Relation Modeling on Knowledge Graph for Interoperability in Recommender Systems

SeungJoo Lee*
Inha University
Incheon, Republic of Korea
hera2376@gmail.com

Seokho Ahn*
Inha University
Incheon, Republic of Korea
sokho0514@gmail.com

Young-Duk Seo†
Inha University
Incheon, Republic of Korea
mysid88@inha.ac.kr

ABSTRACT

Network technology has changed how people consume content through various channels, which constantly generate a large amount of data. To effectively utilize this data, many researchers have focused on interoperability, or the ability to use information from multiple systems together. However, in the field of recommender systems, few studies have considered interoperability. Existing methods for guaranteeing interoperability in recommender systems have limitations in their ability to model low-order relationships for data integration. In particular, there has been no study that ensures interoperability for knowledge graph-based recommender systems, which are suitable structures for integrating heterogeneous data. Therefore, we propose an integration method for multiple systems optimized for knowledge graph-based learning. This method can extend the knowledge graph through deep learning-based relation modeling of entities and ensure interoperability for the recommendation system. Our experimental results confirm that this method improves the performance of existing recommendation algorithms.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Knowledge representation and reasoning*; Neural networks;

KEYWORDS

Recommender systems, Knowledge graph, Interoperability, Knowledge representation

ACM Reference Format:
SeungJoo Lee, Seokho Ahn, and Young-Duk Seo. 2023. Relation Modeling on Knowledge Graph for Interoperability in Recommender Systems. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27-31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3555776.3577700

*These authors contributed equally to this work.
†Corresponding author

SAC '23, March 27-31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/10.1145/3555776.3577700

1 INTRODUCTION

Personalized recommendation services have become increasingly important in various fields, leading to a surge in research on recommendation systems. However, despite these efforts, these systems continue to face the challenges of data sparsity and cold start [11]. Many researchers have proposed using various auxiliary information for users and items to address these issues [29]. For instance, some studies have employed methods that utilize many types of feedback to learn and improve upon users' action sequence information [24]. Other studies have incorporated auxiliary information for the item in a knowledge graph (KG) to enrich the modeling of item features [3, 26]. Combining multiple systems in a single recommender system can also be beneficial in modeling more information about users and items [18]. Given that users' consumption patterns have shifted towards consuming content through various channels, it is crucial to seamlessly exchange the large amounts of data generated from these channels in one system. Therefore, it is necessary to investigate ways to ensure interoperability in the recommender system, which can ultimately enhance the integration of auxiliary information and the quality of both user and item features.

Existing studies have created ontologies to ensure interoperability in recommender systems and proposed methods using machine learning techniques to better predict user preferences with the generated ontology to improve recommendation performance [1, 13, 16, 23]. However, the existing ontology-building methods have limitations in modeling low-order relationships between heterogeneous data, as they only consider the class or property of the data to define entity abstraction without considering the relationships between data from multiple systems [14]. The output from these ontologies is also not in a suitable format for recommendation algorithms to effectively model heterogeneous data, requiring additional pre-processing steps to express it in a single algorithm. Despite these challenges, few works have achieved the perfect interoperability suitable for recommender systems.

The process of building an ontology involves a crucial step called entity abstraction, which defines the relationships between entities hierarchically by analyzing their properties. One way to achieve the purpose of an ontology is by using a KG structure, which is optimized for learning the relationships between entities

and can be easily modeled in a data structure that machines can understand [28]. In a KG, each relationship is represented by a triplelet consisting of a head entity, a relation, and a tail entity. This structure represents the specific relationship between the head and tail entities [20]. However, existing KG-based studies for interoperability are not optimized for use in recommender systems and are not effective at modeling the relationships between users and items [17]. Additionally, previous KG-based recommendation studies have only focused on heterogeneous data learning methods within a single recommender system without considering interoperability between multiple systems [27].

To address the limitations of the existing studies, we propose a KG-based method for modeling the potential relations between data from heterogeneous systems using deep learning rather than relying on manual efforts, such as ontology. Our method can continuously expand the KG by incorporating updated entities and relationships. As a result, we can enhance the recommendation performance with the enlarged KG by learning various auxiliary information to predict the more accurate user and item features.

The main contributions are summarized as follows:

- This is the first study to ensure interoperability of a KG-based recommender system through relational modeling between data from heterogeneous systems.
- Our method uses deep learning to model complex relationships between entities and combines them into a unified KG, which can solve the data sparsity problem of recommender systems.
- Our experimental results showed that our extended KG outperformed the baseline recommender algorithms.

The remainder of this paper is organized as follows. Section 2 discusses related work. In section 3, we describe our KG-based recommender system, which ensures interoperability and then explain case studies. In section 4, we discuss the experimental results. Finally, we present the conclusions and future work in Section 5.

2 RELATED WORKS

This section provides an overview of previous studies on ontology-based and KG-based recommender systems. Specifically, we analyze ontology-based systems in section 2.1 and KG-based systems in section 2.2.

