KGE-RS: Enhancing Recommendation Systems with Knowledge Graph Embedding Strategies

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: July 24th, 2023

BAI Changhao

(5121FG46-1)

Advisor: Prof. Hayato Yamana

Research guidance: Research on Bioinformatics

Abstract

Recommendation Systems (**RS**) play an important role in helping users find the items that they may choose, and RSs have been widely used in many areas, including news, music, restaurant, and movie. However, since the RSs suffer from the problems of cold-starting and data sparsity, much side information has been adopted into RS, such as Knowledge Graph (KG). In recent days Knowledge Graph-based RSs usually start with a certain Knowledge Graph Embedding (**KGE**) layer to help represent the KG from the words in triples into the form of vectors or matrices, and then adopt these vectors or matrices to execute the later processing by enriching the attributes of users and items. However, the existing KGE layers in the original RSs have many problems, including lacking the ability to represent certain types of relations in KG, and ignoring the rich inference patterns contained in the multi-step relation paths. To solve these listed problems in the existing KG-based RS, we adopt three new KGE layers that are specifically designed to address those problems. The overall idea is that a KGE layer with higher accuracy performance can bring the whole system better recommendation results. To the best of our knowledge, this is the first research specifically targeting the KGE layers in RS. To confirm our assumption, we conducted two experiments on two RS models, Deep Knowledge-aware Network (DKN) and Knowledge Graph Attention Network (KGAT) with seven different KGE methods, trying to improve their performance. Compared to the baseline, the proposed DKN model increases by 4.3% and 4.6% on ACC and AUC, and the proposed KGAT model increases by 1.0% and 1.2% on recall@20 and ndcg@20 on three datasets.

CONTENTS

1. Introduction

Since the Internet brings overwhelming information to users, it becomes harder and harder to make the right decision when dealing with multiple choices and big data environments. Thus, building a precise Recommender System (**RS**) is always an important task. RSs have been widely used in many real-life areas. For example, movies, music, news, and book recommendation systems help users make decisions. However, data sparsity and cold start problems prevented the further performance improvement of RS. In recent years, minding multiple side information as a supplement is a potential method to solve the data sparsity and cold-start problem. Among all the side information, the usage of the Knowledge Graph (**KG**) in RS is a hot topic.

KG describes the real-world relations connected between entities. KG stores every piece of information in the form of triples: (head entity, relation, tail entity), and then is formed as a graph. Based on the data structure, many large real-world knowledge bases have been established, such as Freebase [1], DBpedia [2], Yago [3], and Nell [4]. With a KG as supporting side information, users, items, and their attributes can be mapped into the KG, making the items' mutual relations and the user preference can be better captured and understood in the process of user/item representation. Then the learned user/item representation will be used in the further training of a certain recommendation algorithm, forming a KG-based RS.

Since the CKE [5] model is first proposed in 2016, the KG-based RS has become a hot topic in the RS area. CKE adds a Knowledge Graph Embedding (**KGE**) layer to the user-item representation process. KGE aims to represent triples forms (entities, relations, entities) that existed in KGs. By embedding the semantic information of entities and relations into the form of vectors, semantic similarity can be calculated through the distance of those vectors. Then the preliminary representation of entities will be adopted into the later model training process of RS to improve accuracy.

More recently, many new KG-based RS are proposed, such as DKN [6] in 2018, KGAT [7] in 2019, MKR [17] in 2019, and KGCL [8] in 2022. A common characteristic of those models is that they start from a KGE layer (e.g., TransE [9] layer in KGCL, TransH [10] layer in DKN, TransR [11] layer in KGAT, TransD [12] layer in DKN) in their system structure to get a preliminary embedding of entities in KG. After that, they adopt the embeddings into machine learning models, such as Convolutional Neural Network (CNN), Graph Neural Network (GNN), and Attention Mechanism (AM), to calculate the further embedding of users and items and complete recommendation task.

However, adopting traditional KGE methods [9-12] causes the loss of representation ability, compared with state-of-the-art techniques that are used in the later user/item representation learning. Many researches [13-15] have pointed out the drawbacks of those traditional KGE methods. According to the research result of RotatE [13], TransE lacks the ability to represent symmetry relations. TransH, TransR, and TransD cannot represent the inversion and composition relations, affecting the embedding results of triples in KG. The experiment of PTransE [14] confirmed the importance of the rich inference patterns contained in the multi-step relation paths, which are ignored in the traditional KGE methods.

Our intuition is that a KGE layer with higher accuracy performance can bring the whole system better recommendation accuracy. Therefore, the goal of this research is to build an RS enhanced with the KG as side information, where the KGE method with higher representation accuracy is used, to improve the accuracy of the whole system. Specifically, we use a classic news recommendation model DKN and a general recommendation model KGAT as the baselines, which both adopt a KGE layer as their first input layer in the whole model structure. To test the influence of KGE layers, we try seven different KGE methods [9-14] mentioned above and PRRL [15] KGE methods, to improve the recommendation results of the original model.

The main contributions of this thesis can be summarized as follows:

(1) we conduct extensive experiments on the improved RS model, which confirm the effectiveness of the KGE layer in the RS models, that a KGE layer with higher accuracy performance can bring the whole system better recommendation results.

(2) we propose a news recommendation model based on our experiment, which adopt three different KGE methods [13-15], making this news RS can approach all the relations patterns in the KGE phase, and also add the multi-step relation paths in the KGE layer.

The remainder of this thesis is organized as follows: Section 2 explains the related work, including different KGE techniques and their two existing problems, collaborative filtering (CF) RS and other RSs. Section 3 details the proposed method, with different KGE methods. Section 4 presents the evaluation of the proposed method and discussions. Finally, Section 5 is the conclusion and future work for this thesis.

