Trustworthiness-Aware Knowledge Graph Representation for Explainable Recommender Systems

Yan Ge and Haiping Lu

University of Sheffield, Sheffield, UK
{yge5,h.lu}@sheffield.ac.uk

Abstract. Incorporating knowledge graphs (KGs) into recommender systems (RS) is promising in improving the performance and explainability of recommendation. Existing methods assume that the incorporated KG is completely correct without any error. However, errors and noise are inevitably introduced KGs during construction, which probably can hurt the performance of RS that treat errors in KGs as true ones. Motivated by it, we propose a novel end-to-end framework **trust**worthiness-aware knowledge graph representation for explainable **rec**ommender systems (TrustRec). Specifically, TrustRec can detect and measure possible noises by leveraging three-level structural information in the KG from a small to the big picture, namely subgraphs (i.e., motifs), communities and global. The output trustworthiness is viewed as the weight of knowledge to incorporate with RS. If the higher trustworthiness of the knowledge is determined, the more knowledge will be integrated into the RS. Additionally, to distinguish attributes between users (e.g., Thomas as a user) and items (e.g., Titanic as a movie) in which both are very different types of objects, we develop a new translation-based RS method to enable the user and item to utilise different mapping functions.

Keywords: Knowledge Graph Representation Learning · Explainable Recommender Systems · Trustworthiness

1 Introduction

The explosive growth of media services has provided overwhelming choices for users, such as movies, music and series. Recommender systems (RS) aims to ease information explosion and largely reduce users' effort in finding things of interest. Collaborative filtering (CF) [19] is a popular method for RS, which learns user/item similarity from existing historical interactions. However, CF-based methods usually suffer from the sparsity of interaction and cold-start problem. To address these issues, existing works incorporate auxiliary sources as side information, such as social networks [15] and images [41].

Knowledge graphs (KGs) as one type of auxiliary sources contain rich knowledge in the form of heterogeneous graphs where nodes correspond entities and edges correspond to relations. Knowledge in KGs is presented as in the form of the triple (*head entity, relation, tail entity*) [36]. For example, (Donald Trump, president_of, America) indicates that Donald Trump is the president of America. Knowledge graph representation (KGR) aims to learn low-dimensional distributed embedding of entities and relations. Recently, some KGs (e.g., Freebase [4] and Probase [38]) are successfully applied to many applications such as question answering [10], text classification [35].

For recommendation, items can be mapped into KGs, and thus KGs can provide extra semantic connectivity information between items. The usage of knowledge graph within the context of recommender systems can address the item cold-start and sparsity problem of CF. The reason is that (1) KG introduces extra semantic connections among items, which can provide new items with more interactions to RS; (2) KG consists of a variety of relation types, which helps extend a user's interests reasonably and increasing the diversity of recommended items. Moreover, KG can bring explainability to recommender systems since KG connects a user's historical records and the recommended ones based on relations in the KG. Recently, knowledge-aware RS has shown great potential to improve accuracy and explainability. Collaborative Knowledge base Embedding (CKE) [41] combines CF with pre-processed knowledge graph embedding in a unified Bayesian framework. RippleNet [32] propagates users' potential preferences in the KG and explores their hierarchical interests. Multi-task learning for knowledge graph enhanced recommendation (MKR) [34] simultaneously train KGE and RS tasks with a deep end-to-end framework and complement each other. Knowledge-enhanced translation-based user preference model (KTUP) [7] considers the incomplete nature of KG and transfer the relation information from KG to RS for better understanding the reasons that a user likes an item.

However, when incorporating the knowledge from KGs to RS, the most existing methods [7, 32–34, 41] largely assume that the knowledge in KGs is completely correct without any noises. In the real-world KGs, some errors and noises are inevitably introduced in the process of automatically constructing large-scale KGs due to limited labour supervision [39, 18, 11]. Liang *et al.* [21] and Stefan *et al.* [13] verify the existence of errors and noises in KGs. It is essential to consider errors and noises in KGR incorporated with RS since KGR learns entities and relations with distributed representations mainly based on triple facts in KGs. Intuitively, errors and noises in KGs as auxiliary data possibly hurt the performance and interpretability of RS that treat errors in KGs as true ones. Additionally, to achieve explainability, KTUP [7] explicitly model preferences and regard the interaction between user and item as a form of translation. However, it fails to distinguish attributes between users and items in which both are very different types of objects. Typically, users (e.g., Tom as a customer) and items (e.g., Titanic as a movie) are different types of objects and thus they should be proceed in different ways.