2.1 Ontology-based recommender systems

Ontology is a tool used to facilitate interoperability and compatibility between systems by formally specifying shared concepts within metadata. In recommender systems, ontological representations of users and items help predict user preference more accurately [15]. Ontology-based recommender systems are systems that use ontologies, which are structures that represent concepts and their relationships, to make recommendations. These systems can be split into two categories: those that construct ontology [1, 5] and those that use existing ontology [4, 16].

First, Amini et al. [1] developed a reference ontology by combining different domain taxonomies and used this constructed ontology to understand scholars' knowledge. And de Araujo et al. [5] proposed a method for combining multiple ontologies using a hierarchical clustering method, aligning them based on their similarities. Most ontology construction methods have extended their knowledge using existing ontologies, but few studies have proposed ways of constructing new knowledge to make different systems work together.

Second, many researchers have used the domain ontology of recommender systems to infer users' preferences [15]. Daramola et al. [4] presented a knowledge-based tourism recommendation system that uses a tourism services ontology. Nilashi et al. [16] designed a recommendation system using an ontology and dimensionality reduction method to address the sparsity and scalability problems. In general, most existing studies have applied existing ontology to their approaches rather than presenting a new ontology or knowledge to achieve interoperability of recommender systems.

2.2 Knowledge graph-based recommender systems

KGs are a type of data structure that can show the connections between different entities and the reasons for these connections in a graph format. This makes them useful for learning data from diverse systems. Researchers have used KG to represent various supplementary information leading to the development of several KG-based learning methods. These methods can be divided into two main categories: embedding-based [20, 25, 26] and path-based [12, 27]. Embedding-based methods use mathematical representations of the entities and their connections to make recommendations, while path-based methods use the relationships between entities in the KG to make recommendations.

Some studies [20, 25, 26] have used embedding-based methods to learn the entity embeddings for items and improve the representation of user and item interactions. Wang et al. [25] used graph convolutional networks (GCN) to create entity embeddings and predict user preferences. Wang et al. [26] developed the KG attention network for recommendation (KGAT), which enhances entity embeddings by modeling higher-order relations between users and items. These two studies used embedding techniques to model a single system, but their KGs were still incomplete due to data sparsity, as they could not be extended to other systems. Sun et al. [20] created the multi-modal KG attention network for recommendation (MKGAT), which integrates information from multiple domains and obtains better representations for items through multi-modal KG propagation. However, this method only combines multi-domain representations at the embedding level, so it does not fully ensure interoperability between multiple systems.

Second, path-based methods [12, 27] represents user preference through multi-hop paths propagated in KGs. The knowledge-aware path recurrent network (KPRN) [27] extracts multi-hop paths, including data types of entity, entity types, and relations. Ma et al. [12] introduced a joint learning framework called RuleRec

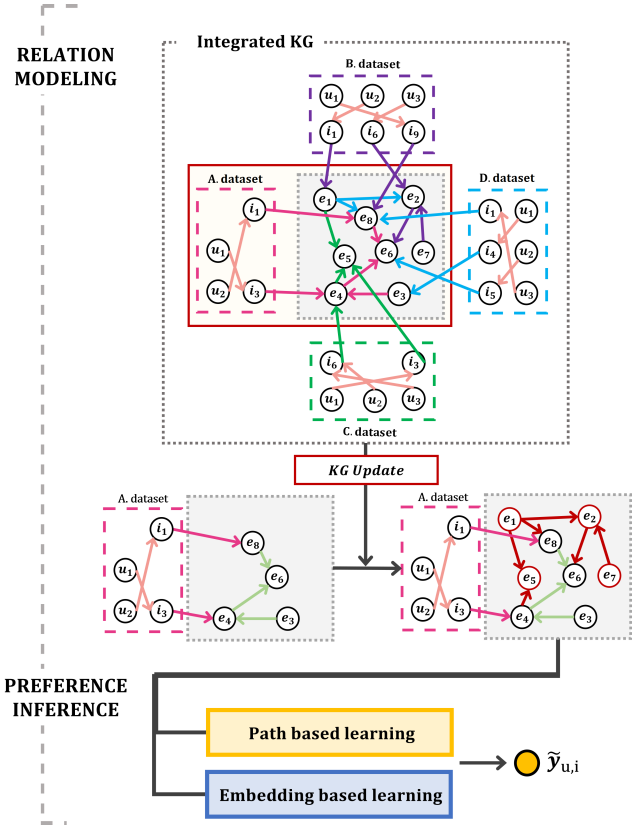

Figure 1: Overview of our work.

for an explainable rule-based neural recommendations. Prior research on path-based recommendation using KG focused on individual systems rather than considering interoperability between heterogeneous systems.

As mentioned above, existing KG-based methods to complete the KG for recommendation have a low level of relation learning between heterogeneous data. Additionally, there are few studies that achieve interoperability for recommender systems based on the KG. Therefore, we propose an integration approach for multiple systems optimized for KG-based learning. Our approach can extend the KG through high-order relation modeling and ensure the interoperability of the recommendation system.

