



McHa: a multistage clustering-based hierarchical attention model for knowledge graph-aware recommendation

Jihu Wang¹ · Yuliang Shi^{1,2} · Dong Li³ · Kun Zhang⁴ · Zhiyong Chen¹ · Hui Li¹

Received: 1 July 2021 / Revised: 7 December 2021 / Accepted: 2 February 2022 /

Published online: 28 February 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Knowledge graph-aware recommendation has become an important research topic in recent years. The user preference representation, which preserves the user's taste towards items (e.g., movies, books.), is obtained through aggregating the information of entities or attributes in knowledge graphs directly. However, the fine-grained heterogeneity information, which can be derived from the groups of items or entities, remains barely exploited in the process of encoding the user interaction intention for the items. To fill up this gap, we propose a *Multistage Clustering-based Hierarchical Attention (McHa)* model to capture the user preference representation. In our work, we first group the items and their neighboring entities in the knowledge graph into item clusters and entity clusters (jointly referred to as multistage clusters), respectively. Then, the user preference representation is obtained by hierarchically aggregating the heterogeneity information derived from the multistage clusters with the weights generated by the hierarchical attention layers. We conduct extensive experimental comparisons with baselines and the variants. The experimental results indicate that **McHa** has achieved state-of-the-art performance on three benchmark datasets in two scenarios.

Keywords Recommender systems · Knowledge graph · Attention mechanism · Multistage clustering

1 Introduction

Prior works [20, 45, 47] have shown that introducing knowledge graphs (KGs) into recommender systems (RS) can effectively improve the accuracy of recommendation and solve the problems of data sparsity and cold start, compared with the traditional recommendation methods, such as content-based methods [30] and collaborative filtering (CF)-based methods [25]. Besides, KGs have been successively used in many intelligent tasks due to their rich side information, such as question answering and information retrieval.

✉ Yuliang Shi
shiyuliang@sdu.edu.cn

Extended author information available on the last page of the article

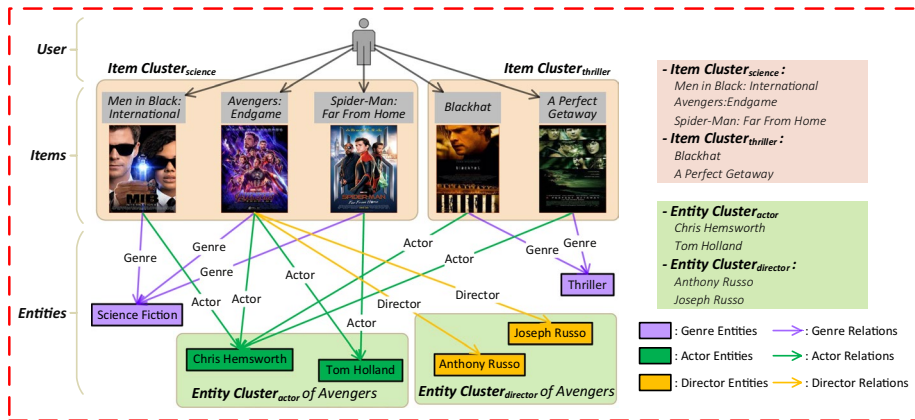


Fig. 1 A toy example of the movie-related User-Item-Entity Knowledge Graph (UIEKG). The entities or attributes are extracted from Microsoft Satori

KG is a kind of semantic network composed of entities and relations. Its basic unit is a triple (h, r, t) , where h and t represent the head entity and the tail entity, respectively, and r represents the relation between h and t . For example, the two triples, $\langle Donald Trump, occupation, Politician \rangle$ and $\langle Donald Trump, occupation, Investor \rangle$, mean that *Donald Trump* is not only a politician but also an investor. The main idea of the existing KG-aware recommendation methods [51, 55, 56] is to build a *User-Item-Entity* KG (UIEKG) by connecting the user’s interacted items with the entities or attributes extracted from the side KG (such as Wikidata [49], DBpedia [2], Yago [44] and Satori [37].), and then obtain the user preference representation through propagating the information of entities or attributes extracted from the UIEKG. For example, KGAT [51] updates the user preference vector through aggregating the embeddings of the neighbors, and recursively performs such updating process to capture the high-order information of neighbors. Meanwhile, the attention mechanism is used for weighting the importance of neighbors. However, directly aggregating or propagating the information of entities instead of further processing them leaves the useful information barely exploited, such as the heterogeneity information [4], which can be derived from different item clusters or entity clusters.

Exploring the focused heterogeneity could enhance the ability of the recommendation model to accurately capture the user’s fine-grained tastes. For example, as shown in Figure 1, given a user’s viewed record, a movie-related UIEKG can be built by connecting the movies to the actors, directors, and genres extracted from Satori. We group the items (movies) into *Item Cluster_{science}* and *Item Cluster_{thriller}* according to their genres, which gives these two item clusters the item-level heterogeneity in terms of movie genre. If *Item Cluster_{thriller}* is paid more attention than *Item Cluster_{science}* when encoding the different item-level heterogeneity into the user preference, which probably means that the user prefers thriller to science fiction. Similarly, we could group the entities headed with *Avengers: Endgame* into *Entity Cluster_{actor}* and *Entity Cluster_{director}* according to their relations, which gives the two entity clusters the entity-level heterogeneity in terms of the relation between head entity and tail entity. If *Entity Cluster_{actor}* is paid more attention than *Entity Cluster_{director}* when encoding the different entity-level heterogeneity into the user preference, which probably means that the user decided to watch *Avengers: Endgame* largely depending on who acted in this movie rather than who

directed it. Through the case analysis, we assume that encoding the multi-level (item-level and entity-level) heterogeneity derived by multistage clustering could enhance the pertinence of user preference.

Based on the above assumption, we propose a *Multistage Clustering-based Hierarchical Attention (McHa)* model to obtain the user preference representation of knowledge graph-aware recommendation. Specifically, we first group the items and their neighboring entities in UIEKG into item clusters and entity clusters (jointly referred to as multistage clusters) according to the attributes of items (e.g., genres of movies or authors of books) and the relations between head entities and tail entities, respectively. Then, we construct the hierarchical attention layers to discriminatively aggregate the multi-level heterogeneity information derived from the multistage clusters into the user preference. Intuitively, our model can produce more focused user preference representation based on the following distinctive designs: 1) multistage clustering could produce the multi-level heterogeneity information of the items and their neighboring entities in UIEKG for encoding the user interaction intention for the characteristic items; 2) the hierarchical attention layers built by integrating attention mechanisms [61] with graph attention networks (GAT) [48] could discriminate the importance of each cluster and its elements; besides, 3) we explicitly encode the relation embedding into the entity cluster representation to enhance the heterogeneity of different entity clusters in terms of triple's relation. Our contributions can be summarized as follows:

- We propose a novel knowledge graph-aware recommendation model, namely **McHa**, to obtain the fine-grained user preference representation strengthened with the multi-level heterogeneity derived by grouping the items and their neighboring entities into multistage clusters.
- We construct the hierarchical attention layers by integrating multi-level attention mechanisms with GAT to discriminate the contribution of each cluster to the user preference representation.
- We demonstrate the effectiveness of our model and the positive effect of each part in **McHa** through the comparative experiments with the state-of-the-art baselines and the ablation analysis with its variants, respectively, on three benchmark datasets in two scenarios.

The remainder of this paper is organized as follows. In Section 2, we survey the related works on KG-aware recommendation as well as recent emerging topics of recommender systems. In Section 3, we present our model in detail. In Section 4, we show our experiments and analyze the results. Finally, we conclude our work and look forward to the work of this paper in Section 5.

2 Related work

On the one hand, we first survey the literature related to KG-aware recommendation in this section. Following [16], we divide KG-aware recommendation methods into three categories: embedding-based, path-based, and unified methods. On the other hand, the recent emerging research topics of recommender systems have also been discussed.

2.1 Knowledge graph aware recommendation

2.1.1 Embedding-based methods

The embedding-based methods [1, 19, 36, 54] generally embed the semantic information of KG into the representations of the items or users. For example, CKE [64] leverages heterogeneous network embedding and deep learning embedding approaches, to automatically extract semantic representations from multi-modal knowledge. Then, it combines collaborative filtering and knowledge embedding components into a unified framework and learns different representations jointly. MKR [50] builds several cross and compress units, which automatically share latent features and learn high-order interactions between items in recommender systems and entities in KGs. However, the connectivity in KGs is ignored in embedding-based methods, which makes it difficult to explain the recommendation results.