To address the above problems, we propose a novel end-to-end framework **trust**worthiness-aware KGR for **rec**ommender systems (TrustRec). TrustRec can detect and measure possible noises in KGR as trustworthiness while incorporating with RS. Trustworthiness is a value within the interval [0, 1] that indicates the degree of certainty that the knowledge expresses. Specifically, to determine the trustworthiness, we propose a new method to leverage three-level structural information in KGs from a small to big picture, namely the subgraph (co-occurrence in the same type of motif), communities (co-occurrence in the same high association group) and global (correlation strength on all paths). The output trustworthiness is viewed as the weight of knowledge to incorporate with RS. If higher trustworthiness of the knowledge is determined, the more knowledge will be integrated into the RS. Furthermore, inspired by TransD [16], TrustRec can distinguish attributes between users and item by projecting users and items in different ways. We summarise our contributions as follows:

1. We propose TrustRec to detect noises in KGR while incorporating with RS by considering internal structural information of KGs that are the motif, communities and global.
2. To better model the interaction between users and items, we propose a translation-based method to distinguish the attribute between users and items.

2 Preliminary

2.1 Notations and Problem Formulation

In this section, we introduce notation and formulate the KG enhanced recommender systems problem. We denote scalars by lowercase italic letters, e.g., a, vectors by lowercase boldface letters, e.g., $\mathbf{a}$, matrices by uppercase boldface, e.g., $\mathbf{A}$.

We have a knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$, which is comprised of massive entity-relation-entity triples (h, r, t), in which $\mathrm{h} \in \mathcal{E}, \mathrm{r} \in \mathcal{R}$, and $\mathrm{t} \in \mathcal{E}$ denote the head, relation, and tail of a knowledge triple, $\mathcal{E}$ and $\mathcal{R}$ are the set of entities and relations in the knowledge graph, respectively. For example, the triple (*Forrest Gump, film.film.director, Robert Zemeckis*) states the fact that Robert Zemeckis is the director of the film "Forrest Gump". In many recommendation scenarios, an item $v \in \mathcal{V}$ corresponds to an entity $\mathrm{e} \in \mathcal{E}$ (e.g., item "Forrest Gump" in MovieLens also appears in the knowledge graph as an entity). The set of entities $\mathcal{E}$ is composed from items $\mathcal{V}$ ($\mathcal{V} \in \mathcal{E}$) and non-items $\mathcal{E}/\mathcal{V}$ (e.g., entities corresponding to item properties). We construct a directed graph G from a KG $\mathcal{G}$. Each entity $e \in \mathcal{E}$ is abstracted into a node. If there is a relation from the entities e_1 to e_2, a directed edge will exist from node e_1 to e_2. Therefore, a KG with n entities can be mapped as a directed graph G with n nodes.

In an explainable recommender system, let $\mathcal{V} = \{v_1, v_2, \cdots\}$ and $\mathcal{U} = \{u_1, u_2, \cdots\}$ denote sets of items and users. The user-item interaction matrix $\mathbf{Y}$ is defined according to users' implicit feedback. We represent each user behaviour as a triplet, $(u, preference, i)$, where $y_{uv} = 1$ indicates that user u has interacted with item v with a preference p transferred from $\mathcal{R}$, otherwise $y_{uv} = 0$. For example, a user Thomas watched a movie Forrest Gump due to a director (i.e., *film.film.director*).

Given user-item interaction matrix $\mathbf{Y}$ and knowledge graph $\mathcal{G}$, our task is to predict whether user u has a potential interest in item v with which this user has not engaged before. Specifically, we aim to learn a prediction function $\hat{y}_{upv}$ = $\mathcal{F}$(u, p, v|Θ, $\mathbf{Y}, \mathcal{G}$), where $\hat{y}_{upv}$ denotes the probability that user u will engage with item v with preference p, and Θ are model parameters of the function $\mathcal{F}$.