2.Related work

In this section, we first introduce the fundamental idea of KGE and three types of KGE methods. Then we introduce the basic steps of collaborative filtering RS, the classic news RS model DKN and the general RS model KGAT in detail.

2.1 Knowledge Graph Embedding Methods

KGE is a hot topic in the research area of KG, aiming to represent entities and relations into the form of vectors/matrices or other forms while keeping the semantic information from the original KG structure. KGE methods can solve the problem that real-world KGs are often incomplete by predicting the missing links in the KG, therefore a recommendation task can be treated as a downstream task of KGE.

The earliest KGE techniques are called **translational distance models**, such as TransE [9], TransH [10], TransR [11], TransD [12], STransE [20], and TranSparse [21]. The general idea of this kind of model is to map the entities and relations into a continuous vector space. By calculating the translational distance based on their well-designed score function and differentiating the score between positive triples and negative triples, the embeddings of entities and relations can be achieved.

The TransE model represents both entities and relations as vectors in vector space. It is simple and efficient in the process of KGE and easy to be adopted in the downstream tasks, but it suffers from the problem of representing the N-to-1 and N-to-N relations. The TransH model solves this problem by embedding relations as vectors on the hyperplane and projects entities into this relation-specific hyperplane, yet the relations and entities are still encoded in the same space, thus the variety of aspects of entities and relations are ignored. The TransR model adopts relations-specific space for the mapping of entities with a projection matrix and fixes the problem of TransH. The STransE model also solves the problem of TransH by adopting two relation matrices to the head entity and the tail entity in each triple. The TransD model is designed to simplify the idea of the TransR model and the STransE model by changing the projection matrix with two vectors and their product. The TranSparse model also reduces the time consumption of TransR and STransE with a sparseness degree to reduce the parameter numbers in the projection matrices.

Figures 1 and 2 adapted from [16] give a simple illustration of the idea of the above six translational distance models [9-13]. In these two figures, the vector of h, t, r represents the head entity, the tail entity, and the relation in a triple (h, r, t). The vector of h_1 and t_1 represents the projected head/tail entity vectors. M_r in Figure 2 is the projection matrix of relation r designed to map the entities into relation space, noticing that the TransR model, the STransE model, the TranSparse model, and the TransD model share the same projection approach. The score functions are calculated by the distance between the tail entity t and the result of head entity h translated by the relation r. Table 1 shows the mapping space of entity/relation and the pros and cons of these four translational distance models.

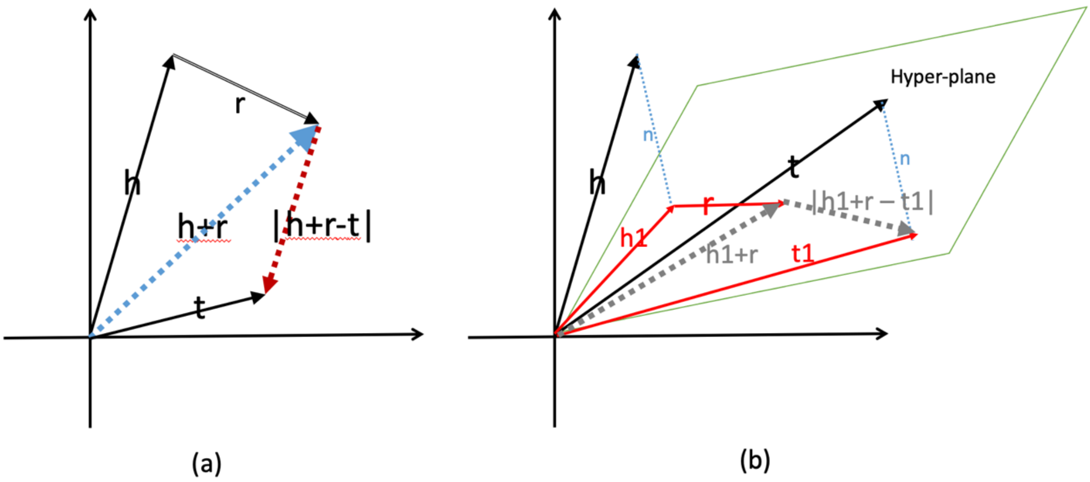

Figure 1. Simple illustrations of TransE (a) and TransH (b) based on Figure 1 in [16]

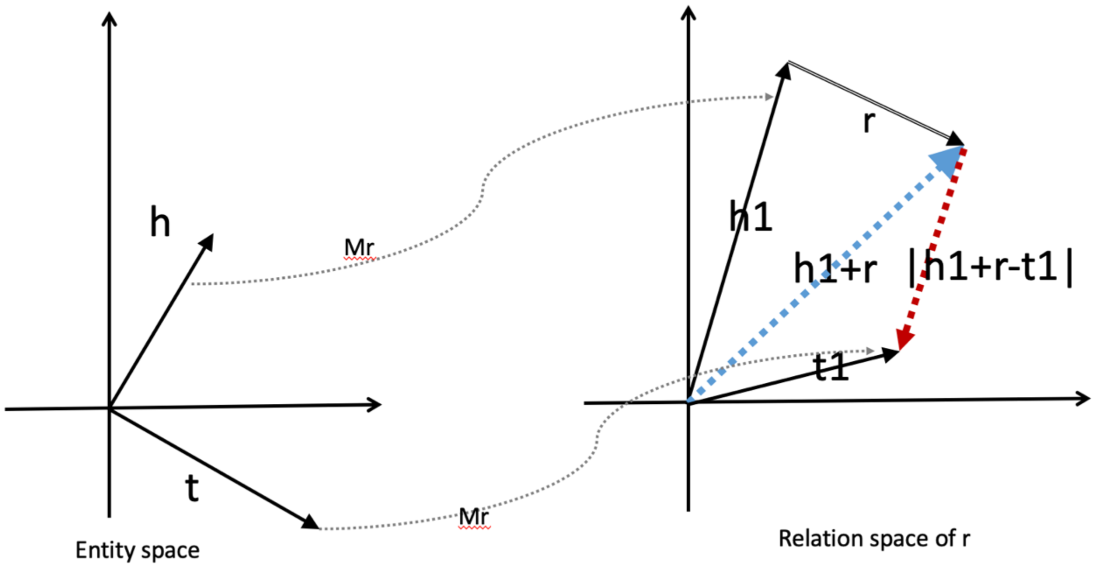

Figure 2. Simple illustrations of TransR, STransE, TranSparse, and TransD, based on Figure 1 in [16]