3 PROPOSED APPROACH

In this section, we first describe the notation and background of KG to introduce the proposed approach (Section 3.1). We then explain a proposed relation modeling approach to extending the KG (Section 3.2). Next, we introduce various real-world scenarios for KG expansion from simple case to interoperable usages (Section 3.3). Finally, we describe how we apply an extended KG to a recommendation system (Section 3.4). As a result, our methodology can contribute significantly to the preference inference of any KG-based recommendation systems (i.e., embedding-based and path-based recommender systems). The overall process of our

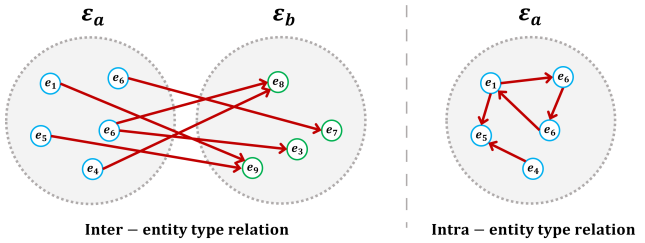

Figure 2: Relation modeling based on entity type.

approach is shown in Figure 1. It involves expanding the KG through relationship modeling and then performing preference inference on the expanded KG.

3.1 Knowledge Graph in Recommender System

A KG is a directed graph in which edges between connected nodes are semantically well-defined for a given knowledge base [6]. Formally, let $\mathcal{E} = \{e_1, e_2, \cdots, e_k\}$ be a set of entities (i.e., nodes) and $\mathcal{R} = \{r_1, r_2, \cdots, r_g\}$ be a set of relations (i.e., edges) where k and g are the number of entities and relations, respectively. Then a $\mathcal{KG}$ is a subset of the cartesian product $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$ defined as follows:

$$\mathcal{KG} = \{(h, r, t) \,|h, t \in \mathcal{E}, r \in \mathcal{R}\} \tag{1}$$

where h, r, and t denotes head, relation, and tail, respectively. The triplet (h, r, t) implies that there is a relationship r from entity h to entity t.

In the recommender system, an entity is usually an object of a knowledge base (e.g., director, starring, and category in the movie knowledge base), but users $\mathcal{U}$ and items $\mathcal{I}$ can also be set as entities. For example, we can generate a triplet $(u, r_{interact}, i) \in \mathcal{KG}$ if the user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ have explicit or implicit feedbacks where $r_{interact}$ denotes a newly defined relation [22, 27]. Then the integrated $\overline{\mathcal{KG}} = \{(h, r, t) \,|h, t \in \bar{\mathcal{E}}, r \in \bar{\mathcal{R}}\}$ can better reflect user and item preferences by providing additional information, where we set $\bar{\mathcal{E}} := \mathcal{E} \cup \mathcal{U}$ and $\bar{\mathcal{R}} := \mathcal{R} \cup \{r_{interact}\}$.

Moreover, in a KG, each entity may have a different entity type. For example, two entities, *Pop* and *Zazz*, in a music knowledge base might be considered the *Category* entity type. Defining entity type of entities provides useful information in a KG [7], and can also be used for user embedding [27]. We define a set of entity types $\mathcal{A}$ and a function ϕ that maps entities to their corresponding entity types ($\phi : \mathcal{E} \rightarrow \mathcal{A}$) to properly relate entities in the KG.

3.2 Relation Modeling for Knowledge Graph

We present a method for expanding the KG by adding new relations between both existing and new entities in this section. This method can improve the KG and enhance the connectivity between entities. To define a suitable relation, we consider a set of all entities whose entity type is $a \in \mathcal{A}$, denoted $\mathcal{E}_a = \{e | e \in \mathcal{E}, \phi(e) = a\}$. We can then consider relations between different entity types or within the same entity type. Therefore, we define new relations in two ways:

inter-entity type and intra-entity type relations as shown in Figure 2.

3.2.1 Inter-entity type relation. An inter-entity type relation connects different types of entity types. If two sets $\mathcal{E}_a$ and $\mathcal{E}_b$ can be modeled based on a relation r_{ab}, we can generate triplets which are of the form (e, r_{ab}, f) for some $e \in \mathcal{E}_a$ and $f \in \mathcal{E}_b$. For example, if we set $\mathcal{E}_a = \mathcal{U}$ and $\mathcal{E}_b = \mathcal{I}$, the relation between the two groups (i.e., user group $\mathcal{U}$ and item group $\mathcal{I}$) may be defined based on the rating given by some user $u \in \mathcal{U}$ to item $i \in \mathcal{I}$. We can then generate two triplets (u, r_{like}, i) and $(i, r_{positivelyRatedBy}, u)$ in this example. Moreover, some collaborative filtering (CF) methods can be applied if not rated; hence more relations can be connected than the rated one.