2.2 Trustworthiness in Knowledge Graph

Most traditional knowledge graph construction methods usually involve huge human supervision or expert annotation, which are extremely labour-intensive and time-consuming [39]. Recently, large-scale knowledge graphs (e.g., DBpedia [1], Freebase [4]) are productively and automatically constructed from from unstructured web text (e.g., NELL [8]). However, some noises and errors are inevitably introduced in the process of automation due to limited labour supervision [13, 21].

Existing KG-based tasks (e.g., knowledge completion [20]) or applications (e.g., question answering [23]) assume knowledge in the existing KG is completely correct. To model errors in KGs, Xie *et al.* [39] proposed a triple confidence awareness knowledge representation learning framework, which detects possible noises in KGs while learning knowledge representations with confidence simultaneously. They introduced the triple confidence to conventional translation-based methods for knowledge representation learning. Jia *et al.* [18] synthetically extracted trustworthiness of the triples from knowledge graph embedding, entity resource and path information of the knowledge graph. Most KGs representations consider deterministic KGs (e.g., Freebase) that consist of deterministic facts. Chen *et al.* [9] proposed a KGs embedding model on uncertain KGs that associate every fact with a confidence score. Dong *et al.* [11] built a large-scale uncertain knowledge graph, and fused multiple extraction sources with prior knowledge derived from an existing knowledge base.

2.3 Knowledge Graph Representation

Knowledge graph representation is used to embed entities and relations into low-dimensional vectors while preserving the semantic and structural information [17]. There are two categories of KGE methods: (1) Translational models exploit distance-based scoring functions. TransE [6] follows an assumption that $\mathbf{h}$ and $\mathbf{t}$ are connected by $\mathbf{r}$ with low error if a triple (h, r, t) holds, and thus formulates $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. However, TransE has flaws when dealing with 1-to-N, N-to-1 and N-to-N relations. To address these issues, TransH [37] introduces relation specific hyperplanes, which each relation r as a vector $\mathbf{r}$ on a hyperplane with $\mathbf{w}_r$ The embeddings $\mathbf{h}$ and $\mathbf{t}$ are first projected to the hyperplane of relation r to obtain vectors $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\perp \mathbf{h} \mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\perp \mathbf{t} \mathbf{w}_r$, and then $\mathbf{h}_\perp + \mathbf{r} \approx \mathbf{t}_\perp$. For TransE and TransH, the embeddings of entities and relations are in the same space. However, entities and relations are different types objects, it is insufficient

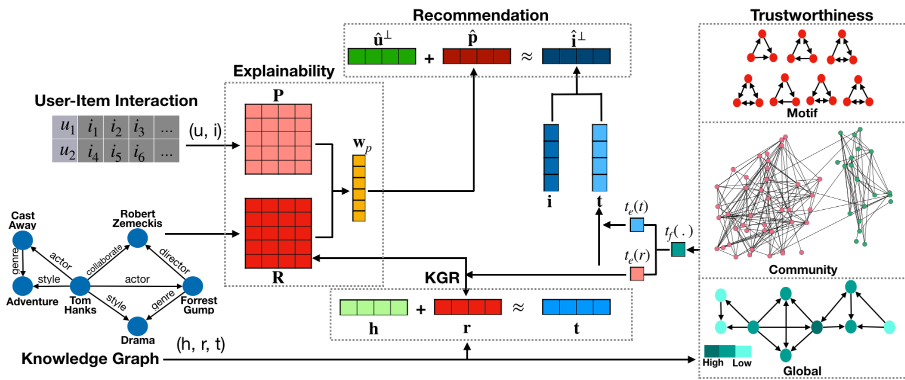

Fig. 1. The whole framework of TrustRec includes trustworthiness, recommendation, interpretability and KGR modules.

to model them in the same space. To address this issue, in TransR [22], $\mathbf{h}$ and $\mathbf{t}$ are projected to a new space so that relation r focuses on through the matrix $\mathbf{M}_r$ and then $\mathbf{M}_r\mathbf{h}+r \approx \mathbf{M}_r\mathbf{t}$. (2) Semantic matching models exploit similarity-based scoring function. They measure plausibility of knowledge triples by matching latent semantics of entities and relations, such as RESCAL [26], DistMult [40], and HolE [25].