Table 1. Comparison of TransE, TransH, TransR, STransE, TranSparse, and TransD

	Projection space	Pros	Cons
TransE [9]	vector space	Efficient in model building and triple representation with a small number of parameters	Ignored N-to-N relation representation
TransH [10]	Relation-specific hyperplane	Solved the N-to-N relation representation problem	Ignored entity/relation aspects
TransR [11]	Relation-specific space	Solved the triple aspects representation problem	Large time consuming in the model building
STransE [20]	Relation-specific space	Solved the triple aspects representation problem	Data sparsity problem, time consuming in model building
TransD [12]	Relation-specific space	Reduced the parameter numbers	Ignored unobserved facts
TranSparse [21]	Relation-specific space	Dealt with entity/relation unbalance problem	Only concerned sparsity patterns

The **rotation model** is designed to solve the problem that the translational distance models cannot capture all the relation patterns. Rotation models represent the entities and relations with the modulus information and the angle information in the complex plane or polar coordinates plane. The RotatE [13] model is inspired by the Euler formula and projects the entities into the complex plane and then represents the relations as the rotation from the head entity to the tail entity. The HAKE [18] model specifically tackles the modulus setting based on the RotatE model and put the entities and relations into the polar coordinates plane where the modulus and angle can be adopted in the representation. The QuatE [22] model introduces the hyper-complex plane, where the entity and relation are trained as quaternionic vectors in the hypercomplex space with three imaginary components. Compared with RotatE, QuatE can be seen as rotation in two planes.

Besides, different from the translational distance KGE models that only use the observed triples for the model training, many **path-enhanced model**s are designed to mine the multi-step relation paths from the original KG, as a supplement to represent the relations and entities. The PTransE [14] model is the first model that mines the multi-step paths with its Path-Constraint Resource Allocation (PCRA) algorithm, then the paths are adopted in the training of relations. Later on, other models try to improve the representation of the paths. The RPJE [19] model adds the logical rules mined from KG to improve the explainability and accuracy in the multi-step path representation. The PRRL [15] model adopts the Hadamard product of relations to represent the path and enhances the relation patterns representation ability of the paths. There are also many works that tackle the usage of the paths. The HRAN [23] model adopts the GNN framework to aggregate the entities and relations based on the multi-step paths and the attributed weight of the paths. The HARPA [24] model uses the original KG triples and the multi-step paths with a well-designed two-layer attention encoder and a GAN framework to learn the entity/relation embeddings. The BKENE [25] model adopts the KG triples and multi-step relation paths to generate a new view of KG for the replacement of the negative sampling process.

2.2 Collaborative filtering RS and Other RS models

To build a recommendation model, traditional collaborative filtering (CF) RS [26] is based on the User-Item interaction matrix, which is usually formed by each user's choice of each item (such as 1 represents like/checked and 0 means not), or the score each user gives to each item. Based on this matrix, RS can learn the user/item embedding with a well-designed objective function and calculate the preference score from each user to each candidate item, then rank items based on the scores and recommend the top-k items to a user.

The DKN model [6] is also based on the general CF idea. Besides the User-News interaction matrix, which contains each user's historical checked news and if this user read the news or not, DKN also adopts a real-world KG as side information, and the title of each news as content information. Each title is represented as vectors through NLP techniques, also the useful words are extracted from those titles to link to their related entities in KG. After that, the embedding of the title, entities that appeared in that title, and the context entities form a three-dimension vector set, and the vector set is processed by a knowledge-aware CNN proposed by DKN to obtain the final embedding of each news. The next step is using the Attention mechanism to extract users' interests in different news. After that, the news' embedding and users' embedding can be obtained, and based on them the recommendation can be made.

The KGAT model [7] is different. Since it can make recommendations for music, book, and spot, it only uses the KG that is related to these three targets as side information. In this model, user/item interactions are treated as a special triple and formed a collaborative KG (CKG) with side information KG. Then KGAT puts the CKG into the KG embedding layer to obtain the preliminary embedding of all the nodes on the CKG. Then KGAT proposes an attention-based graph convolution network to adopt the high-order embedding propagation from entity to user, where the entities are treated as attributes to enrich the features of items, and the items are used to contribute to the representation of users. Finally, in the prediction layer, KGAT uses all the propagated information of users and items from all the graph layers, then conducts the prediction.

Besides, with the development of deep learning, many new technologies are adopted in the KG-based RS area. For example, the KGCL [8] model and the MCCLK [27] model divide the user/item matrix and the KG triples into different cross-view, then both use the Contrastive Learning method to reduce the noise between the cross-views. The Reinforcement Learning method is also a hot topic in the RS area. In the CGKR [28] model, the Reinforcement Learning technique is adopted in two counterfactual generators that generate fake interactions to reduce the spurious correlations in the RS, and the two generators are joint-trained with a GNN framework for the recommendation. The SAPL [29] model improves the explainability of RS with its sentiment-aware KG and Reinforcement Learning policy for the recommendation and reasoning. Yet in those models, the fundamental idea is based on the CF-RS to calculate the user/item preference.