2.1.2 Path-based methods

Path-based methods mainly enhance the ability of recommendation model through exploring the connectivity in KGs [8, 21, 58]. For instance, HeteRec [62] uses the meta-path-based latent features to represent the connectivity between users and items along different paths. Then, a recommendation model with such latent features is defined and optimized through bayesian ranking optimization techniques. Later, FMG [65] improves the accuracy of recommendation by replacing the meta-path with the meta-graph. Moreover, to discriminate the importance of different paths, MCRec [18] is proposed to learn representations for users, items, and the meta-paths extracted through priority-based sampling. Then, the co-attention mechanism is applied to strike a balance between the meta-paths and user-item pairs to mutually improve their representations. RuleRec [33] induces rules from KGs for items and then makes recommendations based on the induced rules. Generally, path-based methods calculate the path-level similarity for items and entities by encoding the predefined paths or meta-paths. However, extracting such paths is a time-consuming and expertise-intensive process.

2.1.3 Unified methods

To fully exploit the information in KG, the unified methods [26, 38, 42, 66] are proposed to integrate the semantic and connectivity information [46]. For example, RippleNet [55] simulates the phenomenon of water wave energy propagation and propagates user preference over the set of KG entities by automatically and iteratively extending the user's potential interests along with the relations. KGCN [56] is an end-to-end framework that discovers both high-order structure information and semantic information of the KG and then considers the neighborhood information when calculating the representation of a given entity. KGAT [51] propagates the embeddings from the node's high-order neighbors to the central node, and employs an attention mechanism to discriminate the importance of neighbors. Recently, MVIN [46] is proposed to learn the item representation from both user-view and entity-view through a novel wide and deep GCN. The unified recommendation methods have become a popular trend to fully exploit the information of KGs [23, 28].

2.1.4 Summary

Through investigating the related works on KG-aware recommendation, we find that the multi-level heterogeneity hidden in the items and their neighboring entities, which preserves user's fine-grained interests, remains barely explored by existing methods. To fill up this gap, we propose **McHa** to provide new insight into exploring more information in KG. To the best of our knowledge, **McHa** is the first method to exploit the multi-level heterogeneity information through aggregating the representations of multistage clusters and their elements with the hierarchical attention layers.

2.2 Other topics of recommendation

2.2.1 Community detection for recommendation

Community detection is to discover subgraphs from a network where the nodes share similar characteristics as well as patterns [29, 32]. It has been applied to many tasks, such as recommender systems, biochemistry, and online social network analysis, etc [43]. In the recommender systems, users with similar interests or preferences can be treated as members of a community. Detecting heterogeneous communities can help recommender systems capture users' differentiated preferences and thus provide personalized recommendations. For example, Eissa et al. [10] proposes a novel recommendation model based on interest-based communities generated from topic-based attributed social networks. SimClusters [40] is a novel recommendation algorithm based on the bipartite communities detected via Metropolis-Hastings sampling technology. Recently, LA-ALS [35] is proposed based on the Louvain's community detection algorithm and alternating least square algorithm. Specifically, Louvain's community detection algorithm is used to recognize the relationship between users to enhance the ability of the recommendation model.

2.2.2 Explainable recommendation

Explainable recommendations [33, 52, 59] have attracted increasing attention as they could improve the persuasiveness of recommendation results. The advances of KGs have made it possible to provide explainable recommendations through integrating graph embedding learning and recommendation techniques [9]. Within this field, KPRN [52], PGPR [58], and PeRN [22] perform reasoning over the paths extracted from KGs to improve the causal inference of recommendations with interpretability. Further, Xie et al. [60] design a novel multi-objective optimization function to jointly optimize the precision, diversity, and explainability of recommendations. Besides, some researchers have tried to derive interpretability from auxiliary information, such as attribute [5], aspect [17], and sentiment [63] etc. For example, AMCF [60] incorporates a novel feature mapping approach to map the uninterpretable general features onto the interpretable aspect features. Another important line of research is to introduce ontology mechanisms into RS to explore the interpretability reflected by the discriminative attention weights. For example, to provide explanations tailored for different target items, Seo et al. [41] and Chen et al. [6] adopt attention mechanisms to derive the importance of different review sentences under the supervision of user-item rating information.

2.2.3 Fairness in recommendation

Recently, research on fair recommendation has drawn a growing interest. There are some efforts [12, 15] on alleviating the unfairness problem of RS. For example, Fu et al. [11] quantify the unfairness in terms of KG path diversity as well as the recommendation performance disparity. Then, a fairness-aware algorithm is proposed so as to produce high-quality explainable recommendations with fairness. Mansoury et al. [34] propose a graph-based algorithm, namely FairMatch, for improving recommendation fairness. It maintains the recommendation lists updated with the items that are rarely recommended yet are high-quality. However, with the change of item popularity and user engagement, such fairness-aware methods can not cope with the dynamic fairness problem. To address this limitation, Ge et al. [14] propose FCPO to capture the long-term dynamic fairness through a fairness-constrained reinforcement learning framework. In detail, they leverage the Constrained Policy Optimization (CPO) with adapted neural network architecture to automatically learn the optimal policy under different fairness constraints.

3 Methodology

3.1 Problem formulation

In knowledge graph-aware recommendation, we let $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ and $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$ denote the user set and item set, respectively. The user-item interaction matrix is represented as $\mathcal{Y} = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$, where

$$y_{uv} = \begin{cases} 1, & \text{if } u \text{ has an interaction with } v; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

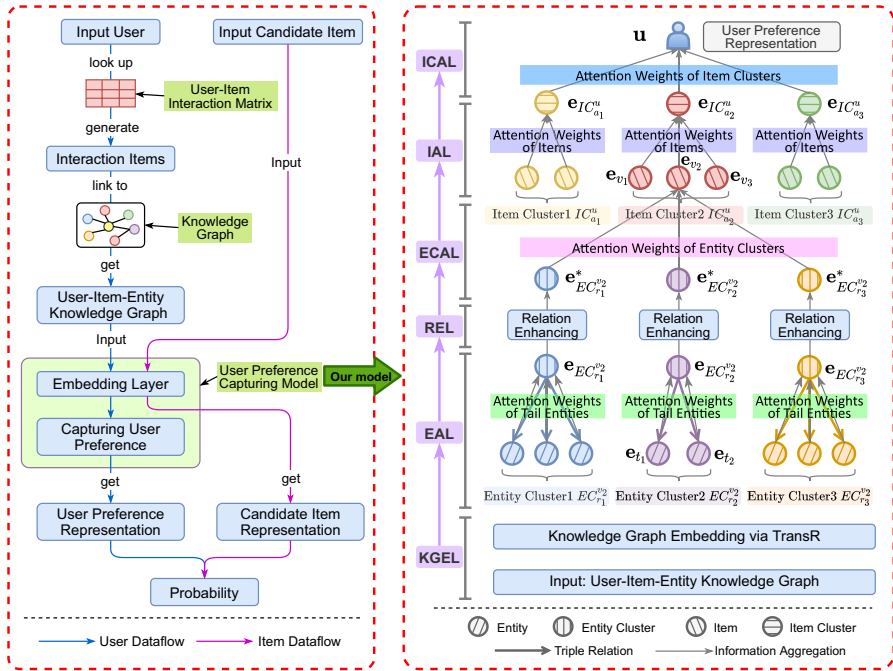
$y_{uv} = 1$ means that the user u has an implicit interaction with the item v , such as clicking, watching and browsing, etc. Additionally, we have a side knowledge graph \mathcal{KG} , which is comprised of triples (h, r, t) . Here, h , r , and t represent the head entity, relation, and tail entity, respectively. Given the input user u , input candidate item v , user-item interaction matrix \mathcal{Y} , and knowledge graph \mathcal{KG} , the goal of our model is to train a prediction model $\hat{y}_{uv} = \mathcal{F}(u, v)$ to predict the probability \hat{y}_{uv} that the user u would adopt the candidate item v .

In detail, as shown in the left part of Figure 2, for each input user $u \in \mathcal{U}$, we can obtain the interaction record of user u by looking up the user-item interaction matrix \mathcal{Y} . The interaction record \mathcal{I} can be formulated as

$$\mathcal{I} = \{v_1, \dots, v_i, \dots, v_{|\mathcal{I}|}\}, \quad v_i \in \mathcal{V} \text{ and } y_{uv_i} = 1. \quad (2)$$

We link all items in the interaction record \mathcal{I} to the entities or attributes of the side KG to generate UIEKG \mathcal{G} . Then, the UIEKG is fed into the user preference capturing model to calculate the final preference representation (denoted by \mathbf{u}) for the input user u . Accordingly, we could feed the input candidate item v into the knowledge graph embedding layer to obtain the candidate item representation (denoted by \mathbf{v}). After that, we calculate the probability \hat{y}_{uv} by inputting \mathbf{u} and \mathbf{v} into a mapping function $f : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$:

$$\hat{y}_{uv} = f(\mathbf{u}, \mathbf{v}). \quad (3)$$



(1) The overview of KG aware recommendation task. (2) The framework of McHa. The calculating process is from bottom to top.