3.2.2 Intra-entity type relation. An intra-entity type relation is a connection within the same entity type. If a set $\mathcal{E}_a$ can be modeled using a relation r_a, we can create triplets of the form (e, r_a, f) for some $e, f \in \mathcal{E}_a$. For example, if we set $\mathcal{E}_a = \mathcal{I}$, the relation between the two items $i_1, i_2 \in \mathcal{I}$ can be defined based on the item similarity of i_1 and i_2. This would result in the two triplets $(i_1, r_{isSimilarToItem}, i_2)$ and $(i_2, r_{isSimilarToItem}, i_1)$. As another example, the objects in a knowledge base can also be defined using some relations. The categories of items in a movie knowledge base, such as romance, fantasy, and horror can be calculated based on the similarity of the items in each category. Furthermore, relations can also be added when combining entities of the same type from different datasets for data interoperability.

3.3 Case Studies

In this section, we introduce three real-world scenarios for expanding a KG through relation modeling. The first scenario involves extending the KG within a single system (Section 3.3.1). The second scenario involves adding a relation among entities in a knowledge base so that it can add relations to new entities (Section 3.3.2). Finally, we will consider a scenario where we model the relation between data from different systems as a way to achieve interoperability (Section 3.3.3).

3.3.1 Relation modeling based on collaborative filtering. Recent studies have demonstrated the efficacy of incorporating user-item feedback into the KG for enhanced preference inference [22, 27]. However, these efforts alone are insufficient for effectively expanding the KG due to the sparsity of available data. To address this issue, we propose the incorporation of unrated user-item feedback through the use of CF techniques. Additionally, we introduce new relationships based on the similarity between users and between items, as determined through the application of CF embeddings. The overall process is depicted in Figure 3, with further details provided in the following explanations.

- *Relation modeling through rating prediction:* We consider two pre-defined inter-entity relations in our model, such as r_{like} and $r_{positivelyRatedBy}$. These relations apply to a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$. If a user $u \in \mathcal{U}$ gives an item $i \in \mathcal{I}$ a rating R that is greater than a threshold θ_R, we add two triplets to the $\overline{\mathcal{KG}}$: (u, r_{like}, i) and $(i, r_{positivelyRatedBy}, u)$. If the item has not been rated by the user, these two triplets are added to the $\overline{\mathcal{KG}}$ only when the predicted rating $\hat{R} = \mathcal{CF}(u, i)$ calculated by the recommender model $\mathcal{CF}(*)$ is greater than the $\hat{\theta}_R$. Note that the predicted rating $\hat{R}$ is less accurate than actual rating R, so we need to be more careful about adding predicted ratings $\hat{R}$ to the $\overline{\mathcal{KG}}$ (e.g., by setting $\hat{\theta}_R > \theta_R$).
- *Relation modeling based on similarity:* We define two new intra-entity relations for users $\mathcal{U}$ and items $\mathcal{I}$, called $r_{isSimilarToUser}$ and $r_{isSimilarToItem}$, respectively. Using the recommender model $\mathcal{CF}(*)$, each user and item have an embedding. We then calculate the similarity between users, $Sim(u_1, u_2)$, or between items, $Sim(i_1, i_2)$, for all pairs of users $(u_1, u_2 \in \mathcal{U})$ and all pairs of items $(i_1, i_2 \in \mathcal{I})$. If $Sim(u_1, u_2)$ is greater than threshold θ_u, we add the two triplets $(u_1, r_{isSimilarToUser}, u_2)$ and $(u_2, r_{isSimilarToUser}, u_1)$ to $\overline{\mathcal{KG}}$. Similarly, if $Sim(i_1, i_2)$ is greater than a threshold θ_i, we add the triplet $(i_1, r_{isSimilarToItem}, i_2)$ and vice versa.

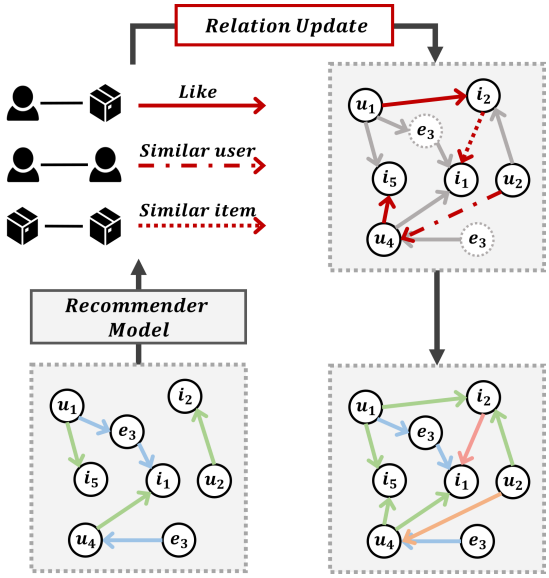

Figure 3: Relation modeling using collaborative filtering.