2.3 Summary

The KGE-based RS model, such as the DKN [6] model, and the KGAT [7] model, use the translational KGE method to get the entity/relation embedding. However, the translational KGE methods have the following problems:

(1) **Relation pattern representation problem**. Research on the Rotation models [13] raises the problem that translational distance KGE methods lack the ability to represent certain types of relations. To be specific, the TransE model cannot represent the variety of symmetry relations. Based on its score function, the symmetry relations tend to be trained as zero vectors, which fail to differentiate all the symmetry relations. The remaining three models TransH, TransR, and TransD are unable to represent the inversion and the composition relations, due to the projections being invertible matrix multiplications.

(2) **Multi-step relation path representation problem**. Research of the Path-enhanced models [14] raises another problem: translational distance models ignore the connections between the multi-step relation paths, as they only adopt the observed triples in the original KG for the model training. Thus, the high-level semantic information will be wasted, which results in reducing the RS performance precision.

We consider these two problems of the KGE layer would affect the whole recommendation system's performance by reducing the recommendation precision. Inspired by KGE model as RotatE [13], PTransE [14], and PRRL [15], we adopt different KGE layers in the DKN model and KGAT model to address these two problems.

3.Proposed Method

In this section, we introduce how we solve the two problems listed in Section 2.3, i.e., 1) the Relation pattern representation problem and 2) the Multi-step relation path representation problem. In Section 3.1, we first explain our idea with examples of the DKN model and the KGAT model, then we explain our idea, including the general structure of an RS and the KGE layer in the RS. Section 3.2 is the solution to the relation pattern representation problem: inspired by the RotatE model, we map the entities and relations into the complex vector space and define each relation as the rotation from the source entity to the target entity. Section 3.3 is the solution to the multi-step relation path representation problem: we use a path extraction approach proposed by the PTransE model and add the path information as a constraint in the KGE layer. In section 3.4, combining these two strategies, we add another method that can solve the two problems simultaneously.

3.1 KGE layer in RS

In this research, our idea is to replace the KGE layer in the original RS model with other KGE layers with better entity/relation representation ability. With the proper KGE layer processing the entity/relation as a supplement of the RS, the ability of the model to predict precision should be improved. For example, the DKN [6] model and the KGAT [7] model both adopt the translational KGE method to obtain entity/relation embedding. However, the translational KGE methods have two problems: lack the ability to represent certain patterns of relations in KG, and ignore the multi-step relation paths, thus affecting the prediction precision. Therefore, we try to adopt different KGE layers in the DKN model and KGAT model to address these two problems.

A KGE layer in a certain KG-based RS is always used as a first layer, as an initial input representation of the user/item in the RS, while preserving the semantic information and the original KG structure. For example, in the DKN model, the KGE layer is TransD; in the KGAT model is TransR; in the KGCL model is TransE. The entities/relations in KG are adopted as attributes to the user/item in the RS, and the KGE layer is designed to help better encode the semantic words of entities and relations into the form of vectors or matrices. Then the embedding results of entities/items will be used with a specialized recommendation algorithm to build a KG-based RS. In Figure 3, we give a general structure of a KG-based RS that uses the KGE layer as the initial input.

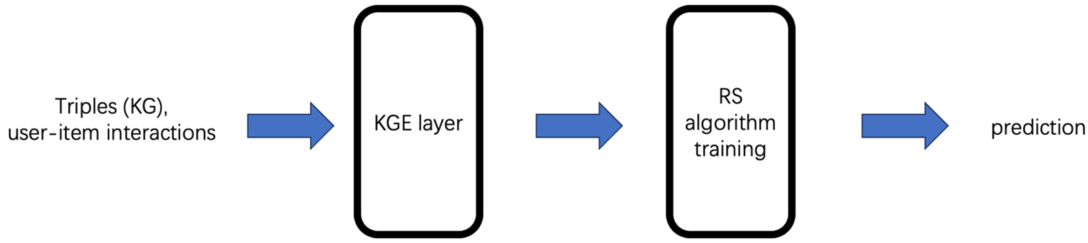

Figure 3. General structure of KGE-RS such as DKN and KGAT

The importance of the KGE layer in the RS can be summarized into two reasons: to improve the RS precision and build the pre-trained model.

First, in the traditional CF-based RS, which only uses user/item interactions as its input resource for model building, it only generates the first user/item embeddings with a random algorithm such as Gaussian distribution. Although they can be trained to be more sophisticated later with other deep learning mechanisms, the embeddings of users and items still lack the variety of different kinds of attributes and related connections that can be obtained from the KG. With the KG as the side information, the prediction precision will be improved.

Second, the KGE as the initial input approach can be used to save the time of model training since the KGE process can be done separately from the RS process in the real-life industry. With the rapidly growing KGE technology, many useful KGE models with delicate score functions and target-specific purposes are proposed, making the retrieving of entities/relations embeddings hard to avoid using KGE methods, especially in the KG-based RS area. After the entities and relations are pre-trained with the KGE methods, the embeddings can be adopted in the downstream task.

3.2 Relation pattern representation

The first problem that existed in the traditional KGE methods is that: certain relation patterns cannot be represented properly with the score function used in those KGE methods. According to RotatE [13], general relation patterns can be divided into four categories: the symmetry relation, the asymmetry relation, the inversion relation, and the composition relation. Among those relation patterns, the TransE model cannot represent the symmetry relations as all the relations will be encoded close to zero vectors; the TransH, TransR, and TransD models cannot represent the inversion and the composition relations due to the projection process of entities and relations are invertible matrix multiplications.