Fig. 2 Graphical depictions of (1) the overview of KG-aware recommendation task in the left part, (2) the framework of McHa in the right part. The abbreviations in this figure are described as follows: KGEL is the knowledge graph embedding layer, EAL is the entity-level attention layer, REL is the relation enhancing layer, ECAL is the entity cluster-level attention layer, IAL is the item attention layer, and ICAL is the item cluster-level attention layer. e_{t_1} stands for the embedding vector of tail entity t_1 . $e_{EC_r_2^2}$ is the representation vector of entity cluster, in which the tail entities share the same head entity (item) v_2 and relation r_2 . e_{v_2} is the representation vector of item v_2 . $e_{IC_{a_2}^u}$ denotes the representation vector of item cluster, in which the items have been interacted with the same user u and share the same attribute a_2 . u represents the final user preference vector

Capturing user preference is the most important part of the knowledge graph-aware recommender systems. An ideal recommendation model should capture the user’s potential interests as accurately as possible. To achieve this, we propose a new KG-aware recommendation model, namely **McHa** (depicted in the right part of Figure 2). Different from that the existing KG-aware recommendation methods directly aggregate neighboring entities to the central item and aggregate items to the user, we additionally deploy the entity cluster-level attention layer between the neighboring entities and the central item and the item cluster-level attention layer between the interaction items and the user to capture the user’s more fine-grained potential interests. We will present our model detailedly in the following Sections 3.2–3.8.

3.2 Knowledge graph embedding layer (KGEL)

As shown in Figure 2, knowledge graph embedding (KGE) layer is to represent entities and relations as vectors to preserve the structural and semantic information in KG. Many attempts have been made for KGE, such as TransE [3], TransH [53], and TransR [27], etc.

In our work, we use TransR to learn the embeddings of the entities and relations in KG because of its superiority in dealing with the multi-relational space projection between head entity and tail entity. Semiotically, we let $\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t$ denote the embeddings of h, r, t for a triple (h, r, t) in UIEKG, respectively. The embeddings \mathbf{e}_h and \mathbf{e}_t in the entity space are projected into the relation space by the r -aware parameter \mathbf{W}_r :

$$\mathbf{e}_h^r = \mathbf{W}_r \mathbf{e}_h, \quad \mathbf{e}_t^r = \mathbf{W}_r \mathbf{e}_t, \quad (4)$$

where $[\mathbf{e}_h, \mathbf{e}_t] \in \mathbb{R}^d, \mathbf{e}_r \in \mathbb{R}^{d_r}$ and $\mathbf{W}_r \in \mathbb{R}^{d_r \times d}$. According to the principle of TransR, we have “ $h + r \approx t$ ”, which means that h can be translated into t through the bridge r . Therefore, the energy score of the triple (h, r, t) can be evaluated by

$$f(h, r, t) = \|\mathbf{e}_h^r + \mathbf{e}_r - \mathbf{e}_t^r\|_2^2. \quad (5)$$

A lower score of $f(h, r, t)$ means that the head entity and its tail entity are more closely in the relation space. It should be noted that the items in UIEKG can be regarded as entities when performing knowledge graph embedding.

3.3 Entity-level attention layer (EAL)

3.3.1 Entity cluster extraction

As shown in Figure 2, given an interaction item $v \in \mathcal{I}$, we can extract many triples that equip the item v as the head entity from the UIEKG. The extracted triples share the same head entity, meanwhile have different relations. We group these tail entities that share the same item (head entity) into several entity clusters (e.g. *Entity Cluster_{actor}* and *Entity Cluster_{director}*) according to their different relations for exploring heterogeneity information in terms of triple’s relation. Entity cluster can be defined as:

Definition 1 *Entity Cluster* (EC_r^v): A group of entities that share the same head entity v under relation r :

$$EC_r^v = \{t_1, t_2, t_3, \dots, t_{|EC_r^v|}\}. \quad (6)$$

Grouping entities according to their relations in this layer can allow our model to purposefully capture the user’s preference for the items with more subdivided characteristics.

3.3.2 Obtain entity cluster representation

After extracting entity cluster, we obtain the entity cluster representation through aggregating the elements in each entity cluster with the entity-level attention weights generated by GAT [48]. Specifically, we first obtain the attention score $s(t)$ for each element by

$$s(t) = \text{LeakyReLU}(\mathbf{W}_{EAL_2} \cdot [\mathbf{W}_{EAL_1} \mathbf{e}_h || \mathbf{W}_{EAL_1} \mathbf{e}_t]), \quad (7)$$

where $\mathbf{W}_{EAL_1} \in \mathbb{R}^{d_r \times d}$ and $\mathbf{W}_{EAL_2} \in \mathbb{R}^{1 \times 2d_r}$ are the learning parameters for the feature augmenting and $[\cdot || \cdot]$ is the concatenating operation for two vectors. A single layer perceptron with *LeakyReLU* activation function is applied to map the latent vector $[\mathbf{W}_{EAL_1} \mathbf{e}_h || \mathbf{W}_{EAL_1} \mathbf{e}_t]$ to the real number $s(t)$. We chose *LeakyReLU* activation function since it attempts to fix the “dying ReLU problem” [31] with a small negative slope instead of zero when the input

value < 0 . By normalizing the attention score $s(t)$ via *softmax* function, we get the attention weight:

$$\alpha(t) = \frac{\exp(s(t))}{\sum_{t \in EC_r^v} \exp(s(t))}. \tag{8}$$

The entity-level attention weight $\alpha(t)$ suggests which neighboring tail entities should be paid more attention when capturing the collaborative information. Finally, we obtain the entity cluster representation through aggregating the embedding vectors of all elements in entity cluster EC_r^v :

$$\mathbf{e}_{EC_r^v} = \sum_{t \in EC_r^v} \alpha(t) \mathbf{W}_{EAL_1} \mathbf{e}_t. \tag{9}$$

$\mathbf{e}_{EC_r^v}$ is the final entity cluster representation that preserves the heterogeneity information in terms of triple relation r .

3.4 Relation enhancing layer (REL)

According to Definition 1, the entities headed with item v can be grouped into different entity clusters according to different relations, which gives each entity cluster different heterogeneity in terms of triple’s relation. To further enhance the heterogeneity of each entity cluster, we explicitly encode the embedding of relation r into the entity cluster representation $\mathbf{e}_{EC_r^v}$. As shown in Figure 2, the relation enhancing process can be formulated as

$$\mathbf{e}'_{EC_r^v} = \mathbf{e}_{EC_r^v} \odot \mathbf{e}_r, \tag{10}$$

$$\mathbf{e}^*_{EC_r^v} = \sigma(\mathbf{W}_{REL} \mathbf{e}'_{EC_r^v}). \tag{11}$$

We first obtain the latent representation $\mathbf{e}'_{EC_r^v} \in \mathbb{R}^d$ by performing the element-wish projection between the entity cluster representation $\mathbf{e}_{EC_r^v}$ and the relation embedding \mathbf{e}_r . Then, we use a fully connected layer with *sigmoid* activation function to compress the latent representation $\mathbf{e}'_{EC_r^v}$ into $\mathbf{e}^*_{EC_r^v}$. $\mathbf{W}_{REL} \in \mathbb{R}^{d_r \times d}$ is the learning parameter. $\mathbf{e}^*_{EC_r^v} \in \mathbb{R}^d$ is the final entity cluster representation enhanced with the relation information.

3.5 Entity cluster-level attention layer (ECAL)

As depicted in the right part of Figure 2, we assume that the entities headed with the item v can be grouped into several entity clusters according to different relations, which can be formulated as

$$S_v = \{EC_{r_1}^v, EC_{r_2}^v, EC_{r_3}^v, \dots, EC_{r_{|S_v|}}^v\}, \tag{12}$$

where S_v is the entity cluster set of item v . Not all entity clusters equally contribute to the central item representation. For example, if the desire of the user to watch a movie largely depends on who acted in this movie rather than who directed it, the entity cluster *Entity Cluster_{actor}* should be paid more attention than *Entity Cluster_{director}* in the process of capturing the user’s preference.

