e., the last vector of outputs of the self-attention layers, which captures the global preference evolution from historical interactions. Then we obtain the user representation as follows:

$$\mathbf{f}_u = \tanh(\mathbf{W}_u[\mathbf{F}_n^{(S)}; \mathbf{u}] + \mathbf{b}_u), \tag{14}$$

where $\mathbf{W}_u \in \mathbb{R}^{d\times 2d}$ and $\mathbf{b}_u \in \mathbb{R}^d$. We compute the interaction probability between the user u and a candidate item v as follows:

$$\hat{y} = \sigma(\mathbf{f}_u^\top \mathbf{c}_v), \tag{15}$$

where σ is the sigmoid function to scale $\hat{y} \in [0, 1]$ and $\mathbf{c}_v$ is the knowledge enhanced representation of a candidate item v.

To optimize embeddings and model parameters, we adopt the cross-entropy loss as the objective function, which is defined as:

$$\mathcal{L} = -\sum_{u_i\in\mathcal{U}}\sum_{t=2}^{n}(\log \hat{y}_{i,s_t} + \sum_{j\in\mathcal{V}_{i,t}^-} log(1 - \hat{y}_{i,j})) + \lambda\mathcal{L}_{reg}, \tag{16}$$

where $\hat{y}_{i,j}$ denotes the interaction probability between user u_i and item v_j. The $\mathcal{V}_{i,t}^-$ is the set of sampled negative items for the t-th interaction of user u_i. The ground truth of interaction probability is set to 1 for interacted items and 0 for negative instances. $\mathcal{L}_{reg}$ is the regularization term on all learnable parameters and embeddings to avoid overfitting. We utilize the Adam [17] to perform optimization by minimizing the loss of mini-batch samples.

Table 1: Statistics of the datasets

Datasets	MovieLens	LastFM	Amazon-Book
#Users	6,036	47,183	66,898
#Items	2,445	55,431	22,072
#Interactions	547,018	2,540,574	798,765
Avg. seq. len.	88.63	51.85	9.94
#Entities	182,011	108,930	79,682
#Relations	12	45	38
#Triples	1,241,995	914,842	400,787

5 EXPERIMENTS

In this section, we evaluate our proposed method on real-world recommendation datasets. We first introduce the setup of the experiments including datasets, evaluation metrics, and comparison baselines. Then we discuss the experimental results in detail.

5.1 Experimental Setup

5.1.1 Datasets. In this work, we study the effectiveness of our model over three recommendation datasets from real-world applications, including **MovieLens** [6], **LastFM** [27] and **Amazon-book** [12]. The MovieLens dataset[1] contains 1M rating records for movies from the *MovieLens* website. The LastFM dataset collects listening interactions of music in 2012 from the whole LFM-1b dataset[2]. The Amazon book dataset[3] contains user rating scores for book products in the *Amazon* E-commerce platform. For the knowledge graph data, we adopt the KB4Rec dataset [46], which consists of subgraphs from a large knowledge base namely Freebase. The subgraphs contain knowledge triples that are extracted via breath-first search starting from the aligned item entities.

We follow the same preprocessing procedure in previous studies [7, 12]. For each dataset, we first discard items without aligned entities in KG and then filter unpopular items and inactive users involved in less than k interactions, where k is 10 for the LastFM dataset and 5 for others. We summarize the statistics of three preprocessed datasets as shown in Table 1. For dataset partitioning, we hold out the last item in the interaction sequence of each user as test data and the second last item as validation data. The remaining interactions are treated as train data. For each interaction, previously interacted items constitute the historical interaction sequence.

5.1.2 Evaluation Metrics. Following the previous settings [12, 16], we adopt two widely used evaluation metrics in terms of top-K recommendation, i.e., Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). Higher HR and NDCG indicate better performance. To alleviate the heavy computation with all items as the candidates, we randomly sample 100 negative items for each ground-truth item following [16, 28] and rank them according to the

[1]https://grouplens.org/datasets/movielens/1m/
[2]http://www.cp.jku.at/datasets/LFM-1b/
[3]http://jmcauley.ucsd.edu/data/amazon/

Table 2: Recommendation performance comparison of different approaches. Bold scores are the best results for each metric, while the second-best scores are underlined. %Improv. indicates the relative improvements of our model over the best baselines.