We adopt the layer of RotatE method to solve this problem. The idea of this KGE layer is to project entity and relation into a complex plane, where a relation can be treated as rotation from its head entity to its tail entity. Figure 4 is an example adapted from [13] that shows the RotatE methods with only one dimension of embedding. The vectors **h** and **t** are the head entity and the tail entity. Relation r is represented as a rotation angle, where lead the vector **h** rotates into a new vector **hr**. After that, the distance between the head entity and the tail entity can be calculated as |hr -t|.

We give the score function of the KGE layer used for solving the problem that translational KGE methods can not represent a certain type of relations into various vector forms. Notice that in order to keep the embeddings of entities and relations in the form of vectors, we drop the dimension of the modulus in the embeddings, and only keep the angle dimension. The score function for learning the embedding of triple(h, r, t) is shown in Equation (1), where $\theta_h, \theta_r, \theta_t$ are high-dimensional vectors, which are used to represent the angles of entities and relations, and each dimension of those θ vectors should range between$[-\pi,\ \pi]$:

$$E(h, r, t) = ||sin((\theta_h + \theta_r - \theta_t)/2)||. \quad (1)$$

With the help of score function (1), all the relations will be represented as vectors formed by angles, and each pattern of relations will be represented by vectors that satisfy the mathematical connections with their angles. For example, when representing a symmetry relation r, each dimension of the vector θ_r will be 0, π, or 2π. For a pair of inversion relations r_1 and r_2, the sum of the two vectors θ_{r1} and θ_{r2} on each dimension should be 0 or 2π, as $\theta_{r1}, +\theta_{r2} = 0$ or $\theta_{r1} + \theta_{r2} = 2\pi$. For a set of three combination relations, such as r_1 and r_2 can combine as r_3, each set of their dimension satisfies $\theta_{r1} + \theta_{r2} = \theta_{r3}$ or $\theta_{r1} + \theta_{r2} = \theta_{r3} + 2\pi$. The formal mathematical proofs of these three conclusions are listed in the paper on RotatE [13] model. Therefore, the relation pattern representation problem can be solved.

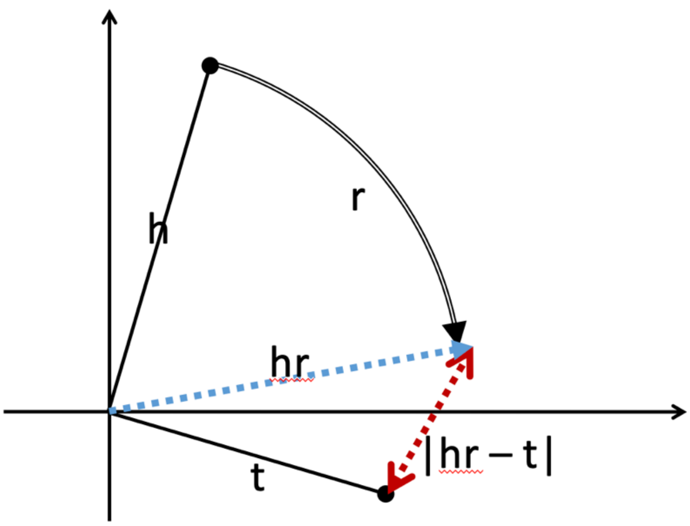

Figure 4. RotatE method with 1-dimensional embedding, based on Figure 1 in [13].

3.3 Multi-step relation path representation

Translational KGE models do not consider the connections between the multi-step relation paths, as they only use the observed triples in the original KG for the model training, making the embedding of entities and relations only need to be suitable for the original triples in KG but not compatible with the unobserved triples. Thus, the embeddings may not be useful for downstream tasks.

Inspired by PTransE [14], we first use the Path-Constraint Resource Allocation (PCRA) algorithm to mine the path combined between two relations. PCRA algorithm measures the reliability of a multi-step relation path by comparing how many resources will be distributed from a head entity to a tail entity through a certain path.

The multi-step relation path can be seen as the constraints of the relations. Sharing the same idea with entities on the translational distance space, if a triple (h, r, t) and a path (h, p, t) hold respectively, the relation r and the path p should be as close as possible on the translational space. Figure 5 adapted from PTransE [14] shows the difference between translational distance KGE and Path-enhanced KGE, where the latter adopts the path information into consideration. As Figure 5 is shown, three relations *BornInCity, CityInState, and StateInCountry* can form a multi-step relation path, while the head entity is *Bob* and the tail entity is *America*. The path representations are computed by the semantic composition of relation embeddings, then the embedding of this path should be calculated nearly the same as the relation *BornInCountry* in the training phase, since *(Bob, BornInCountry, America)* is an observed triple in the KG.

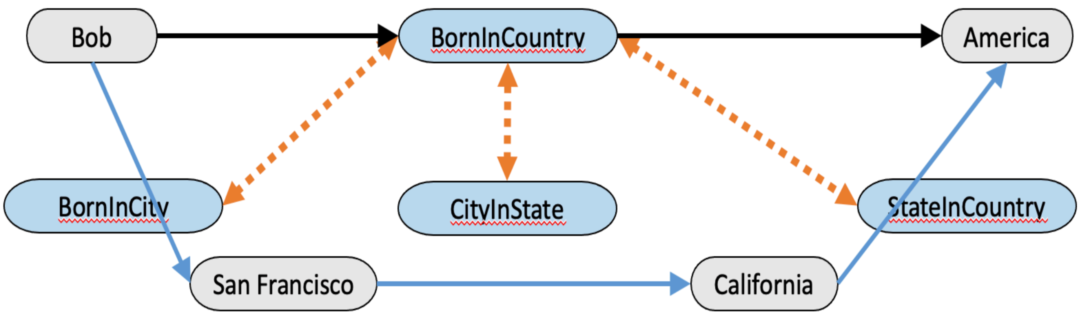

Figure 5. An example of a multi-step relation path used as a relation in a triple, based on Figure 2 in [14].