Motivated by the above analysis, we calculate the representation of item v by differently aggregating the representations of all entity clusters of item v . In detail, we apply the entity cluster-level attention mechanism to discriminate those informative and uninformative entity clusters, which can be formulated as

$$\mathbf{e}_v = \sum_{EC_r^v \in S_v} \alpha(EC_r^v) \mathbf{e}_{EC_r^v}^* \tag{13}$$

$$\alpha(EC_r^v) = \frac{\exp(\mathbf{s}(EC_r^v)^T \mathbf{s}_{ECAL})}{\sum_{EC_r^v \in S_v} \exp(\mathbf{s}(EC_r^v)^T \mathbf{s}_{ECAL})} \tag{14}$$

$$\mathbf{s}(EC_r^v) = \tanh(\mathbf{W}_{ECAL} \mathbf{e}_{EC_r^v}^* + \mathbf{b}_{ECAL}). \tag{15}$$

Inspired by [61], we first utilize a single-layer feedforward neural network with the *tanh* activation function to calculate the hidden representation $\mathbf{s}(EC_r^v)$ of the entity cluster EC_r^v . $\mathbf{W}_{ECAL} \in \mathbb{R}^{d' \times d}$ and $\mathbf{b}_{ECAL} \in \mathbb{R}^{d'}$ are the learning weight matrix and bias, respectively. We chose *tanh* activation function since it can solve the non zero-centered problem of popular *sigmoid* function by squashing a real-valued number to the range $[-1, 1]$. Then, the entity cluster-level attention weight $\alpha(EC_r^v)$ is calculated by normalizing the projection between $\mathbf{s}(EC_r^v)$ and $\mathbf{s}_{ECAL} \in \mathbb{R}^{d'}$ via *softmax* function. \mathbf{s}_{ECAL} can be regarded as the entity cluster-level context vector. Finally, we calculate the item representation \mathbf{e}_v by aggregating all $\mathbf{e}_{EC_r^v}^*$ in S_v with the entity cluster-level attention weights. The vector \mathbf{e}_v is the final representation of item v that summarizes the information of all entity clusters that equip this item as the head entity.

3.6 Item-level attention layer (IAL)

3.6.1 Item cluster extraction

Similar to the entity cluster extraction, we group the items in the user’s interaction record I into different item clusters according to their attributes (e.g., genres for movies). Item cluster can be defined as:

Definition 2 *Item Cluster (IC_a^u)*: A group of items that share the same user u and attribute a :

$$IC_a^u = \{v_1, v_2, v_3, \dots, v_{|IC_a^u|}\}. \tag{16}$$

The purpose of grouping items into different item clusters in this layer is to allow our model to exploit the heterogeneity of different item clusters in terms of the item’s attribute and strengthen the pertinence of the user preference.

3.6.2 Obtain item cluster representation

As presented in the right part of Figure 2, to obtain the representation of item cluster IC_a^u , we aggregate the representations of all items in IC_a^u based on the item-level attention mechanism, which can be formulated as

$$\mathbf{e}_{IC_a^u} = \sum_{v \in IC_a^u} \alpha(v) \mathbf{e}_v, \tag{17}$$

$$\alpha(v) = \frac{\exp(\mathbf{s}(v)^T \mathbf{s}_{IAL})}{\sum_{v \in IC_a^u} \exp(\mathbf{s}(v)^T \mathbf{s}_{IAL})}, \tag{18}$$

$$\mathbf{s}(v) = \tanh(\mathbf{W}_{IAL} \mathbf{e}_v + \mathbf{b}_{IAL}), \tag{19}$$

where $\mathbf{W}_{IAL} \in \mathbb{R}^{d' \times d}$, $\mathbf{b}_{IAL} \in \mathbb{R}^{d'}$ and $\mathbf{s}_{IAL} \in \mathbb{R}^{d'}$ are the learning parameters. $\alpha(v)$ is the item-level attention weight. The vector $\mathbf{e}_{IC_a^u}$ is the item cluster representation that summarizes the information of all items in the item cluster IC_a^u .

3.7 Item cluster-level attention layer (ICAL)

In this layer, the interaction items of the user u can be grouped into different item clusters, which can be formulated as

$$S_u = \{IC_{a_1}^u, IC_{a_2}^u, IC_{a_3}^u, \dots, IC_{a_{|S_u|}}^u\}, \tag{20}$$

where S_u is the item cluster set of user u . As we discussed in Section 1, user may have different interests for different movie clusters. Therefore, the user preference representation \mathbf{u} can be obtained by discriminatorily aggregating the representations of all item clusters of user u based on the item cluster-level attention mechanism, which can be formulated as

$$\mathbf{u} = \sum_{IC_a^u \in S_u} \alpha(IC_a^u) \mathbf{e}_{IC_a^u}, \tag{21}$$

$$\alpha(IC_a^u) = \frac{\exp(\mathbf{s}(IC_a^u)^T \mathbf{s}_{ICAL})}{\sum_{IC_a^u \in S_u} \exp(\mathbf{s}(IC_a^u)^T \mathbf{s}_{ICAL})}, \tag{22}$$

$$\mathbf{s}(IC_a^u) = \tanh(\mathbf{W}_{ICAL} \mathbf{e}_{IC_a^u} + \mathbf{b}_3), \tag{23}$$

where $\mathbf{W}_{ICAL} \in \mathbb{R}^{d' \times d}$, $\mathbf{b}_{ICAL} \in \mathbb{R}^{d'}$ and $\mathbf{s}_{ICAL} \in \mathbb{R}^{d'}$ are the learning parameters. $\alpha(IC_a^u)$ is the item cluster-level attention weight. The vector \mathbf{u} is the final user preference representation.

3.8 Probability prediction

So far, we have obtained the final user preference representation \mathbf{u} of user u . Given a candidate item v , we feed it into the knowledge graph embedding layer to obtain the candidate item representation \mathbf{v} . Hereafter, the probability \hat{y}_{uv} that user u would adopt candidate item v is calculated by feeding \mathbf{u} and \mathbf{v} into the following equation:

$$\hat{y}_{uv} = \sigma(\mathbf{u}^T \mathbf{v}), \tag{24}$$

where $\sigma(\cdot)$ is the *sigmoid* function. \hat{y}_{uv} is the final output of our model.

3.9 Learning algorithm

3.9.1 Loss function

In the training process of knowledge graph embedding (KGE), we learn the embeddings of entities and relations in UIEKG \mathcal{G} by optimizing the BPR [39] loss with L_2 regularization, which can be formulated as

$$\mathcal{L}_{\text{KGE}} = \sum_{\substack{(h,r,t) \in \mathcal{G} \\ (h,r,t') \notin \mathcal{G}}} -\ln \sigma(f(h,r,t') - f(h,r,t)) + \lambda \|\Theta_{\text{KGE}}\|_2^2. \quad (25)$$

\mathcal{L}_{KGE} is the knowledge graph embedding loss. In detail, the first term is the BPR loss, where (h, r, t') is the negative triple generated by negative sampling for tail entity, and $f(\cdot)$ (see (5)) is the energy function for evaluating the plausibility of a triple. The second term is the L_2 regularizer with the coefficient λ for preventing overfitting and Θ_{KGE} (including \mathbf{W}_r) stands for the parameter set for training KGE. \mathcal{G} stands for the user-item-entity knowledge graph.

In the training process of recommendation model (RM), we adopt the cross-entropy loss with L_2 regularization to optimize the learning parameters, which can be formulated as

$$\mathcal{L}_{\text{RM}} = - \sum_{(u,v) \in \mathcal{P}} (y_{uv} \log(\hat{y}_{uv}) + (1 - y_{uv}) \log(1 - \hat{y}_{uv})) + \lambda \|\Theta_{\text{RM}}\|_2^2. \quad (26)$$

\mathcal{L}_{RM} is the recommendation loss. In detail, the first term is the cross-entropy loss, where \mathcal{P} stands for the mixed training interactions including the observed interactions and the unobserved (negative) interactions generated by negative sampling strategy, and \hat{y}_{uv} (see (24)) is the CTR probability. The second term is the L_2 regularizer with the coefficient λ and Θ_{RM} (including $\mathbf{W}_{\text{EAL}_1}$, $\mathbf{W}_{\text{EAL}_2}$, \mathbf{W}_{REL} , \mathbf{W}_{ECAL} , \mathbf{b}_{ECAL} , \mathbf{s}_{ECAL} , ...) stands for the parameter set for training recommendation model.

3.9.2 Training strategy

Inspired by [51], we optimize \mathcal{L}_{KGE} and \mathcal{L}_{RM} alternatively with the widely used optimizer-Adam [24]. We chose Adam since it keeps the learning rate adaptive. The learning algorithm of our model is presented in Algorithm 1. For every epoch of training, we perform KGE training (corresponding to lines 3-8) and recommendation model training (corresponding to lines 9-24) alternately.

Algorithm 1: Learning Algorithm of McHa

Input: User-Item-Entity Knowledge Graph \mathcal{G} ,
User-Item Interactions \mathcal{P} .
Output: Prediction function $\mathcal{F}(u, v; \Theta_{\text{KGE}}, \Theta_{\text{RM}})$.