Method	MovieLens			LastFM			Amazon-Book		
	HR@5	HR@10	NDCG@10	HR@5	HR@10	NDCG@10	HR@5	HR@10	NDCG@10
BPR-MF	0.5199	0.6978	0.4211	0.4714	0.6040	0.3868	0.5332	0.6517	0.4443
NCF	0.4843	0.6627	0.3864	0.5605	0.6804	0.4988	0.4826	0.5911	0.3978
GRU4Rec	0.5744	0.7334	0.4645	0.4378	0.5608	0.3586	0.4972	0.6278	0.4085
SASRec	<u>0.6652</u>	<u>0.7961</u>	<u>0.5287</u>	0.5749	0.7021	0.4816	0.5731	0.6972	0.4677
BERT4Rec	0.4867	0.6624	0.3808	0.5532	0.6987	0.4542	0.5473	0.6748	0.4498
KGCN	0.4430	0.6332	0.3561	0.5470	0.6950	0.4549	0.5100	0.6469	0.4331
KGAT	0.5597	0.6913	0.4691	0.6380	0.7747	0.5266	0.5553	0.6850	0.4599
CKAN	0.6092	0.7255	0.5023	<u>0.6743</u>	<u>0.7993</u>	<u>0.5988</u>	<u>0.5762</u>	<u>0.7045</u>	<u>0.4938</u>
KSR	0.6123	0.7243	0.4901	0.5761	0.6881	0.4888	0.5667	0.6873	0.4671
FDSA	0.6473	0.7740	0.5248	0.6326	0.7243	0.5634	0.5552	0.6648	0.4718
KGIE	**0.7160**	**0.8240**	**0.5833**	**0.7675**	**0.8532**	**0.6681**	**0.6720**	**0.7797**	**0.5662**
%Improv.	7.64%	3.50%	10.33%	13.82%	6.74%	11.57%	16.62%	10.67%	14.66%

calculated interaction probabilities. We evaluate the performance of all methods over the same metrics and test data.

5.1.3 Baselines. We compare our model KGIE with the following ten representative and state-of-the-art approaches.

- **BPR-MF** [25] is a classical collaborative filtering (CF) method, which optimizes matrix factorization using BPR loss.
- **NMF** [9] is a state-of-the-art neural network based CF method which combine the linearity of MF and non-linearity of MLP.
- **GRU4Rec** [10] utilizes the GRU structure to handle interaction sequences for sequential recommendation.
- **SASRec** [16] employs the self-attention networks to capture pairwise dependencies among historical interactions.
- **BERT4Rec** [28] employs a bidirectional self-attention architecture to encode historical interaction sequences.
- **KGCN** [35] extends non-spectral GCN approaches to knowledge graph, which aggregates semantic information from multi-hop neighborhood for candidate items.
- **KGAT** [37] utilizes attentive embedding propagation in the collaborative KG which unifies both user-item interaction data and knowledge graph data.
- **CKAN** [39] designs explicit collaboration propagation to enhance knowledge graph based recommendation.
- **KSR** [12] integrates knowledge enhanced memory networks with a GRU-based sequential recommender
- **FDSA** [45] leverages parallel self-attention blocks to encode item-level and feature-level sequences. We utilize pretrained knowledge graph embeddings as feature embeddings.

Table 3: The categorization of comparison methods.

Squential order	Knowledge graph: without	Knowledge graph: with
without	BPR-MF, NMF	KGCN, KGAT, CKAN
with	GRU4Rec, SASRec, BERT4Rec	KSR, FDSA

To summarize, We group the comparison methods above according to whether the sequence order of interactions is considered and whether the knowledge graph is used, as shown in Table 3.