After the mining phase, we can get both the path and its weight in the embedding training. The score functions for triples and paths are designed as equations (2) and (3) respectively:

$$E(h,r,t) = ||h + r - t||, \tag{2}$$

$$E(p,r) = \sum_{p \in P(h,t)} R\,(p|h,t)\,||p - r||, \tag{3}$$

where the path $p = \{r_1, r_2 \cdots r_n\}$ is represented as $p = r_1 + r_2 + \cdots + r_n$, $P(h,t)$ is the set of paths from h entity to t entity, and $R(p|h,t)$ is the reliability of each path. And this is the solution to adding the path information to the translational distance KGE methods.

3.4 Combination of the two solutions

We also propose a KGE layer that can deal with the relation pattern representation problem and the multi-step relation path representation problem simultaneously. PRRL [15] shows us the possibility to solve these two problems at the same time. In this method, we project the project entity and relation into a complex plane, where a relation can be treated as a rotation from its head entity to its tail entity. Likewise, we also project multi-step relations into this complex plane, where a path can be treated as a combination of rotations by the relations that forms this path. We adapt Figure 6 from [15] to illustrate how the path is projected into the complex plane. The multi-step path p can be seen as a combination of relations r1 and r2, where relations are the rotation directions on the complex plane. The head entity vector h is translated by the rotation path, then the distance with the tail entity t can be calculated.

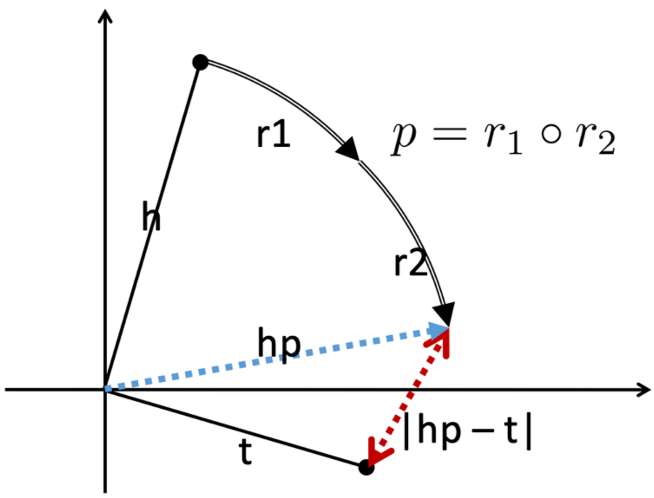

Figure 6. Path combined with its relations on 1-dimensional space, based on Figure 2 in [15].

The scores functions are similar to what we used in the former two sections, where we adopt the angle vector θ as the representation of entities and relations, and the representation of path can be encoded by the combination of the relations:

$$E(h,r,t) = \left||sin(\theta_h + \theta_r - \theta_t/2)|\right| \tag{4}$$

$$E(p,r) = \sum_{p \in P(h,t)} R\,(p|h,t)||\sin\,(\theta_p - \theta_r/2)||\,, \tag{5}$$

and here we have $\theta_p = \theta_{r_1} + \theta_{r_2} + \cdots + \theta_{r_n}$, which is the representation of the multi-step path on the complex plane. The same to section 3.2, each dimension of a symmetry relation will be encoded as 0, π, or 2π. And for a pair of inversion relations r_1 and r_2, the sum on each of their dimensions is 0 or 2π. For a set of three combination relations, such as r_1 and r_2 can combine as r_3, on each dimension, the result of $\theta_{r1} + \theta_{r2}$ should be $\theta_{r3}\ or\ \theta_{r3} + 2\pi$. Therefore, the relation pattern representation problem can be solved. Also, since the multi-step path is used as a constraint to its target relation in the model training, we add the path information to the translational distance KGE methods.

4. Experimental Evaluation

This section presents the datasets, the baselines, the experimental setup, the experimental results, and our analysis. To validate our proposed methods, we conducted two experiments on improving the DKN model and the KGAT model separately.

4.1 Datasets

We used the dataset provided by the DKN group [6]. They minded the User and News information and the historical interactions from the Bing News logs, then separated the datasets into the training set and testing set by the timeline. The KG datasets related to the User/News interactions are extracted from the Microsoft Satori knowledge graph.

Because of privacy reasons, DKN cannot provide the original news data, and a small sample of KG and News data was provided instead, about 10% of the size of the original data set used in the DKN paper. The dataset is separated by the DKN group in chronological order. Table 2 shows the data size of the KG we used, including the number of entities, relations, and triples. Table 3 shows the data size of the user/item interaction matrix, including users, news, and interactions we used in our tests.

Table 2. Basic statistics of the KG.

Entity	Relation	Triple
36,350	1,166	772,258

Table 3. Basic statistics of the User/News dataset for training and testing.

	Train	Test
User	469	138
News	8,663	407
Interactions	10,401	462

For the experiments on the KGAT model, we used the same datasets used in the original KGAT model [7], which are three datasets mined from three open source databases Amazon-book, Yelp2018, and Last-FM firstly by the KGAT group, then the three raw datasets are separated into the training set and testing set. The usage domains of these datasets are book, spot, and music recommendation, respectively. The KG datasets related to these three User/Item datasets are extracted from the Freebase [1] KG database, which are also public datasets we got from the original paper. Notice that the dataset is separated by the KGAT group. Table 4 and Table 5 give the basic statistics of the KG datasets and the User/Item datasets.

Table 4. Basic statistics of the KG, based on Table 1 in [7].

	Entity	Relation	Triple
Amazon-book	88,572	39	2,557,746
Yelp2018	90,961	42	1,853,704
Last-FM	58,266	9	464,567

Table 5. Basic statistics of the User/Item dataset for training and testing.