- 1 Initialize all learning parameters.
- 2 **for** every epoch **do**
- 3 # Training for Knowledge Graph Embedding
- 4 **for** every batch of \mathcal{G} **do**
- 5 sample negative triples (h, r, t') .
- 6 calculate KGE loss \mathcal{L}_{KGE} via Eq. (5) and Eq. (25).
- 7 update Θ_{KGE} of \mathcal{F} according to the gradient descent of \mathcal{L}_{KGE} .
- 8 **end**
- 9 # Training for Recommendation Model
- 10 **for** every batch of \mathcal{P} **do**
- 11 extract entity cluster set $S_v = \{EC_{r_1}^v, EC_{r_2}^v, \dots, EC_{r_{|S_v|}}^v\}$ for
 every item.
- 12 extract item cluster set $S_u = \{IC_{a_1}^u, IC_{a_2}^u, \dots, IC_{a_{|S_u|}}^u\}$ for
 every user.
- 13 **for** every entity cluster EC in S_v **do**
- 14 calculate entity cluster representation \mathbf{e}_{EC} via Eq. (9)-(7).
- 15 obtain \mathbf{e}_{EC}^* enhanced by relation via Eq. (10)-(11).
- 16 **end**
- 17 calculate item representation \mathbf{e}_v over S_v via Eq. (13)-(15).
- 18 **for** every item cluster IC in S_u **do**
- 19 calculate item cluster representation \mathbf{e}_{IC} via Eq. (17)-(19).
- 20 **end**
- 21 calculate user preference vector \mathbf{u} over S_u via Eq. (21)-(23).
- 22 calculate recommendation loss \mathcal{L}_{RM} via Eq. (24) and Eq. (26).
- 23 update Θ_{RM} of \mathcal{F} according to the gradient descent of \mathcal{L}_{RM} .
- 24 **end**
- 25 **end**
- 26 **Return:** Prediction function $\mathcal{F}(u, v; \Theta_{\text{KGE}}, \Theta_{\text{RM}})$.

4 Experiments

4.1 Datasets

We choose the following three widely used benchmark datasets of recommendation tasks to evaluate our model.

- **MovieLens-1M**¹ is a movie rating dataset widely used in recommendation task. It includes ratings (ranging from 1 to 5) for movies and demographic data (age, gender, and occupation, etc.) about users.

¹ <https://grouplens.org/datasets/movielens/1m/>

Table 1 Statistics of the three benchmark datasets. # stands for the number

	MovieLens-1M	Last.FM	Book-Crossing
# Users	6,036	1,872	17,860
# Items	2,445	3,846	14,967
# Interactions	753,772	42,346	139,746
# KG Entities	182,011	9,366	77,903
# KG Relations	12	60	25
# KG Triples	1,241,995	15,518	151,500

- **Last.FM**² is a dataset collected from an online music website for providing music recommendations. This dataset includes the listened artist records of users and the metadata about users and artists.
- **Book-Crossing**³ is a book rating dataset from Book-Crossing community. It includes the ratings (ranging from 0 to 10) for books and the metadata about users and books.

The statistics of the three benchmark datasets are shown in Table 1. As suggested in [46, 55], we convert the ratings in MovieLens-1M and Book-Crossing into binary feedback. Each entry is marked as 1 if the item had been positively rated by the user. Practically, the rating threshold of MovieLens-1M is set to 4, which means that if the rating score is not smaller than 4, the entry is marked as 1. While no threshold is set for Book-Crossing due to its sparsity of interactions, which means that if the entry is observed, it is marked as 1. For Last.FM, the user-artist entry is marked as 1 if it is recorded in the listened artist records. For the three benchmark datasets, these entries marked as 1 are regarded as the observed interactions. Accordingly, we randomly sample unobserved interactions marked as 0 for each user, which is of equal size with the observed interactions. We split the mixed interactions including the observed and unobserved interactions into training, validation, and test datasets with the ratio of 6:2:2. We train our model on the training data, tune hyper-parameters on the validation data, and evaluate the performance of our model on the test data. Following [46, 55, 56], we use Microsoft Satori⁴ to construct the UIEKG for each dataset. Specifically, we link the items to the entities by matching their names with the confidence level > 0.9 . For MovieLens-1M and Last.FM, we group the interaction items (movies and artists) of the user into item clusters according to their genres, while for Book-Crossing, we group the items (books) into item clusters according to their authors.

4.2 Baselines

We choose the following representative or state-of-the-art models as baselines:

- **SVD++** [25] is an improved version of Singular Value Decomposition (SVD), which considers the user's implicit feedback to the item.

² <https://grouplens.org/datasets/hetrec-2011/>

³ <http://www2.informatik.uni-freiburg.de/~ziegler/BX/>

⁴ <https://searchengineland.com/library/bing/bing-satori>

Table 2 Hyper-parameter settings of **McHa**

Dataset	d	d_r	d'	$ EC $	$ S_v $	$ IC $	$ S_u $	λ
MovieLens-1M	16	32	16	4	5	6	4	1e-4
Last.FM	16	16	16	3	5	3	5	1e-2
Book-Crossing	32	16	8	8	4	6	4	1e-3

- **CKE** [64] is a unified framework that combines collaborative filtering with knowledge base embedding to learn different representations jointly.
- **MKR** [50] builds several cross and compress units, which automatically share latent features and learn high-order interactions between items in recommender systems and entities in the knowledge graph.
- **KGCN** [56] captures inter-item relatedness effectively by mining their associated attributes in KG. Besides, it samples from the neighbors for each entity in KG and then combines the neighborhood information when calculating the representation of a given entity.
- **KGAT** [51] is a model that propagates the embeddings from the node's high-order neighbors to the central node and employs an attention mechanism to discriminate the importance of the neighbors.
- **MVIN** [46] improves representations of items from both the user view gathering personalized knowledge information and the entity view considering the difference among layers.
- **RippleNet** [55] propagates user preferences over the set of entities by extending a user's potential interests along links extracted from KG.
- **FairGo** [57] is a model-agnostic framework, which considers fairness from a user-item bipartite graph perspective. In detail, it eliminates the unfairness through a graph-based adversarial training process.

It should be noted that the hyper-parameters of baselines are set to the default or recommended parameters in the published literature.

4.3 Experiment setup

4.3.1 Hyper-parameters

The hyper-parameter settings are listed in Table 2. In detail, d and d_r stand for the embedding dimension of entity and relation, respectively, and d' is the dimension of the context vector. We let $|EC|$ and $|S_v|$ denote the number of entity elements in each entity cluster and the number of entity clusters in each S_v , respectively. Similarly, $|IC|$ and $|S_u|$ are the number of item elements in each item cluster and the number of item clusters in each S_u , respectively. It should be noted that the size of EC , S_v , IC , and S_u are not fixed for each user. As suggested in [55], we apply the sampling strategy to fix these unfixed sizes for every user. λ is the regularization coefficient. The batch size and learning rate are set to 128 and 0.001 for both KGE and recommendation training. The hyper-parameters given in this paper are selected by grid search.

4.3.2 Evaluation metrics

For CTR prediction task, we use the metrics of *AUC*, *ACC* and *F1-score* to evaluate the performance of our model. For top-*N* recommendation task, we adopt the metrics of *Precision@N*, *Recall@N* and *F1-score@N* to evaluate the ability of our model in selecting *N* highest click probability items for the user. Each experiment is repeated 5 times, and the average results (*mean*) with standard deviation (*std*) on the test dataset are reported.

4.4 Results and discussion

We evaluate our model in two recommendation tasks: (1) CTR prediction, and (2) top-*N* recommendation. We have the following observations.

4.4.1 CTR prediction task

As shown in Table 3, our model has achieved the best performance in the CTR prediction task, compared with baselines. Specifically, the performance has been averagely improved by 2.3%, 5.0%, and 10.7% of *F1-score* on MovieLens-1M, Last.FM, and Book-Crossing, respectively. Compared with our model, KGCN, KGAT, MVIN, and RippleNet achieve poorer performances probably because the noisy information of the irrelevant high-order nodes could be unintentionally introduced and amplified step by step during the information propagation in these methods. CKE does not perform well when missing the visual embedding, compared with other KG-aware methods. Besides, due to the lack of external information, SVD++ achieves poorer performance, especially in face of sparser data (e.g., Last.FM and Book-Crossing). Although FairGo attempts to improve the recommendation performance via mitigating the unfairness issue, it doesn't perform well compared with KG-aware methods due to the lack of external information provided by KG.

4.4.2 Top-N recommendation task

As shown in Figure 3, our model has also achieved the best performance compared with baselines. Given the fact that Last.FM is a smaller dataset than MovieLens-1M and Book-Crossing, the outstanding improvement of our model performance on this dataset indicates that our model has strong adaptability when facing a smaller dataset in top-*N* recommendation task.

4.5 Ablation study

4.5.1 Ablation setup

In this part, we conduct the ablation experiment to prove the positive effect of every attention layer in **McHa**. Experimentally, we perform the ablation by replacing every attention layer of **McHa** with the single-layer feedforward neural network with *tanh* activation function. For the ablation of relation enhancing layer, we only eliminate \mathbf{e}_r in (10). We use abbreviations to represent **McHa**'s variants. For example, we let the abbreviation " $\text{McHa}_{\text{w/o EAL}}$ " denote **McHa** with the ablation of Entity-level Attention Layer (EAL).