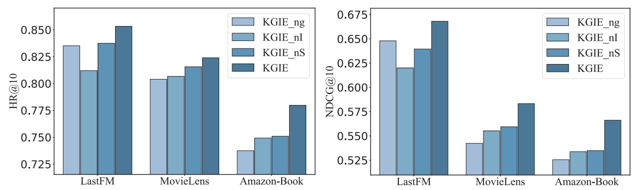
5.1.4 Implementation Settings. For SASRec, BERT4Rec, KGCN, KGAT, CKAN, and KSR, we use the source codes released by the authors. For the remaining baselines, we implemented them ourselves. We perform a grid search for setting key hyper-parameters of these baselines and configure test models with the optimal settings obtained over evaluation data.

In our model, we tune embedding dimension d in $\{16, 32, 64, 128, 256\}$ and set 64 as default to tradeoff performance and costs. For the max sequence length n, we set 100, 50, and 20 for MovieLens, LastFM, and Amazon-Book dataset according to the average length of interaction sequences in different datasets. After tuning the key hyperparameters on the evaluation set, the number of stacked knowledge graph convolution layers K is 2 for the MovieLens dataset and 1 for others. The fixed size of itemset H is set as 8 in default. We also analyze the influence of both K and H in section 5.4. In addition, we sample $L=8$ neighbors for each entity in the knowledge graph convolution layers. For searching for itemsets, we use the breath-first search algorithm and set path truncation threshold M as 4 for MovieLens and 6 for the other two datasets. Following [16, 28], the number of self-attention layers, i.e., S is 2 and each layer contains $h = 2$ heads. For the training process, the learning rate and batch size are set as 10^{-3} and 128. We use L2 regularization with a weight of $\lambda = 10^{-4}$.

5.2 Performance Comparison

Table 2 shows the recommendation performance of all methods over three datasets in terms of HR@5, HR@10, and NDCG@10. From the results, we have the following observations.

Compared with non-sequential recommendation methods including BPR-MF and NMF, sequential-based methods achieve better performance in general. It suggests that capturing sequential patterns is helpful for depicting dynamic user preferences. We also observe that GRU4Rec obtains poor results over two sparse datasets, i.e., LastFM and Amazon-Book datasets, while SASRec and BERT4Rec



(a) Hit Ratio@10 (b) NDCG@10

Figure 4: Performance comparison of different variants.

Table 4: Effect of different personalization signals in itemset-level interest aggregation unit in top-10 recommendation.

Variants	MovieLens		LastFM		Amazon-Book	
	HR	**NDCG**	**HR**	**NDCG**	**HR**	**NDCG**
w/o user	0.8027	0.5683	0.8490	0.6614	0.7690	0.5562
w/o cont.	0.7878	0.5244	0.8342	0.6472	0.7533	0.5382
st-cont.	**0.8240**	**0.5833**	0.8472	0.6567	0.7675	0.5524
lt-cont.	0.8153	0.5629	0.8443	0.6528	0.7695	0.5534
att-con.	0.8168	0.5658	**0.8532**	**0.6681**	**0.7797**	**0.5662**

perform better. The reason is that the self-attention networks are more effective to adaptively handle datasets with various densities.

By incorporating item knowledge, most of the KG-based methods obtain superior results than collaborative filtering methods (BPR-MF and NMF), especially over LastFM and Amazon-Book dataset. It suggests that leveraging extra knowledge information is essential for alleviating the data sparsity problem. Among these methods, KGAT and CKAN explicitly integrate user-item interactions with knowledge graphs and achieve considerable performance.

For enhancing the ability of sequential recommendation, both KSR and FDSA extract attribute information of interacted items from the knowledge graph. By comparison, FSDA achieves better performances on most evaluation metrics. The possible reason is that KSR updates the external attribute-level memories in a simple recurrent manner, while FSDA utilizes more effective self-attention networks to encode the whole attribute sequence.

Finally, our approach KGIE consistently achieves the best performances over all datasets compared with state-of-the-art methods. KGIE obtains significant improvements over the strongest baselines on three datasets, respectively (e.g., 10.33%, 11.57%, and 14.66% in terms of NDCG@10). In particular, our model significantly outperforms FDSA and KSR, which suggests that our model could better utilize knowledge graphs for enhancing sequential recommendation with effective itemset-level interest modeling expansion.