3.3.2 Relation modeling for extending knowledge graph. Our relational modeling approach not only considers user entities and item entities, but also objects within a knowledge base. These entities can incorporate relational information from online databases such as IMDB, DBpedia, and Wikipedia. Additionally, our approach allows for the incorporation of relational information within KGs, even when new entities are added. The processes involved in this are illustrated in Figure 4 and are further described as follow:

- *Relation modeling among objects in a knowledge base:* We present simple examples to demonstrate that the KG can

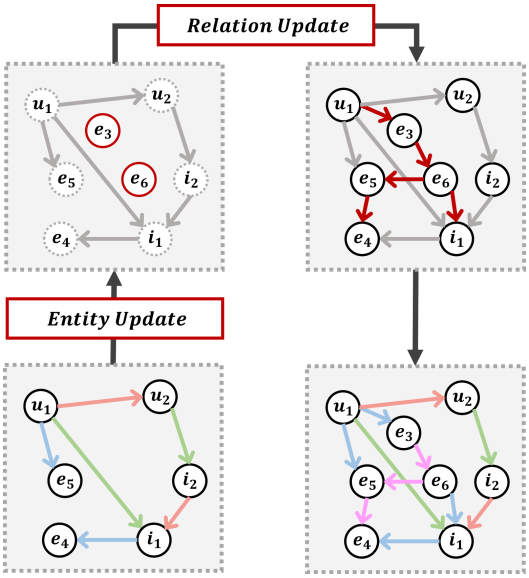

Figure 4: Updated entity and relation in Knowledge Graph.

add new relations on its own. Suppose we are attempting relational modeling of *Category* entity types in a movie knowledge base. We consider two types of relations (i.e., inter-entity and intra-entity relations) that are connected to $\mathcal{E}_{Category}$. For example, in terms of inter-entity relations, we can define two relations between $\mathcal{E}_{Category}$ and $\mathcal{U}$, namely r_{prefer} and $r_{isPreferedBy}$, respectively. In this case, we can connect the user's preferred categories by clustering items rated highly by the user. If it is expected that the user u prefers category $c \in \mathcal{E}_{Category}$, we add two triplets (u, r_{prefer}, c) and $(c, r_{isPreferedBy}, u)$ to the KG. As another example of relational modeling in terms of intra-entity relations, we can define a relation $r_{isSimilarToCategory}$ by calculating the similarity between clustered items. Then we add two triplets $(c_1, r_{isSimilarToCategory}, c_2)$ and $(c_2, r_{isSimilarToCategory}, c_1)$ to the KG when the similarity is greater than a threshold. It is worth noting that these relations are not defined in online databases (e.g., IMDB, DBpedia, Wikipedia), so we can provide more diverse relations as a result.

- *Relation modeling for new entity:* Our approach ensures that new entities can also be modeled as long as their entity type is determined. For example, suppose a new entity e_T is added at time T that did not exist until time $T-1$. We can update the set $\mathcal{E}_a$ to $\overline{\mathcal{E}_a} := \mathcal{E}_a \cup \{e_T\}$ if the entity type of entity e_T is $a \in \mathcal{A}$. We then consider pre-defined two relations (i.e., inter-entity and intra-entity type relations) that are connected to the set $\overline{\mathcal{E}_a}$ and connect relations using our approach mentioned above. This means that our approach allows the KG to expand dynamically, so it can be increased in real-time.

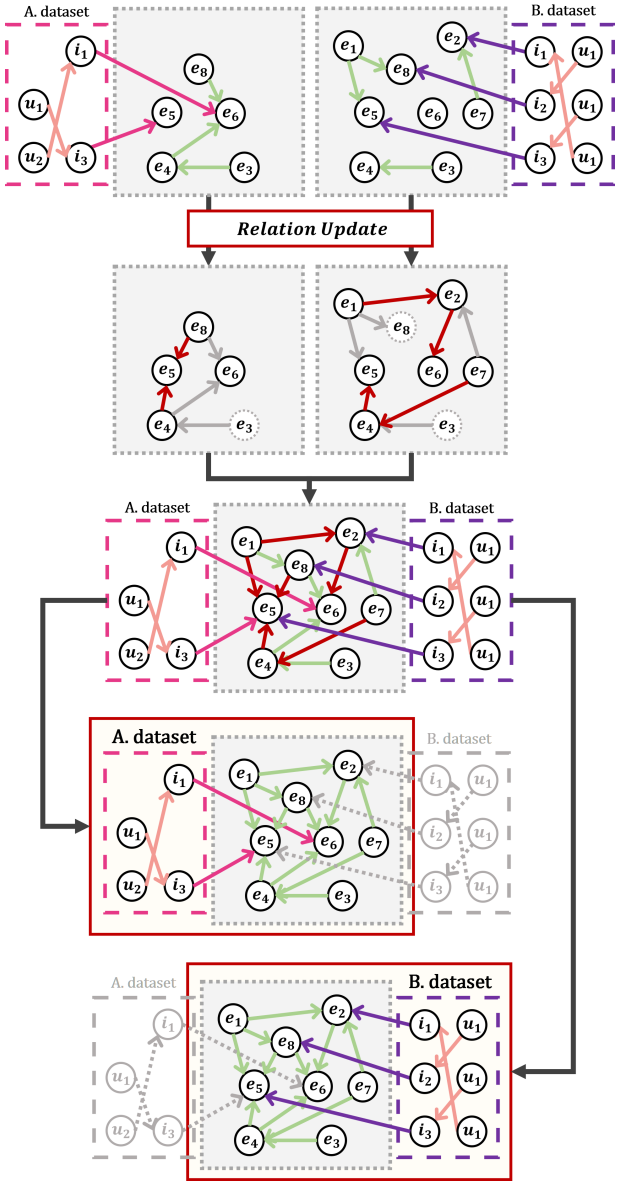

Figure 5: A knowledge graph that achieves interoperability.