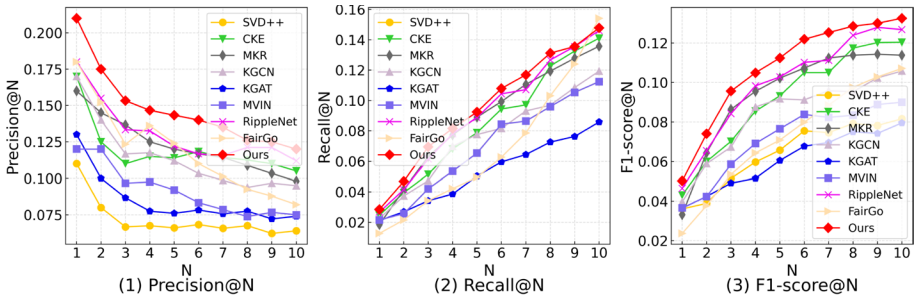
Table 3 Results (*Mean ±std* of testing 5 times) for the CTR prediction task

Methods	<i>AUC</i>	<i>ACC</i>	<i>F1-score</i>
(a) Results on MovieLens-1M (To be continued).			
SVD++	0.892±0.011	0.815±0.010	0.810±0.013
CKE	0.896±0.005	0.822±0.004	0.829±0.007
MKR	0.912±0.010	0.838±0.009	0.841±0.011
KGCN	0.902±0.006	0.827±0.004	0.834±0.004
KGAT	0.895±0.017	0.821±0.016	0.828±0.012
MVIN	0.923±0.009	0.849±0.015	0.853±0.013
RippleNet	0.921±0.007	0.845±0.005	0.848±0.006
FairGo	0.907±0.006	0.832±0.009	0.838±0.005
Ours	0.928± 0.008	0.853± 0.012	0.854± 0.011
Avg Improv.	+2.5%	+2.7%	+2.3%
(b) Results on Last.FM (Continued).			
SVD++	0.780±0.017	0.727±0.011	0.733±0.008
CKE	0.777±0.007	0.705±0.006	0.688±0.003
MKR	0.793±0.012	0.751±0.007	0.722±0.005
KGCN	0.796±0.011	0.739±0.008	0.723±0.013
KGAT	0.792±0.007	0.722±0.010	0.726±0.009
MVIN	0.811±0.011	0.740±0.009	0.735±0.006
RippleNet	0.799±0.016	0.731±0.014	0.725±0.013
FairGo	0.796±0.009	0.729±0.010	0.700±0.007
Ours	0.823± 0.013	0.764± 0.007	0.757± 0.010
Avg Improv.	+3.8%	+4.9%	+5.0%
(c) Results on Book-Crossing (Continued).			
SVD++	0.696±0.014	0.647±0.009	0.663±0.012
CKE	0.723±0.002	0.635±0.004	0.656±0.004
MKR	0.733±0.004	0.703±0.006	0.659±0.010
KGCN	0.728±0.021	0.697±0.014	0.643±0.019
KGAT	0.703±0.013	0.649±0.018	0.646±0.018
MVIN	0.735±0.012	0.676±0.015	0.638±0.014
RippleNet	0.726±0.018	0.653±0.021	0.652±0.019
FairGo	0.716±0.015	0.655±0.012	0.661±0.013
Ours	0.798± 0.014	0.724± 0.011	0.722± 0.012
Avg Improv.	+10.9%	+9.1%	+10.7%

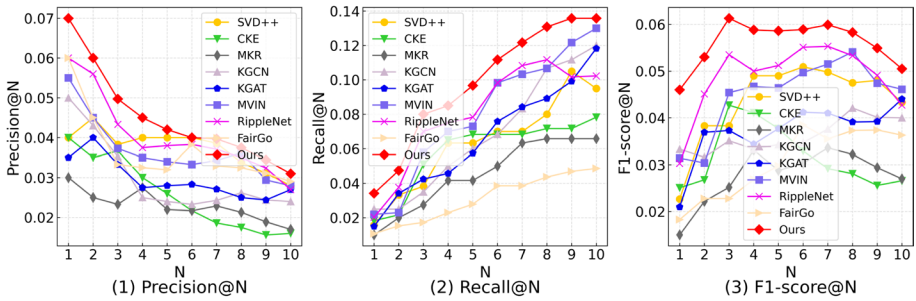
The best results are marked in boldface. Avg Improv. is calculated via averaging the improvement percentages of our model relative to the baseline methods

4.5.2 Ablation results

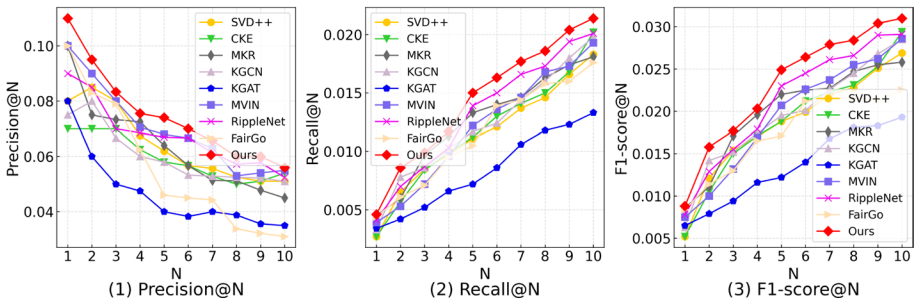
As shown in Table 4, **McHa** outperforms all variants. This observation demonstrates that every attention layer of our proposed framework has an essential and positive contribution to the performance of our model. Specifically, $\text{McHa}_{\text{w/o ECAL}}$ (**McHa** without Entity Cluster-level Attention Layer) and $\text{McHa}_{\text{w/o ICAL}}$ (**McHa** without Item Cluster-level Attention Layer)



(a) Precision@N, Recall@N, and F1-score@N for MovieLens-1M



(b) Precision@N, Recall@N, and F1-score@N for Last.FM



(c) Precision@N, Recall@N, and F1-score@N for Book-Crossing

Fig. 3 Results (Mean of testing 5 times) for the top-N recommendation task

achieve the poorer performance than other variants, which indicates that multistage clustering plays a significant positive role in capturing the user’s preference.

4.6 Parameter sensitivity analysis

4.6.1 Embedding dimension

We vary $d \in [4, 8, 16, 32, 64, 128]$ to study the influence of dimension in the knowledge graph embedding layer. As shown in Figure 4(a), increasing the dimension boosts the

Table 4 Ablation study results (*Mean ± std* of testing 5 times). “w/o” means without. The best results are reported in boldface

Methods	<i>AUC</i>	<i>ACC</i>	<i>F1-score</i>
(a) Results on MovieLens-1M (To be continued).			
McHa	0.928± 0.008	0.853± 0.012	0.854± 0.010
McHa _{w/o} ICAL	0.919±0.010	0.842±0.013	0.845±0.011
McHa _{w/o} IAL	0.921±0.004	0.845±0.004	0.847±0.005
McHa _{w/o} ECAL	0.915±0.019	0.837±0.016	0.841±0.017
McHa _{w/o} REL	0.925±0.008	0.850±0.012	0.851±0.010
McHa _{w/o} EAL	0.924±0.010	0.848±0.011	0.849±0.009
(b) Results on Last.FM (Continued).			
McHa	0.823± 0.013	0.764± 0.007	0.757± 0.010
McHa _{w/o} ICAL	0.810±0.010	0.725±0.008	0.733±0.009
McHa _{w/o} IAL	0.820±0.003	0.746±0.002	0.744±0.002
McHa _{w/o} ECAL	0.817±0.008	0.735±0.010	0.740±0.012
McHa _{w/o} REL	0.823±0.005	0.746±0.006	0.751±0.005
McHa _{w/o} EAL	0.820±0.008	0.748±0.008	0.749±0.006
(c) Results on Book-Crossing (Continued).			
McHa	0.798± 0.014	0.734± 0.011	0.722± 0.012
McHa _{w/o} ICAL	0.755±0.012	0.671±0.014	0.676±0.015
McHa _{w/o} IAL	0.778±0.006	0.698±0.005	0.701±0.004
McHa _{w/o} ECAL	0.754±0.011	0.668±0.013	0.674±0.013
McHa _{w/o} REL	0.765±0.015	0.680±0.017	0.683±0.014
McHa _{w/o} EAL	0.797±0.012	0.721±0.009	0.720±0.013

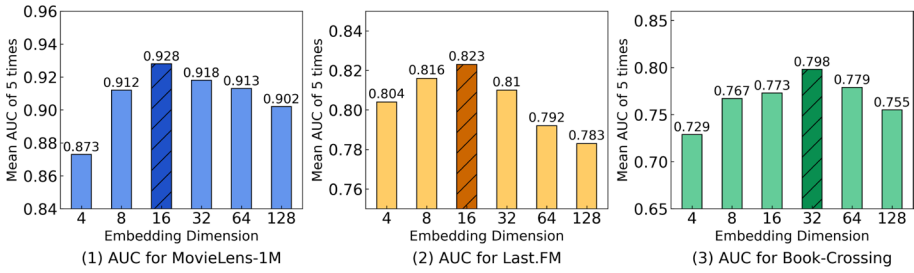
performance since a high-dimensional vector preserves more information. While, if the embedding dimension is too large, the model suffers from overfitting.