3 Methodology

A general intuition behind TrustRec is that if the higher trustworthiness of the knowledge is determined, the more knowledge will be integrated into the RS. The whole framework of TrustRec (Fig. 1) can be functionally decomposed into four modules, namely trustworthiness, recommendation, interpretability and KGR. The left side is the input user-item interactions and a KG. The right side of the input is an explainability module. The middle top and bottom are the recommendation and KGR modules respectively. The right is a pool of three-level trustworthiness estimators. The output of estimators is refined and forms the input to weight the relation and entity embeddings while incorporating with RS.

3.1 Trustworthiness Module

Given a triplet (h, r, t), we propose a method to determine the trustworthiness of this triple from three-levels structural information that are subgraph (i.e., motifs), communities and global, and corresponding based on the below three estimators. 1) A triple (h, r, t) is contained by the same subgraph connectivity pattern; 2) A triple (h, r, t) is found in the same high association community; 3) resource is allocated to tail t from head h through all paths in the entire KG.

Motif-aware Trustworthiness. Motifs are fundamental subgraph patterns in graphs, and show complex connectivity patterns beyond two nodes [3, 2]. Motifs demonstrate very important local structures underlying various complex networks, such as social networks [12].

We use the strength of a tie between head h and tail t linked by a relation r to measure the trustworthiness of triple (h, r, t). If head h and tail t have a strong tie, it seems to be hopeful that the relation r occurred between head h and tail is trustful. Motifs is an effective approach to measure ties between two entities [3, 30]. For example, in a social network, two people who have a common friend are likely to be friends, so this common friend and two people constitute a triangular connectivity pattern. Intuitively, if two people have more common friends, the stronger strength of a tie between them can occur under the same connectivity pattern. This paper will focus on all triangular motifs as shown in Fig. 1, though our proposed method can be easily extended to other motifs.

Based on the above analysis, we take the input (h, r, t) from G, and quantify the strength of a tie for triple (h, r, t) by counting the number of motif type M_i containing this triple. Different type of triangular motifs reflect different connectivity patterns. Thus, we construct a feature vector $\mathbf{m}_i$(h, r, t) to consider all:

$$\mathbf{m}_i(\mathrm{h}, \mathrm{r}, \mathrm{t}) = \sum_{\mathrm{h},\mathrm{t} \in \mathcal{E}, \mathrm{r} \in \mathcal{R}} \mathbb{1}\left(\mathrm{h}, \mathrm{r}, \mathrm{t} \text{ occur in } \mathcal{M}_i\right) \tag{1}$$

where $\mathbb{1}(s)$ is the truth-value indicator function, i.e., $\mathbb{1}(s) = 1$ if the statement s is true and 0 otherwise. We form a feature vector $\mathbf{m}_i(\mathrm{h}, \mathrm{r}, \mathrm{t})$ where the i-th element indicates the number of type $\mathcal{M}_i$ motif containing (h, r, t). Given motif feature vector $\mathbf{m}_i(\mathrm{h}, \mathrm{r}, \mathrm{t})$, we use a L-layer multi-layer perceptron (MLP) [42] to extract a motif-aware trustworthiness value t_m (h, r, t)[1]:

$$t_m(\mathrm{h}, \mathrm{r}, \mathrm{t}) = \mathcal{M}(\mathcal{M}(\cdots \mathcal{M}(\mathbf{m_i}(\mathrm{h}, \mathrm{r}, \mathrm{t})))) = \mathcal{M}^L(\mathbf{m_i}(\mathrm{h}, \mathrm{r}, \mathrm{t})), \tag{2}$$

where $\mathcal{M}(\mathbf{x}) = \sigma(\mathbf{W}_m \mathbf{x} + \mathbf{b}_m)$ is a fully-connected neural network[2] layer with weight $\mathbf{W}_m$, bias $\mathbf{b}_m$, and nonlinear ReLU activation function $\sigma(\cdot)$. In the output layer of $\mathcal{M}^L(\cdot)$, instead of ReLU, we use a sigmoid function (Eq.(3)) to a return $t_m(\mathrm{h}, \mathrm{r}, \mathrm{t})$ in the range 0 to 1.

$$\delta(t_m(\mathrm{h}, \mathrm{r}, \mathrm{t})) = \frac{1}{1 + e^{-t_m(\mathrm{h}, \mathrm{r}, \mathrm{t})}}. \tag{3}$$

Community-aware Trustworthiness. A well-defined community in graphs should include nodes that have relatively higher association than nodes in different communities [24]. A pair of nodes are more likely to be connected if both are in the same community, and less likely to be connected if they do not share communities [28].