5.3 Component Analysis

To investigate the effectiveness of the design of our proposed model, we first analyze the impact of several major modules via ablation study and then explore the influence of various personalization signals for aggregating itemset-level user interests.

5.3.1 Ablation Study. To verify the effectiveness of the main components of our model, we conduct an ablation study as shown in Figure 4. We compare the performance of KGIE and its variant by removing or replacing some key components. Three variants are obtained as follows, including (1) KGIE-nS, which removes the itemset-level interest modeling module. (2) KGIE-nI, which removes the knowledge graph convolution layers for incorporating item knowledge at the item level. (3) KGIE-ng, which replaces the gating mechanism with simple additional aggregation.

From the results on Hit Ratio@10 and NDCG@10 over all datasets shown in Figure 4, we have the following observations. First, the KGIE-ng achieves the worst performance over MovieLens and Amazon-Book datasets, which indicates the necessity of the gating fusion unit for adaptively combining both item-level and itemset-level interest modeling. Second, without explicitly learning itemset-level interests in KGIE_nS, consistent performance degradation is observed over all datasets. This indicates that semantically related itemsets can reveal potential user interests to some extent. Finally, KGIE_nI obtains low recommendation performance, especially over the LastFM dataset where other KG-based methods achieve considerable performances. It shows that rich attribute information is an important factor to enrich item representations.

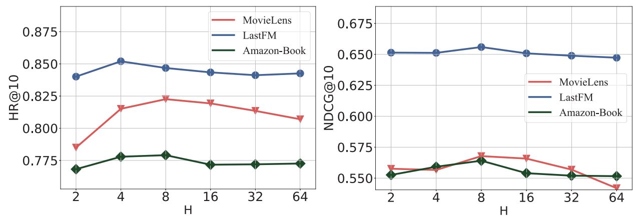
5.3.2 Effect of different personalization signals. To observe the impact of various personalization signals when refining itemset-level interest, we construct several variants by configuring different signals in the attentive aggregation unit. Specifically, we first disable inherent and contextual user preferences respectively (abbreviated as w/o user and w/o cont.). In addition, we vary three types of contextual user preferences, including short-term, long-term, and attentive context (abbreviated as st-cont. and lt-cont. and att-cont.).

By analyzing the results presented in Table 4, we have the following findings. First, removing inherent user embeddings causes performance degradation of the model. It suggests that considering inherent user preferences could be beneficial for capturing personalized interests for different users, which attributes to the performance improvements. Second, the variant without contextual user preferences achieves the lowest results in terms of all metrics. The results confirm our intuition that user interests are variable in different contexts. Third, for different datasets, best performances are obtained with different types of context aggregation. In specific, the variant with short-term context performs best over the Movielens dataset, and the variant with attentive context performs better over other two datasets. One possible reason is that the short-term dependencies among interactions are more obvious in the Movielens dataset, while richer context information is more in line with the characteristics of the others.

5.4 Influence of Hyper-parameters

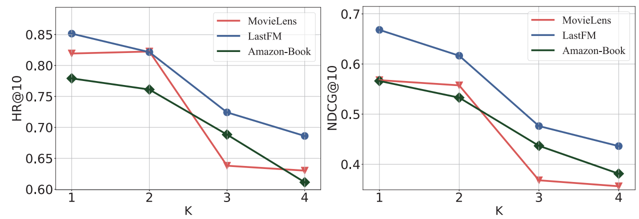
To study the influence of different settings on key hyper-parameters, we evaluate our model by varying the size of relevant itemsets H and the step of embedding aggregation K, respectively.

First, we vary the sampled number of relevant items for each interaction from 2 to 64 as shown in Figure 5. We can see that higher performance is obtained when H is 4 on the MovieLens dataset and 8 for others. When the itemset size is relatively small, increasing the size of itemsets is capable of improving performance significantly. The reason for performance degradation of further



(a) Hit Ratio@10 (b) NDCG@10

Figure 5: Effect of the itemset size H .