4.6.2 Number of entity clusters

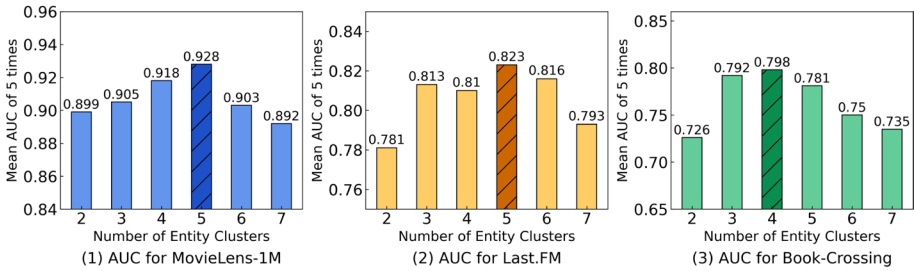
We vary $|S_v| \in [2, 3, 4, 5, 6, 7]$ to verify the influence of the number of entity clusters. As shown in Figure 4(b), performance would deteriorate when setting $|S_v|$ to the smaller or greater values than the ideal value. This observation can be explained as that a smaller or greater size of S_v generated by the negative sampling strategy would lead to the loss of information or the introduction of noisy entities, respectively. This observation implies that the performance of our model is sensitive to the number of entity clusters in S_v and grouping the entities into different entity clusters in our proposed model is effective for capturing the user’s fine-grained preferences.

4.6.3 Number of item clusters

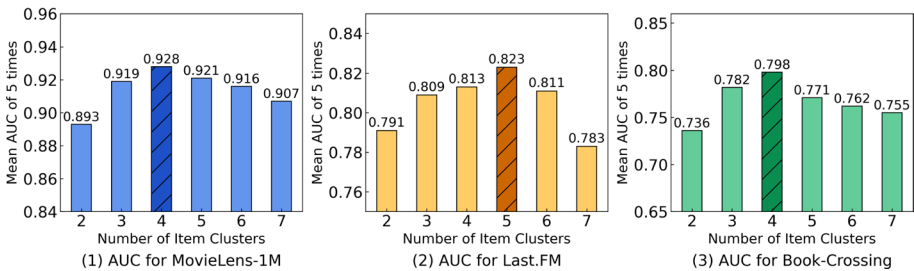
We vary $|S_u| \in [2, 3, 4, 5, 6, 7]$ to verify the influence of the number of item clusters. As shown in Figure 4(c), our model achieves the best results when setting $|S_u|$ to 4, 5, and 4 for MovieLens-1M, Last.FM, and Book-Crossing, respectively. This observation indicates that the performance of our model is sensitive to the number of item clusters in S_u . In another



(a) AUC under different embedding dimension.



(b) AUC under different numbers of entity clusters.



(c) AUC under different numbers of item clusters.

Fig. 4 Parameter sensitivity for embedding dimension d , number of entity clusters $|S_v|$, and number of item clusters $|S_u|$. The other hyper-parameters are fixed according to Table 2

word, properly grouping the items into different item clusters contributes to the recommendation performance positively.

4.7 Interpretability with case study

Prior works [7, 13] have shown that the attention mechanism can benefit and explain the recommendation results. On this basis, we provide a visual case to intuitively explain the recommendation results of our model. We randomly sample a user (User ID: 9) from MovieLens-1M. As shown in Figure 5, movies in this user’s viewed record extracted from MovieLens-1M are grouped into four item clusters by our model according to their genres. This case shows that $Item\ Cluster_{Comedy}$ and $Item\ Cluster_{Animation}$ are assigned with the biggest and smallest attention weight when calculating the user preference

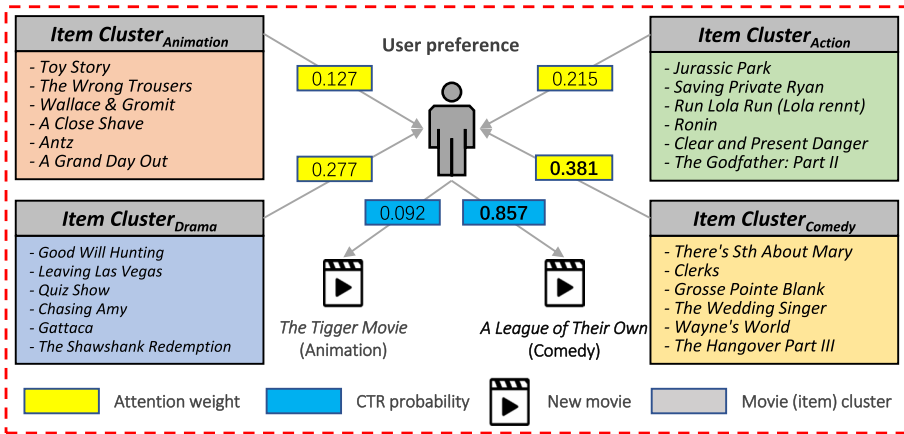


Fig. 5 A real case (User ID: 9) from MovieLens-1M

representation, respectively. This means that the fine-grained and focused information that the user is more interested in comedy rather than animation would be encoded into the user preference representation. To verify the effectiveness of such user preference, we feed two new candidate movies *The Tigger Movie* (an animation) and *A League of Their Own* (a comedy) into our model to calculate their CTR probabilities, respectively. The output results for these two candidate movies show that comedy *A League of Their Own* received a higher CTR probability (0.857) than animation *The Tigger Movie* (0.092), which demonstrates that the user preference captured by our model works. In summary, this case implies that our model could accurately generate the expressive user preference representation and the recommendation results can be explained by the attention weights.

5 Conclusion and future work

In this paper, we propose a novel KG-aware recommendation model, namely **McHa**. It overcomes the limitation that the more fine-grained and focused multi-level heterogeneity information remains barely exploited in existing methods. Specifically, we first capture the multi-level heterogeneity information by grouping the items and their connected entities into item clusters and entity clusters (jointly referred to as multistage clusters), respectively. Then, the user preference is obtained by hierarchically aggregating the multi-level heterogeneity information with the weights generated by the hierarchical attention layers. The extensive experiments show the effectiveness of our model.

However, further research is still needed. For example, we only consider the nearest (1-hop) entities around the item in the knowledge graph. How to extend our model to efficiently process the multi-hop entities needs to be further studied to explore more potential information in the knowledge graph.

Acknowledgements This work was supported by the National Key Research and Development Plan of China (No.2018YFB1003804) and the Project of State Grid Shandong Electric Power Company (2020A-135).

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References


1. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* **11**(9), 137 (2018)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., et al.: Dbpedia: A nucleus for a web of open data. In: *The Semantic Web*, pp. 722–735. Springer (2007)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: *Advances in Neural Information Processing Systems*, pp. 2787–2795 (2013)
4. Chen, H.-C., Wei, C.-P., Dai, Y.-S., Lin, Y.-K.: Exploiting item heterogeneity for collaborative filtering recommendation. In: *Proceedings of the 4th China Summer Workshop on Information Management* (2010)
5. Chen, T., Yin, H., Ye, G., Huang, Z., Wang, Y., Wang, M.: Try this instead: Personalized and interpretable substitute recommendation. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 891–900 (2020)
6. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: *Proceedings of the 2018 World Wide Web Conference*, pp. 1583–1592 (2018)
7. Chen, X., Zhang, Y., Qin, Z.: Dynamic explainable recommendation based on neural attentive models. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 53–60 (2019)
8. Chen, J., Yu, J., Lu, W., Qian, Y., Li, P.: Ir-rec: An interpretive rules-guided recommendation over knowledge graph. *Information Sciences* **563**, 326–341 (2021)
9. Chicaiza, J., Valdiviezo-Diaz, P.: A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information* **12**(6), 232 (2021)
10. Eissa, A.H.B., El-Sharkawi, M.E., Mokhtar, H.M.O.: Towards recommendation using interest-based communities in attributed social networks. In: *Companion Proceedings of the The Web Conference 2018*, pp. 1235–1242 (2018)
11. Fu, Z., Xian, Y., Gao, R., Zhao, J., Huang, Q., Ge, Y., Xu, S., Geng, S., Shah, C., Zhang, Y., et al.: Fairness-aware explainable recommendation over knowledge graphs. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 69–78 (2020)
12. Gao, R., Shah, C.: How fair can we go: Detecting the boundaries of fairness optimization in information retrieval. In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 229–236 (2019)
13. Gao, J., Wang, X., Wang, Y., Xie, X.: Explainable recommendation through attentive multi-view learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3622–3629 (2019)
14. Ge, Y., Liu, S., Gao, R., Xian, Y., Li, Y., Zhao, X., Pei, C., Sun, F., Ge, J., Ou, W. et al.: Towards long-term fairness in recommendation. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 445–453 (2021)
15. Geyik, S.C., Ambler, S., Kenthapadi, K.: Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In: *Proceedings of the 25th ACM Sigkdd International Conference on Knowledge Discovery & Data Mining*, pp. 2221–2231 (2019)
16. Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., et al.: A survey on knowledge graph-based recommender systems. [arXiv:2003.00911](https://arxiv.org/abs/2003.00911) (2020)
17. Hou, Y., Yang, N., Wu, Y., Yu, P.S.: Explainable recommendation with fusion of aspect information. *World Wide Web* **22**(1), 221–240 (2019)
18. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1531–1540 (2018)