Inspired by the above, we assume that if a head h and a tail t is more likely to be contained in the same community, the relation from h to t has stronger

---

[1] We use the exponent notation L in Eq. 2 and following equations in the rest of this paper for simplicity

[2] Exploring a more elaborate design of layers is an important direction of future work.

reliability. We thus perform community detection task on G. We first convert G to an undirected graph G_u due to focus on the association of triples. We then use a standard spectral clustering [31] on G_u to conduct communities. The specific details are described below. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be an unweighted adjacency matrix of G_u where $\mathbf{A}(\mathrm{h,t}) = 1$ if (h, r, t) is a fact in $\mathcal{G}$, otherwise $\mathbf{A}(\mathrm{h,t}) = 0$. The degree matrix $\mathbf{D}$ is a diagonal matrix with diagonal entries $\mathbf{D}(i,i) = \sum_{t=1}^{n} \mathbf{A}(\mathrm{h,t})$, which is the degree of the entity h. The random walk matrix of G_u is defined as:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}, \tag{4}$$

which denotes the probability transition matrix of random walks on the G_u. We then compute the first k largest eigenvectors as input to k-means [14] to conduct final k communities.

To increase reliability and robustness of community-aware triple trustworthiness, we use different community detection strategies by adjusting the number of output communities (e.g., output five communities as the first strategy). We thus form a community indicator vector $\mathbf{s}$: $\mathbf{s}_i$ (h, r, t) = 1 if (h, r, t) in the same community under i-th strategy and 0 otherwise 0. We finally feed $\mathbf{s}_i$ (h, r, t) to a MLP to extract a community-aware trustworthiness $t_c(\mathrm{h,r,t})$:

$$t_c(\mathrm{h,r,t}) = \mathcal{C}^L(\mathbf{s}(\mathrm{h,r,t})), \tag{5}$$

where $\mathcal{C}(\mathbf{x}) = \sigma(\mathbf{W}_c\mathbf{x} + \mathbf{b}_c)$.

Global-aware Trustworthiness. We adopt source allocation theory in PageRank [27] to determine the trustworthiness at the level of global. We assume that the trustworthiness between entity pairs (h, t) will be higher, and more resource is passed from the head h through all paths to the tail t in a whole graph G. The amount of resource aggregated into t indicates the trustworthiness value between h and t.

Specifically, starting from h each node in the graph should be reached. In the initial state, the resource amount of h is 1, and all others is 0. In the process of resource allocation, the sum of all resources of nodes is always 1. We simulate resource flowing until distribution steady. The value of the resource on the tail entity is $t_g(\mathrm{t} \mid \mathrm{h})$, it is calculated as follows:

$$t_g(\mathrm{t} \mid \mathrm{h}) = (1-\alpha) \sum_{e_i \in \mathcal{D}} \frac{t_g(\mathrm{e_i} \mid \mathrm{h})}{d(e_i)} + \frac{\alpha}{n}, \tag{6}$$

where $\mathcal{D}$ is a set of nodes that have outgoing links to the node t, $d(e_i)$ is the out-degree of the node e_i. Thus, for each node e_i in $\mathcal{D}$, the resource flows from e_i to t should be $\frac{t_g(\mathrm{e_i}|\mathrm{h})}{d(e_i)}$. The nodes without outgoing links can cause the absorption of the resource. In order to prevent it, resource flow from each node may directly jump to a random node with the same probability α. This part of the resource that flows to t randomly is $\frac{1}{n}$.

Fusion of Estimators and Refinement. We fuse the above three-level trustworthiness values $t_m(\mathrm{h,r,t})$, $t_c(\mathrm{h,r,t})$and $t_g(\mathrm{t} \mid \mathrm{h})$ as a vector and feed it to a MLP to extract a final triple trustworthiness as follows:

$$t_f(\mathrm{h},\mathrm{r},\mathrm{t}) = \mathcal{F}^L([t_m(\mathrm{h},\mathrm{r},\mathrm{t}), t_c(\mathrm{h},\mathrm{r},\mathrm{t}), t_g(\mathrm{t} \mid \mathrm{h})]). \tag{7}$$

We generate a trustworthiness value for triple (h, r, t), but it cannot be directly used in knowledge transfer from KG to RS. The reason is that knowledge representation is in the form of the entity and relation rather than the triple. We thus need to refine triple trustworthiness $t_f(h,r,t)$ to entity $t_e(h)/t_e(t)$ and relation trustworthiness $t_e(r)$.