		Train	Test
Amazon-book	User	70,679	70,679
	Item	24,915	21,832
	Interaction	652,514	193,920
Yelp2018	User	23,566	23,566
	Item	48,123	46,720
	Interaction	2,418,427	616,336
Last-FM	User	45,919	45,919
	Item	45,538	39,469
	Interaction	930,032	253,578

4.2 Baselines and experiment settings

In the experiment of improving the performance of the DKN model, we reproduced four different translational distance KGE layers as baselines: TransE [9], TransH [10], TransR [11], and TransD [12], on the dataset provided by DKN. Since the usage of the TransD layer got the highest prediction precision among the four baselines in the original DKN experiments, the comparison is mainly based on the DKN model with the TransD KGE layer and our three proposed methods, where we separately adopted the RotetE [13] KGE layer to solve the relation pattern representation problem, and the PTransE [14] layer for the multi-step path representation, and the PRRL [15] layer that can solve these two problems at the same time.

We also conducted a small experiment on the KGAT model, to confirm our idea and methods are applied to different KGE-based RS models. We reproduced the KGE layer in the KGAT model with the TransE method as the baseline, and the RotatE method as the comparison, as for trying to improve the recommendation precision by enhancing the relation pattern representation ability. Table 6 shows the baselines and proposed methods used in these two experiments.

Table 6. Baselines and proposed methods

	DKN	KGAT
Baseline	TransE TransH TransR TransD	TransE
Proposed method	RotatE PTransE PRRL	RotatE

For evaluation metrics, in the experiment of the DKN model, we chose AUC, ACC, and F1 values following [30]. In the experiment of the KGAT model, we chose Recall@20 and NDCG@20 following the original KGAT [7] experiments. For the hyper-parameters, Table 7 and Table 8 show the hyper-parameters used in these two models.

Table 7. Hyper-parameters of DKN

Embedding dimension	50
Number of filters	100
Filter size	[1, 2]
L2 regularization weight	0.01
Learning rate	0.0001
Batch size	128
Optimizer	Adam

Table 8. Hyper-parameters of KGAT

Embedding dimension	64
Aggregation output	[64, 32, 16]
Message dropout	[0.1, 0.1, 0.1]
L2 regularization weight	1e-5
Learning rate	0.0001
Batch size	1024
Optimizer	Adam

4.3 Experimental results

4.3.1 DKN model

Table 9 shows the results of the DKN model of four baselines and our three proposed methods. The proposed KGE layers and the best results are in bold font. Notice that the improvement percentage we listed in the parenthesis is compared to the TransD layer, which is marked by the underscores.

We conducted the statistically significant test between the four baseline layers and the three proposed layers, turned out they are statistically different as the p-values are all less than 0.05. From Table 9, we can observe the following results:

Table 9. Experiment results on the DKN model

KGE layers	AUC	ACC	F1
TransE	0.5774	0.4762	0.5526
TransH	0.5795	0.4870	0.5604
TransR	0.5907	0.5043	0.5663
<u>TransD</u>	<u>0.5925</u>	<u>0.5173</u>	<u>0.5685</u>
PTransE(proposed)	**0.6177**(+4.3%)	**0.5411**(+4.6%)	**0.5688**
RotatE(proposed)	0.5987	0.5346	0.5680
PRRL(proposed)	0.5935	0.5325	0.5681

Significant T-Test was performed to confirm the statistically different ($p<0.05$) between baselines and the proposed methods

(1) Among the four baselines with TransE, TransH, TransR, and TransD layer, DKN with the TransD layer got the best performance, and the significant t-test between TransD and the other three baselines also shows the statistical difference ($p<0.05$), which is the same as observed in the DKN [6] paper.

(2) Although there is not a large gap between TransD and the three proposed methods in the F1 score, our proposed three methods can outperform TransD on AUC and ACC, which proves that the two problems we observed in the original DKN model affected the performance of the whole RS model. Compared to DKN with the TransD layer, the PTransE layer method gets the highest improvements, with a 4.3% improvement in AUC and a 4.6% improvement in ACC.

(3) Although there are differences among the results of the three proposed methods, the significant T-test between each of the two proposed methods showed that there is no statistical difference among the three methods, as $p>0.05$. We suppose the reason probably is that the essence of these three methods is to enhance the representation of relations from different perspectives, so that their processing of relation embeddings may have similar expressions, making the statistical difference not significant. We will leave the more detailed research in future work.

4.3.2 KGAT model

Tables 10 to 12 are the results of the experiments on the KGAT model in three different datasets. This experiment is used as side evidence to prove that our methods work on different RS models that adopt the KGE layer.

We also conducted the statistically significant test between the baseline TransE layer and proposed RotatE layers on three datasets, turned out they are all statistically different as the p-values are all less than 0.05. From these three tables, we can observe that:

Table 10. Experiment results on the Amazon-book dataset of KGAT

Amazon-book	Recall@20	NDCG@20	p-value
TransE	0.1352	0.0717	
RotatE(proposed)	0.1367(+1.1%)	0.0722(+0.7%)	0.0286

Table 11. Experiment results on the Last-FM dataset of KGAT

Last-FM	Recall@20	NDCG@20	p-value
TransE	0.0806	0.0684	
RotatE(proposed)	0.0814(+1.0%)	0.0694(+1.4%)	0.0218

Table 12. Experiment results on the Yelp2018 dataset of KGAT

Yelp2018	Recall@20	NDCG@20	p-value
TransE	0.0656	0.0419	
RotatE(proposed)	0.0662(+0.9%)	0.0425(+1.4%)	0.0078

Significant T-Test was performed on each dataset to confirm that the Baseline model and our proposed model are statistically different ($p<0.05$)

(1) All the improvements compared from the RotatE layer to the TransE layer baseline are around 1%, suggesting that the KGE layer in an RS that can represent the variety of relation patterns can bring an outperformed result.