3.3.3 Extending knowledge graph based on multiple systems for ensuring interoperability. In this example, we demonstrate how to make multiple systems interoperable by extending their KG. In the previous two cases (Section 3.3.1 and 3.3.2), we discussed how to extend the KG from two perspectives. Specifically, the first example involves using CF models to add relations between user and item entities (Section 3.3.1). The second example shows that objects in a knowledge base can be inter-connected by various models, and it is possible to add new entities (Section 3.3.2). When both of these conditions are satisfied, it becomes possible to expand the KG across multiple systems. The steps for doing this are shown in Figure 5 and will be explained in more detail below.

In practice, two or more heterogeneous systems must be considered for interoperability. But for simplicity, we only consider the two user-item pair datasets (i.e., two heterogeneous systems) which are in a similar knowledge base, shown in Figure 5. Each system will have a different KG depending on the users and items in the system. However, we can make relation modeling from each system in a similar way to the previous scenarios. If we combine these two KGs, each system can use new relations that cannot be induced from its system. Then, recommendation performance can be improved since each system uses an extended KG reflecting heterogeneous information. Furthermore, this process enables the KG expansion that satisfies interoperability across multiple heterogeneous systems.

3.4 Preference Inference

The expended KG generated from Sections 3.2 and 3.3 can be employed in any recommender system that uses a KG, such as embedding-based and path-based recommender systems. However, it is worth noting that adding these various relations does not pose a significant problem for embedding-based recommender systems because they are embedded in a fixed embedding space. On the other hand, path-based recommender systems may be impacted by the addition of these relations, as it leads to a rapid increase in the number of paths, as noted by Wang et al. [27]. Furthermore, there is currently no general methodology for extracting paths in path-based recommender systems, as these tasks are typically labor-intensive and passive. To make our method more widely applicable to all knowledge-based recommender systems, we present a consistent method for path extraction to improve usability. Our method is described in Algorithm 1, which returns a sample of high-quality paths $\mathcal{P}(u, i)$ for a given start user entity u and end item entity i of the $\mathcal{KG}$.

Our method is based on the depth-first search (DFS) algorithm, which generates all paths from a given start to an end point in the graph. However, extracting all paths from the KG can result in millions of paths, so the paths must be sampled. Moreover, Sun et al. [21] found that paths that are too long are of low quality, so it is important to select a path with a sufficiently short length. To solve this problem, we use weighted random sampling to select the relation and tail (Line 22-23). Weighted sampling of the relations help to reduce the imbalance in the number of tails, while sampling the tails speeds up execution time and reduces the number of samples. The *dropout_threshold* argument (Line 25) can also be used to sample a smaller number of routes. The variable L sets the maximum length of the path, which is small enough to produce a short-length with high-quality path and adjusts the depth bound of the recursive function.

As a result, both embedding-based and path-based methods can be derived from extended KGs, which can incorporate a greater range of diverse knowledge in recommender systems.

Algorithm 1: Path Extraction

Input :
- $\mathcal{KG}$ Knowledge graph
- u User id which is the first entity in the path
- i Item id which is the last entity in the path
- L Maximum path length

Output:
- $\mathcal{P}(u, i)$ Sampled paths between entity pair (u, i)

```
1  Function GeneratePath(KG, u, i, L):
2      visited ⟵ {False}
3      P(u, i) ⟵ list ()
4      foreach tail t in (u, givenGoodRatingTo, t) ∈ KG do
5          path ⟵ list(u, t)
6          _GeneratePathRecursive(KG, t, i, path, L − 1)
7      end
8      return P(u, i)

9  Procedure _GeneratePathRecursive(KG, u, i, path, L):
10     if visited [u] = True then
11         return
12     end
13     visited [u] ⟵ True
14     if u = i then
15         path.append(u)
16         P(u, i).append (path)
17         return
18     end
19     if L = 1 then
20         return
21     end
22     relations ⟵
         u.relations().weightedRandomSampling()
23     tails ⟵ relations.tails().randomSampling()
24     foreach tail t in (u, sampleRelation, t) ∈ KG do
25         if random[0, 1] < dropoutThreshold then
26             return
27         end
28         if t in tails and not visited [t] then
29             path.append(t)
30             _GeneratePathRecursive(KG, t, i, path, L − 1)
31         end
32     end
33     return
```

4 EXPERIMENTS

This section presents the results of an empirical experiment investigating the effect of relation modeling on recommendation performance. The process of setting up the experimental environment, including the datasets and preprocessing methods, is described in Section 4.1. The experimental results and their discussion can be found in Section 4.2.