The knowledge representation learns entities and relations with distributed representations based on triples in KGs. Therefore if an entity is likely to involve in triples with high trustworthiness, the representation of this entity is trustful. In our method, we consider all triples involving this entity h by sum averaged the trustworthiness of all triples the h or tail r involves in. It is formulated as follows:

$$t_e(\mathrm{h}) = \frac{\sum_{\mathrm{t}' \in \mathcal{E}, \mathrm{r}' \in \mathcal{R}} t_f(\mathrm{h},\mathrm{r}',\mathrm{t}')}{n_h}, \quad t_e(\mathrm{r}) = \frac{\sum_{\mathrm{t}',\mathrm{h}' \in \mathcal{E}} t_f(\mathrm{h}',\mathrm{r},\mathrm{t}')}{n_r}, \tag{8}$$

where n_h and n_r are the number of triples entity h and relation r involve in. Note that entity tail t has the same case with entity head h. In the following modules, the output trustworthiness $t_e(\mathrm{h})$ and $t_e(\mathrm{r})$ will weight the entity h and relation r representation of KG while incorporating with RS. It instructs our model to pay more attention to those more convincing triples.

3.2 Recommendation Module.

In recommender systems, there are two common scenarios can be found 1) a variety of preferences from users exist; 2) different users may share the same preference to different items (i.e., N-to-N issue). To model the above two scenarios, inspired by TransH, a recent work [7] explicitly models user preferences and regards them as translational relationships (i.e., $\mathbf{u} + \mathbf{p} \approx \mathbf{i}$) between users and items to address the N-to-N issue. However, for a typical preference, all the user and item share the same mapping matrix. Therefore, it fails to distinguish attributes between users (e.g., Thomas as a user) and items (Titanic as a movie) in which both are very different types of objects.

To address this issue, inspired by TransD [16], we define a new score function

$$g(u,i;p) = ||\mathbf{u}^{\perp} + \mathbf{p} - \mathbf{i}^{\perp}||_2, \tag{9}$$

where $\mathbf{u}^{\perp} = \mathbf{M}_u\mathbf{u}$ and $\mathbf{i}^{\perp} = \mathbf{M}_i\mathbf{i}$. We interpret $\mathbf{M}_u$ and $\mathbf{M}_i$ are two different mapping matrix for the user u and item i respectively. A lower score of $g(u,i;p)$ indicates that an item i is recommended to a user u due to a preference p is likely to be true. we define the different mapping matrix of users ($\mathbf{M}_u$) and items ($\mathbf{M}_i$) to discriminate attributes between the user $\mathbf{u}$ and item $\mathbf{i}$ as follows:

$$\mathbf{M}_u = \mathbf{w}_p\mathbf{u}_p + \mathbf{I}, \quad \mathbf{M}_i = \mathbf{w}_p\mathbf{i}_p + \mathbf{I}, \tag{10}$$

where $\mathbf{w}_p$ is the preference projection vector, $\mathbf{u}_p$ and $\mathbf{i}_p$ are user and item projection vectors, $\mathbf{I}$ is a identity matrix.

To model user preferences, we introduce a set of preference latent matrix $\mathbf{P}$, and a row $\mathbf{p}' \in \mathbf{P}$ denotes one preference. Correspondingly, we define a preference projection matrix $\mathbf{W}_p$ and one row $\mathbf{w}_{p'} \in \mathbf{W}_p$. With the help of KG, the number of preferences can be automatically set and each preference is assigned with explanations (Sec 3.4). Given a user and an item, we attempt to explore the reason from a variety of preferences that this user likes this item. We design the below preference selection strategy by measuring the similarity between a user-item pair and a preference:

$$\theta(\text{u, i; p'}) = \sum_{\mathbf{p}' \in \mathbf{P}} \text{sim}(\mathbf{u} + \mathbf{p}', \mathbf{i}) \tag{11}$$

where sim(·) is a measurement of similarity (e.g., inner product). The high θ(u, i; p) indicates the significance of the preference p for the user u when purchasing the item i. Noting that we design a different preference selection strategy from [7] due to keep the consistency of definition of translation-based user-item interaction.