(2) The success of the KGAT model confirms our idea and methods are applied to different KGE-based RS models, which means a KG-based RS adopted a KGE layer as its initial input layer should consider a better KGE layer to obtain more accuracy and specific KG embeddings.

5.Conclusion

In this research, we proposed a new strategy to improve the performance of an RS that adopted a KGE layer as its initial input layer. We adopted three different KGE layers and solved two problems that existed in the original model, including lacking the ability to represent certain types of relations in KG and ignoring the rich inference patterns contained in the multi-step relation paths. We assume that a KGE layer with higher accuracy performance will bring the whole system better recommendation results, and we conducted a large number of experiments to prove the correctness of our hypothesis, where we got around 4.5% improvement on AUC and ACC in the experiment of the DKN model, and around 1% improvement on Recall@20 and NDCG@20 on the KGAT model.

Actually, in the real-life industry, the training of the KGE process is time-consuming, but since the training process of entity/relation embedding can be separated from the RS model build training, the time-consuming can be avoided. Also, the growing corpus of KG can help retrieve the KG embeddings with higher accuracy. Yet the time-consuming is still a problem. For future work, I may conduct research on how to deploy an RS in the real-life industry and try to reduce the time-consuming, to better exploit the potential of the knowledge graph.

Reference

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.
[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “Dbpedia: A nucleus for a web of open data,” in *Proceedings of international semantic web conference*, 2007, pp. 722–735.
[3] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: a core of semantic knowledge,” in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.
[4] T. Mitchell *et al.*, “Never-ending learning,” *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.
[5] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, “Collaborative knowledge base embedding for recommender systems,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 353–362.
[6] H. Wang, F. Zhang, X. Xie, and M. Guo, “DKN: Deep knowledge-aware network for news recommendation,” in Proceedings of the world wide web conference, 2018, pp. 1835–1844.
[7] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, “Kgat: Knowledge graph attention network for recommendation,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.
[8] Y. Yang, C. Huang, L. Xia, and C. Li, “Knowledge Graph Contrastive Learning for Recommendation,” arXiv preprint arXiv:2205.00976, 2022.
[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” *Advances in neural information processing systems*, vol. 26, 2013, pp. 1-9.
[10] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proceedings of the AAAI conference on artificial intelligence*, 2014, vol. 28, no. 1, pp. 1-8.
[11] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in *Proceedings of the AAAI conference on artificial intelligence*, 2015, vol. 29, no. 1, pp. 1-7.
[12] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, “Knowledge graph embedding via dynamic mapping matrix,” in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.
[13] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, “Rotate: Knowledge graph embedding by relational rotation in complex space,” arXiv preprint arXiv:1902.10197, 2019.
[14] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, “Modeling relation paths for

representation learning of knowledge bases," *arXiv preprint arXiv:1506.00379*, 2015.
[15] C. Bai and P. Wu, "PRRL: Path Rotation based Knowledge Graph Representation Learning method," in *2021 IEEE/ACM 8th International Conference on Big Data Computing, Applications and Technologies (BDCAT'21)*, 2021, pp. 38–45.
[16] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.
[17] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo, "Multi-task feature learning for knowledge graph enhanced recommendation," in Proceedings of the world wide web conference, 2019, pp. 2000–2010.
[18] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 3, pp. 3065–3072.
[19] G. Niu *et al.*, "Rule-guided compositional representation learning on knowledge graphs," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 3, pp. 2950–2958.
[20] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, "Stranse: a novel embedding model of entities and relationships in knowledge bases," *arXiv preprint arXiv:1606.08140*, 2016.
[21] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30, no. 1, pp.985-991.
[22] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," *Advances in neural information processing systems*, vol. 32, 2019, pp. 1-11.
[23] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong, "Learning knowledge graph embedding with heterogeneous relation attention networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3961–3973, 2021.
[24] N. Zhang, J. Wang, and J. He, "HARPA: hierarchical attention with relation paths for knowledge graph embedding adversarial learning," *Data Mining and Knowledge Discovery*, vol. 37, no. 2, pp. 521–551, 2023.
[25] J. S. Kim, S. J. Ahn, and M. H. Kim, "Bootstrapped Knowledge Graph Embedding based on Neighbor Expansion," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4123–4127.
[26] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.
[27] D. Zou *et al.*, "Multi-level cross-view contrastive learning for knowledge-aware recommender system," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1358–1368.
[28] S. Mu, Y. Li, W. X. Zhao, J. Wang, B. Ding, and J.-R. Wen, "Alleviating spurious correlations in knowledge-aware recommendations through counterfactual generator," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1401–1411.
[29] S.-J. Park, D.-K. Chae, H.-K. Bae, S. Park, and S.-W. Kim, "Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable

recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 784–793.
[30] X. Cao *et al.*, "DEKR: description enhanced knowledge graph for machine learning method recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 203–212.

Acknowledgment

First, I want to sincerely express my thanks to my supervisor, Yamana-sensei. He is a true gentleman I have ever seen, who always has patience in the course teaching and group discussion, and shares with us students with carefulness and suggestions. I would also extend my gratitude to my senior, doctor student Mo Fan, who always helps his group member with his optimistic attitude, also he helped me a lot in finishing this master thesis. And for Chen Chongxian and Fan Xin, my dear group members who shared so many days and nights in Building 55 with me, these guys are always my good friends and role models I will look up to. Zhang Haohuan, Amanda, and Gai Yang, who helped me a lot when I remote studied in China during the covid years; senior Feng Jiaying and Huang Houtao and all my other friends, we spent our days in school and outside. Finally, I really appreciate that I have the chance to study at Waseda University as a master student. Not only I have learned much knowledge from here in the last two years, but also the growth spiritually will affect my life. I will always, always, always be proud to be a student of Waseda University.