Table 1: Details of our knowledge graphs

Knowledge graph	$\mathcal{KG}_{\text{base}}$	$\mathcal{KG}_{\text{extend}}$
#Users	6,040	6,040
#Items	3,706	3,706
#Entities on knowledge base	59,275	59,275
#Total entities	**69,021**	**69,021**
#Entity types	7	7
#Relation types	16	18
#User-item interactions	1,000,210	1,857,359
#User-user interactions	0	38,498
#Item-item interactions	0	30,327
#Triplets	**1,839,342**	**2,696,591**
#Positive paths	10,697,880	11,153,006
#Negative paths	7,599,676	8,085,492
#Total paths	**18,297,556**	**19,238,468**
#Average paths of user-item pair	9.3	11.7
Length of average paths	5	4.9

4.1 Experimental Settings

An experiment was conducted based on the first scenario, which is the simplest case. Our experimental goal is to determine how much the extended KG affects recommendation performance. We expect that good performance will be obtained even in more complex examples if our experiment is verified.

The structure of this subsection is as follows: First, we look at the dataset and the process of constructing KGs used in the experiment (Section 4.1.1). We then introduce the evaluation metrics used in the experiment (Section 4.1.2). The implementation details are shown in Section 4.1.3.

4.1.1 Knowledge graph construction. In order to couduct our experiment, we first preprocessed datasets, modeled relations between entities, and extracted paths. We used MovieLens-1m[1] and DBpedia[2] datasets to generate a KG, which we then divided into two versions based on the first scenario: one using only basic triplets (called $\mathcal{KG}_{\text{base}}$) and the other using relation modeling (called $\mathcal{KG}_{\text{extend}}$).

To model the relations using CF algorithm, we employed Neural CF (NCF) [9] to predict ratings for each user-item pair. We set $\theta_R, \hat{\theta}_R, \theta_u$, and θ_i to 4, 5, 0.6, and 0.6, respectively, in order to generate triplets. We then used Algorithm 1 to extract an average of ten paths per user-item pair from each KG, with the paths in $\mathcal{KG}_{\text{extend}}$ containing most of those in $\mathcal{KG}_{\text{base}}$. Table 1 summarizes the key details of our KG.

4.1.2 Evaluation metrics. We evaluated the model performance of different KG based on Precision and Recall [19], which is widely used for evaluation metrics. More specifically, Precision@K denotes the ratio of the correct recommended items among the top-K item recommendation results. Recall@K is the ratio of the correct recommended items to the user's preferred item list. K was set to 5, 10 and 20 in our experiments. The performance was calculated by the average of Precision@K and Recall@K for all test users and items.

Table 2: Comparison results of Precision@K

	Precision@5	Precision@10	Precision@20
KPRN-$\mathcal{KG}_{\text{base}}$	0.0247	0.0263	0.0290
KPRN-$\mathcal{KG}_{\text{extend}}$	**0.0258**	**0.0270**	**0.0298**

Table 3: Comparison results of Recall@K

	Recall@5	Recall@10	Recall@20
KPRN-$\mathcal{KG}_{\text{base}}$	0.1527	0.2045	0.2814
KPRN-$\mathcal{KG}_{\text{extend}}$	**0.1536**	**0.2088**	**0.2892**

4.1.3 Implementation details. We trained them on the a machine with AMD Ryzen 5 5600X 6-Core and NVIDIA GeForce RTX 2060, and tested the accuracy of ranking. We compare the performance of two KGs (i.e., $\mathcal{KG}_{\text{base}}$ and $\mathcal{KG}_{\text{extend}}$) using knowledge-aware path recurrent network (KPRN) [27] which is a state-of-the-art path-based model. We denote these two models as KPRN-$\mathcal{KG}_{\text{base}}$ and KPRN-$\mathcal{KG}_{\text{extend}}$. The additional details we used are as follows:

- We implemented and tested all the models in Pytorch[3].
- The ratio of train data to test data was set to 8:2.
- Binary cross-entropy (BCE) loss [2] was used as a loss function.
- Adam [10] was used as an optimizer.
- The sigmoid [8] activation function was used as a pooling function. The final score of the paths of the user-item pair was calculated as an average value.
- *Learning rate*, *batch size*, *number of epochs*, and *hidden size* of KPRN was set to 0.02, 256, 15, and 16, respectively.

4.2 Experimental Results

Tables 2 and 3 present a comparison of the performance of KPRN-$\mathcal{KG}_{\text{base}}$ and KPRN-$\mathcal{KG}_{\text{extend}}$. Our investigation shows that the extended KG, which was created using relation modeling, exhibits significantly higher recommendation accuracy than the original KG. As shown in Tables 2 and 3, on average, KPRN-$\mathcal{KG}_{\text{extend}}$ achieves a 3.3% and 1.82% improvement in precision and recall, respectively, compared to KPRN-$\mathcal{KG}_{\text{base}}$. This demonstrates that the expansion of the KG, which integrates new entities and relations, generates more reliable paths for inferring user preference and improves the recommendation accuracy.