Considering that a user may like an item according to various factors, which have no distinct boundary. We thus adopt attention mechanism to combine multiple preferences as follows:

$$\mathbf{p}(\text{u}, \text{i}; \mathbf{P}) = \sum_{\mathbf{p}' \in \mathbf{P}} \theta(\text{u, i; p'}) \cdot \mathbf{p}'. \tag{12}$$

$$\mathbf{w}_p(\text{u}, \text{i}; \mathbf{W}_p) = \sum_{\mathbf{w}_{p'} \in \mathbf{W}_p} \theta(\text{u, i; p'}) \cdot \mathbf{w}_{p'}. \tag{13}$$

For simplicity, we use $\mathbf{p}$ and $\mathbf{w}_p$ to indicate $\mathbf{p}(\text{u}, \text{i}; \mathbf{P})$ and $\mathbf{w}_p(\text{u}, \text{i}; \mathbf{W}_p)$ in Eq. (9) and Eq. (10). We encourage that the translation distances of interacted items are smaller than random ones for each user through BPR Loss function [29]:

$$\mathcal{L}_p = \sum_{(u,i) \in \mathbf{Y} (u,i') \in \mathbf{Y}'} - \log \delta \left[g\left(u, i'; p'\right) - g(u, i; p) \right] \tag{14}$$

where $\mathbf{Y}'$ contains negative interactions by randomly corrupting an interacted item to a non-interacted one for each user.

3.3 Knowledge Graph Representation Module

Our TrustRec framework is designed to be sufficiently flexible. Some existing KGR methods covering from translational-based methods (e.g., TransD [16]), semantic matching (e.g., DistMult [40]) and deep learning (e.g., SME [5]), can be combined into TrustRec framework, i.e. without requiring changing the model or affecting its computational complexity. Here we take TranD as an example. The scoring function of trustworthiness-aware TransD is defined as follows:

$$E(\mathcal{G}) = \sum_{(h,r,t)\in\mathcal{G}} g(h,r,t)\cdot t_f(\mathrm{h},\mathrm{r},\mathrm{t}), \tag{15}$$

where $g(\mathrm{h},\mathrm{r},\mathrm{t}) = ||\mathbf{h}^{\perp} + \mathbf{r} - \mathbf{t}^{\perp}||_2$. Differing from conventional methods, we also introduce the triple trustworthiness $t_f(\mathrm{h},\mathrm{r},\mathrm{t})$. A higher triple trustworthiness implies that the corresponding triple is more credible, and thus should be more considered. In detail, $\mathbf{h}^{\perp}$ and $\mathbf{t}^{\perp}$ are projected entity vectors by projection matrix $\mathbf{M}_{rh}$ and $\mathbf{M}_{rh}$ respectively:

$$\mathbf{h}^{\perp} = \mathbf{M}_{rh}\mathbf{h}, \quad \mathbf{t}^{\perp} = \mathbf{M}_{rt}\mathbf{t}, \tag{16}$$

$$\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p + \mathbf{I}, \quad \mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p + \mathbf{I}, \tag{17}$$

where $\mathbf{r}_p \in \mathbf{R}_p$, $\mathbf{h}_p \in \mathbf{H}_p$ and $\mathbf{t}_p \in \mathbf{T}_p$ are the relation, head and tail projection vector. Finally, the training of TransH encourages the discrimination between valid triplets and incorrect ones using margin-based ranking loss:

$$\mathcal{L}_k = \sum_{(\mathrm{h},\mathrm{r},\mathrm{t})\in\mathcal{G},(\mathrm{h}',\mathrm{r}',\mathrm{t}')\in\mathcal{G}^-} [g(\mathrm{h},\mathrm{r},\mathrm{t}) + \gamma - g(\mathrm{h}',\mathrm{r}',\mathrm{t}')]_+ \cdot t_f(\mathrm{h},\mathrm{r},\mathrm{t}), \tag{18}$$

where $[\cdot]_+ \triangleq \max(0,\cdot)$, $\mathcal{G}^-$ contains incorrect triplets constructed by replacing head entity or tail entity in a valid triple randomly, and γ controls the margin between positive and negative triples. Noting that the triple trustworthiness $t_f(\mathrm{h},\mathrm{r},\mathrm{t})$ instructs our model to pay more attention on those more convincing facts.