(a) Hit Ratio@10 (b) NDCG@10

Figure 6: Effect of the embedding aggregation step K.

increasing itemset size could be that taking relatively irrelevant items into account might bring noises.

Next, we vary the step of embedding aggregation in the knowledge graph convolution layers from 1 to 4. The results shown in Figure 6 demonstrate that the performance of our model is quite sensitive to K, which is consistent with the conclusion in [35]. In specific, the best performance is obtained when K is 1 or 2 in most cases. Huge performance degradations are observed when increasing K to 3 or higher. It shows that simply incorporating high-order knowledge information via multi-hop graph convolution is not effective to utilize abundant semantic information.

5.5 Recommendation Interpretability

Besides accurate recommendation results, another benefit of the design of our model is that KGIE can provide better interpretability of recommendations. To demonstrate this, we conduct a case study in real-world datasets as shown in Figure 7. In specific, we randomly sample a user from the MovieLens dataset and recommend top-3 items based on his/her previous interaction sequence with KGIE. For each interaction, we calculate the average of gating signals to indicate contributions of item-level interests. Furthermore, we present personalized attention scores for items in knowledge enriched itemsets. For conciseness, we show the latest three interactions and two items with higher attentions in each itemset.

From the example, we could observe that our model provides interpretability of the recommendations to users in the following aspects. On the one hand, in all itemsets, movies with same genre *Drama* and *Thriller* as the ground-truth next movie *Green miles* are assigned with high attention weights, e.g., *Raise the Titanic* (0.875) and *Normal life* (0.802). In other words, these items are automatically highlighted in our model as possible explanations of the recommendation results. On the other hand, one interacted animation movie *Toy Story* 2 which is inconsistent with user preferences on the genre, e.g., *Thriller*, is assigned with low gating values. Assigned with higher gating values, the interactions on two thriller movies are indicated as having great contributions for final recommendations. It suggests that the learnable gating fusion unit could infer users' real preferences to some extent.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel knowledge graph based sequential recommendation model namely KGIE. We enhance the ability to model both item-level and itemset-level user interests based

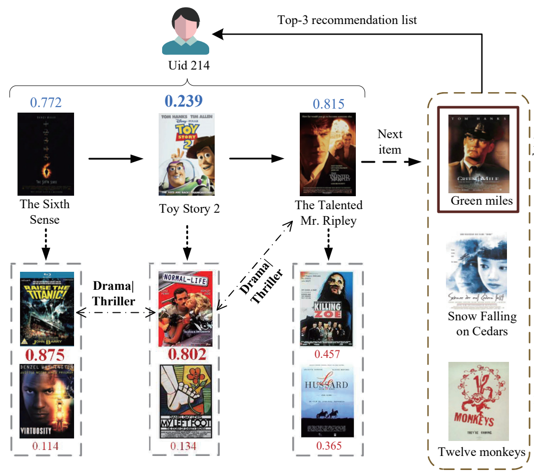


Figure 7: A real example from a sampled user in the MovieLen dataset. We present attention scores of items in relevant itemsets (marked in red) and the average of gating values for item-level interest (marked in blue).

on the knowledge graph. Personalized itemset-level interest representations are learned from knowledge-enriched itemsets with a personalization-aware attentive aggregation unit. Extensive experiments on three real-world datasets demonstrate the effectiveness of our proposed model. For the future work, we will extend the concept of knowledge-enriched itemsets towards other types of itemsets by utilizing other auxiliary information. Meanwhile, we will also explore more effective itemset sampling strategies, e.g., taking the length of connections or item popularity into account.

7 ACKNOWLEDGEMENTS

This research is supported in part by the 2030 National Key AI Program of China 2018AAA0100503 (2018AAA0100500), National Science Foundation of China (No. 62072304, No. 61772341, No. 61472254, No. 61770238), Shanghai Municipal Science and Technology Commission (No. 18511103002, No. 19510760500, and No. 19511101500), the Program for Changjiang Young Scholars in University of China, the Program for China Top Young Talents, the Program for Shanghai Top Young Talents, SJTU Global Strategic Partnership Fund (2019 SJTU-HKUST), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University and Scientific Research Fund of Second Institute of Oceanography (No. SL2020MS032).