Our experimental results indicate that an extended KG created from multiple systems outperforms a KG created from a single system. By using the method of interoperably integrating

[1] http://grouplens.org/datasets/movielens
[2] https://dbpedia.org/ontology/Film
[3] https://pytorch.org/

heterogeneous systems into a KG, we can build an extended KG that incorporates a large amount of auxiliary information from various sources. This informative KG generates reliable paths for expressing user-item preferences and enhances entity embeddings learned from a large amount of auxiliary information, allowing us to model high-order relations between users and items more effectively.

5 CONCLUSION AND FUTURE WORK

In this study, we propose a method for integrating multiple systems into a KG using two types of relation modeling. Our method can easily distinguish whether an entity from one system can be related to entities from other systems, and it can gradually expand the KG. We present three real-world scenarios for KG expansion through relation modeling and demonstrate the effectiveness of our method through experimental results, which show that it outperforms existing recommender systems based on KG without expansion. As a result, our method enables interoperability of the recommendation system based on expanded KG.

In the future, we plan to build a model structure that can integrate multiple systems in a single KG. The model can define whether entities in one system can be integrated with other systems in the KG. In addition, we expect to develop an algorithm that can interoperably integrate multiple systems from other domains.

ACKNOWLEDGEMENTS

This work was partly supported the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1C1C1012408), Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No.RS-2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development (Inha University), and No.2022-0-00448, Deep Total Recall: Continual Learning for Human-Like Recall of Artificial Neural Networks), and the INHA UNIVERSITY Research Grant.

REFERENCES

[1] Bahram Amini, Roliana Ibrahim, Mohd Shahizan Othman, and Mohammad Ali Nematbakhsh. 2015. A reference ontology for profiling scholar's background knowledge in recommender systems. *Expert Systems with Applications* 42, 2 (2015), 913–928.

[2] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[3] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Jointly non-sampling learning for knowledge graph enhanced recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 189–198.

[4] Olawande Daramola, Mathew Adigun, and Charles Ayo. 2009. Building an ontology-based framework for tourism recommendation services. In *ENTER*. 135–147.

[5] Fabiana Freire de Araujo, Fernanda Lígia R Lopes, and Bernadette Farias Lóscio. 2010. MeMO: A clustering-based approach for merging multiple ontologies. In *2010 Workshops on database and expert systems applications*. IEEE, 176–180.

[6] Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a Definition of Knowledge Graphs.

[7] Xiou Ge, Yun-Cheng Wang, Bin Wang, and C.C. Jay Kuo. 2022. CORE: A knowledge graph entity type prediction method via complex space regression and embedding. *Pattern Recognition Letters* 157 (may 2022), 97–103. https://doi.org/10.1016/j.patrec.2022.03.024

[8] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA, USA. http://www.deeplearningbook.org.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

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

[11] R Kiran, Pradeep Kumar, and Bharat Bhasker. 2020. DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications* 144 (2020), 113054.

[12] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The world wide web conference*. 1210–1221.

[13] Carmen Martinez-Cruz, Carlos Porcel, Juan Bernabé-Moreno, and Enrique Herrera-Viedma. 2015. A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling. *Information Sciences* 311 (2015), 102–118.

[14] Sabino Metta, Paolo Casagranda, Alberto Messina, Maurizio Montagnuolo, and Francesco Russo. 2016. Leveraging MPEG-21 user description for interoperable recommender systems. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. 1072–1074.

[15] Stuart E Middleton, David C De Roure, and Nigel R Shadbolt. 2001. Capturing knowledge of user preferences: ontologies in recommender systems. In *Proceedings of the 1st international conference on Knowledge capture*. 100–107.

[16] Mehrbakhsh Nilashi, Othman Ibrahim, and Karamollah Bagherifard. 2018. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications* 92 (2018), 507–520.

[17] Daniela Oliveira and Mathieu d'Aquin. 2022. Extracting data models from background knowledge graphs. *Knowledge-Based Systems* 237 (2022), 107818.

[18] N Shilov. 2020. Recommender system for navigation safety: Requirements and methodology. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation* 14, 2 (2020).

[19] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*. 213–220.

[20] Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. 2020. Multi-modal knowledge graphs for recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1405–1414.

[21] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proc. VLDB Endow.* 4, 11 (aug 2011), 992–1003. https://doi.org/10.14778/3402707.3402736

[22] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent Knowledge Graph Embedding for Effective Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 297–305. https://doi.org/10.1145/3240323.3240361

[23] John K Tarus, Zhendong Niu, and Abdallah Yousif. 2017. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems* 72 (2017), 37–48.

[24] Quyen Tran, Lam Tran, Linh Chu Hai, Ngo Van Linh, and Khoat Than. 2022. From implicit to explicit feedback: A deep neural network for modeling sequential behaviours and long-short term preferences of online users. *Neurocomputing* 479 (2022), 89–105.

[25] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*. 3307–3313.

[26] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[27] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5329–5336.

[28] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1437–1445.

[29] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 239–248.