19. Hu, X., Xu, J., Wang, W., Li, Z., Liu, A.: A graph embedding based model for fine-grained poi recommendation. *Neurocomputing* **428**, 376–384 (2021)
20. Huang J., et al.: Improving sequential recommendation with knowledge-enhanced memory networks. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 505–514 (2018)
21. Huang, X., Fang, Q., Qian, S., Sang, J., Li, Y., Xu, C.: Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In: *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 548–556 (2019)
22. Huang, Y., Zhao, F., Gui, X., Jin, H.: Path-enhanced explainable recommendation with knowledge graphs. *World Wide Web* **24**(5), 1769–1789 (2021)
23. Ji, S., Pan, S., Cambria, E., Martinen, P., Philip, S.Y.: A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst* (2021)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
25. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434 (2008)
26. Li, Q., Tang, X., Wang, T., Yang, H., Song, H.: Unifying task-oriented knowledge graph learning and recommendation. *IEEE Access* **7**, 115816–115828 (2019)
27. Lin, H., Liu, Y., Wang, W., Yue, Y., Lin, Z.: Learning entity and relation embeddings for knowledge resolution. *Procedia Computer Science* **108**, 345–354 (2017)
28. Liu, J., Duan, L.: A survey on knowledge graph-based recommender systems. In: *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 2450–2453. *IEEE* (2021)
29. Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., Yu, P.S.: Deep learning for community detection: Progress, challenges and opportunities. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, vol. 5, pp. 4981–4987 (2020)
30. Lops, P., Gemmis, M.D., Semeraro, G.: *Content-based recommender systems: state of the art and trends*. Springer, US (2011)
31. Lu, L., Shin, Y., Su, Y., Karniadakis, G.E.: Dying relu and initialization: Theory and numerical examples. [arXiv:1903.06733](https://arxiv.org/abs/1903.06733) (2019)
32. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.*, 1–1 (2021)
33. Ma, W., Zhang, M., Cao, Y., Jin, W., Wang, C., Liu, Y., Ma, S., Ren, X.: Jointly learning explainable rules for recommendation with knowledge graph. In: *The World Wide Web Conference*, pp. 1210–1221 (2019)
34. Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., Burke, R.: Fairmatch: A graph-based approach for improving aggregate diversity in recommender systems. In: *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 154–162 (2020)
35. Paleti, L., Krishna, P.R., Murthy, J.V.R.: Approaching the cold-start problem using community detection based alternating least square factorization in recommendation systems. *Evolutionary Intelligence* **14**(2), 835–849 (2021)
36. Palumbo, E., Rizzo, G., Troncy, R.: Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 32–36 (2017)
37. Qian, R.: Understand your world with bing. *Bing search blog* (2013)
38. Qu, Y., Bai, T., Zhang, W., Nie, J., Tang, J.: An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. In: *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pp. 1–9 (2019)
39. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. [arXiv:1205.2618](https://arxiv.org/abs/1205.2618) (2012)
40. Satuluri, V., Wu, Y., Zheng, X., Qian, Y., Wichers, B., Dai, Q., Tang, G.M., Jiang, J., Lin, J.: Sim-clusters: Community-based representations for heterogeneous recommendations at twitter. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3183–3193 (2020)
41. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 297–305 (2017)
42. Shafqat, W., Byun, Y.-C.: Incorporating similarity measures to optimize graph convolutional neural networks for product recommendation. *Applied Sciences* **11**(4), 1366 (2021)

43. Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., Sheng, Q.Z., Yu, P.S.: A comprehensive survey on community detection with deep learning. [arXiv:2105.12584](https://arxiv.org/abs/2105.12584) (2021)
44. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: The World Wide Web Conference, pp. 697–706 (2007)
45. Sun, Z., Yang, J., Zhang, J., Bozzon, A., Huang, L.-K., Xu, C.: Recurrent knowledge graph embedding for effective recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 297–305 (2018)
46. Tai, C.-Y., Wu, M.-R., Chu, Y.-W., Chu, S.-Y., Ku, L.-W.: Mvin: Learning multiview items for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 99–108 (2020)
47. Tang, X., et al.: Akupm: Attention-enhanced knowledge-aware user preference model for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1891–1899 (2019)
48. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. [arXiv:1710.10903](https://arxiv.org/abs/1710.10903) (2017)
49. Vrandečić, D.: Wikidata: A new platform for collaborative data collection. In: The World Wide Web Conference, pp. 1063–1064 (2012)
50. Wang, H., et al.: Multi-task feature learning for knowledge graph enhanced recommendation. In: The World Wide Web Conference, pp. 2000–2010 (2019)
51. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950–958 (2019)
52. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.-S.: Explainable reasoning over knowledge graphs for recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5329–5336 (2019)
53. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Aaai, vol. 14, pp. 1112–1119. Citeseer (2014)
54. Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Liu, Q.: Shine: Signed heterogeneous information network embedding for sentiment link prediction. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 592–600 (2018)
55. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: Ripplet: Propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 417–426 (2018)
56. Wang, H., Zhao, M., et al.: Knowledge graph convolutional networks for recommender systems. In: The World Wide Web Conference, pp. 3307–3313 (2019)
57. Wu, L., Chen, L., Shao, P., Hong, R., Wang, X., Wang, M.: Learning fair representations for recommendation: A graph-based perspective. In: Proceedings of the Web Conference 2021, pp. 2198–2208 (2021)
58. Xian, Y., Fu, Z., Muthukrishnan, S., De Melo, G., Zhang, Y.: Reinforcement knowledge graph reasoning for explainable recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 285–294 (2019)
59. Xian, Y., Zhao, T., Li, J., Chan, J., Kan, A., Ma, J., Dong, X.L., Faloutsos, C., Karypis, G., Muthukrishnan, S., Zhang Y.: Ex3: Explainable attribute-aware item-set recommendations. In: Fifteenth ACM Conference on Recommender Systems, pp. 484–494 (2021)
60. Xie, L., Hu, Z., Cai, X., Zhang, W., Chen, J.: Explainable recommendation based on knowledge graph and multi-objective optimization. *Complex & Intelligent Systems* 7(3), 1241–1252 (2021)
61. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
62. Yu, X., Ren, X., Sun, Y., Sturt, B., Khandelwal, U., et al.: Recommendation in heterogeneous information networks with implicit user feedback. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 347–350 (2013)
63. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: Proceedings of the 37th international ACM SIGIR conference on Research& Development in Information Retrieval, pp. 83–92 (2014)
64. Zhang, F., Yuan, N.J., Lian, D., et al.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)

65. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 635–644 (2017)
66. Zhao, J., Zhou, Z., Guan, Z., Zhao, W., Ning, W., Qiu, G., He, X.: Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2347–2357 (2019)

Authors and Affiliations

Jihu Wang¹ · Yuliang Shi^{1,2}  · Dong Li³ · Kun Zhang⁴ · Zhiyong Chen¹ · Hui Li¹

Jihu Wang
jihu_wang@mail.sdu.edu.cn

Dong Li
lidong@sd.sgcc.com.cn

Kun Zhang
kunzhangcs@126.com

Zhiyong Chen
chenzy@sdu.edu.cn

Hui Li
lih@sdu.edu.cn

¹ School of Software, Shandong University, Jinan, China

² Dareway Software Co., Ltd, Jinan, China

³ Information&Telecommunications Company, State Grid Shandong Electric Power Company, Jinan, China

⁴ School of Information Science and Engineering, University of Jinan, Jinan, China