REFERENCES

[1] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *WSDM*. 46–54.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 1–9.

[3] Sung Min Cho, Eunhyeok Park, and Sungjoo Yoo. 2020. MEANTIME: Mixture of Attention Mechanisms with Multi-temporal Embeddings for Sequential Recommendation. In *RecSys*. 515–520.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[5] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *arXiv preprint arXiv:2003.00911* (2020).

[6] F Maxwell Harper and Joseph A Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2016), 19.

[7] Gaole He, Junyi Li, Wayne Xin Zhao, Peiju Liu, and Ji-Rong Wen. 2020. Mining Implicit Entity Preference from User-Item Interaction Data for Knowledge Graph Completion via Adversarial Learning. In *WWW*. 740–751.

[8] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM*. IEEE, 191–200.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and D Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.

[11] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *KDD*. 1531–1540.

[12] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*. 505–514.

[13] 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 *ACMMM*. 548–556.

[14] Xiaowen Huang, Shengsheng Qian, Quan Fang, Jitao Sang, and Changsheng Xu. 2018. Csan: Contextual self-attention network for user sequential recommendation. In *ACMMM*. 447–455.

[15] Xuanwen Huang, Yang Yang, Ziqiang Cheng, Shen Fan, Zhongyao Wang, Juren Li, Jun Zhang, and Jingmin Chen. 2021. How Powerful are Interest Diffusion on Purchasing Prediction: A Case Study of Taocode. In *SIGIR*. 1308–1317.

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

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *CIKM*. 1419–1428.

[19] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *WSDM*. 322–330.

[20] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: fusing item similarity models with self-attention networks for sequential recommendation. In *RecSys*. 130–139.

[21] Haobing Liu, Yanmin Zhu, Tianzi Zang, Jiadi Yu, and Haibin Cai. 2020. Jointly Modeling Individual Student Behaviors and Social Influence for Prediction Tasks. In *CIKM*. 865–874.

[22] Jianhua Liu, Jiadi Yu, and Shigen Shen. 2017. Energy-efficient two-layer cooperative defense scheme to secure sensor-clouds. *IEEE Transactions on Information Forensics and Security* 13, 2 (2017), 408–420.

[23] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–36.

[24] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys*. 130–137.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI Press, 452–461.

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*. 811–820.

[27] Markus Schedl. 2016. The lfm-1b dataset for music retrieval and recommendation. In *ICMR*. 103–110.

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

[29] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *RecSys*. 297–305.

[30] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *RecSys, Workshop*. 17–22.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[32] Chunyang Wang, Yanmin Zhu, Tianzi Zang, Haobing Liu, and Jiadi Yu. 2021. Modeling Inter-station Relationships with Attentive Temporal Graph Convolutional Network for Air Quality Prediction. In *WSDM*. 616–634.

[33] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*. 417–426.

[34] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *KDD*. 968–977.

[35] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *WWW*. 3307–3313.

[36] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *SIGIR*. 1101–1110.

[37] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[38] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, Vol. 33. 5329–5336.

[39] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In *SIGIR*. 219–228.

[40] Zhichen Wu, Jianda Li, Jiadi Yu, Yanmin Zhu, Guangtao Xue, and Minglu Li. 2016. L3: Sensing driving conditions for vehicle lane-level localization on highways. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[41] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*. 3940–3946.

[42] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *SIGIR*. 729–732.

[43] Tianzi Zang, Yanmin Zhu, Yanan Xu, and Jiadi Yu. 2021. Jointly Modeling Spatio–Temporal Dependencies and Daily Flow Correlations for Crowd Flow Prediction. *TKDD* 15, 4 (2021), 1–20.

[44] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *KDD*. 353–362.

[45] 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 *IJCAI*. 4320–4326.

[46] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. KB4Rec: A Data Set for Linking Knowledge Bases with Recommender Systems. *Data Intelligence* 1, 2 (2019), 121–136.

[47] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2020. A Knowledge-Aware Attentional Reasoning Network for Recommendation.. In *AAAI*. 6999–7006.