3.4 Interpretability Module.

In the recommendation module, we explicitly model users' preferences, but it still cannot semantically explain based on users' purchase. To understand the reasons that a user likes an item, inspired by [7], we transfer the relation presentation ($\mathbf{R}$) in KG to preference representation ($\mathbf{P}$) in RS. For example, if a user has watched several movies directed by (relation) the same person (entity), we can hold that the director relation plays a critical role when the user makes decisions. Note that we need to consider trustworthiness of relations while transferring to preferences in RS. If a relation representation is trustful, and it should be more considered.

We extend our recommendation module by transfer the knowledge of relation to preference in RS. Specifically, we first define knowledge-enhanced score function

$$g(u,i;p) = ||\hat{\mathbf{u}}^{\perp} + \hat{\mathbf{p}} - \hat{\mathbf{i}}^{\perp}||_2, \tag{19}$$

where $\hat{\mathbf{p}}$ is knowledge-enhanced preference. It is generated by producing a similarity score $\gamma(\mathrm{u},\mathrm{i};\mathrm{p},\mathrm{r})$, and select a preference from relation $\mathbf{R}$ and preference $\mathbf{P}$:

$$\gamma(\mathrm{u},\mathrm{i};\mathrm{p}',\mathrm{r}) = \mathrm{sim}(\mathbf{u} + \mathbf{p}' + t_r(r)\cdot\mathbf{r}, \mathbf{i}) \tag{20}$$

$$\hat{\mathbf{p}}(\mathrm{u},\mathrm{i};\mathbf{P},\mathbf{R}) = \sum_{\mathbf{p}'\in\mathbf{P},\mathbf{r}\in\mathbf{R}} \gamma(\mathrm{u},\mathrm{i};\mathrm{p}',\mathrm{r})\cdot(\mathbf{p}'+t_r(r)\cdot\mathbf{r}). \tag{21}$$

where $t_r(\mathrm{r})$ is the trustworthiness and give a weight to the relation r so that trustful relation can get more attention while incorporating with the preference $\mathbf{p}'$ in RS. We define $\hat{\mathbf{u}}^{\perp}$ and $\hat{\mathbf{i}}^{\perp}$ as follows:

$$\hat{\mathbf{M}}_u = \hat{\mathbf{w}}_p\mathbf{u}_p + \mathbf{I}, \quad \hat{\mathbf{M}}_i = \hat{\mathbf{w}}_p\hat{\mathbf{i}}_p + \mathbf{I}, \tag{22}$$

$$\hat{\mathbf{w}}_p(\mathrm{u},\mathrm{i};\mathbf{R}_p,\mathbf{R}) = \sum_{\mathbf{r}\in\mathbf{R},\mathbf{r}_p\in\mathbf{R}_p} \gamma(\mathrm{u},\mathrm{i};\mathrm{p}',\mathrm{r})\cdot(\mathbf{r}_p+t_r(r)\cdot\mathbf{r}). \tag{23}$$

$$\hat{\mathbf{i}}_p^{\perp} = \mathbf{i} + t_e(h)\cdot\mathbf{h}. \tag{24}$$

In the above equations, $t_e(\mathrm{h})$ and $t_r(\mathrm{r})$ is used to weight the representation of the head entity and relation while transferring knowledge from the KG to RS. Higher triple confidence implies that the corresponding triple is more credible, and thus should be more considered. We train TrustRec using the overall objective function $\mathcal{L} = \mathcal{L}_p + \mathcal{L}_k$.

4 Conclusion and Future Work

In this paper, we proposed a TrustRec that can detect and measure possible noises in knowledge graph representations as trustworthiness while incorporating with recommender systems. To detect and measure possible noises, we propose a method to leverage three-level structural information in the KG from a small to the big picture, namely the motifs, communities and global. The output trustworthiness is viewed as the weight of knowledge to incorporate with RS. If the higher trustworthiness of the knowledge is determined, the more knowledge will be integrated into the RS. Additionally, to distinguish attributes between users and items in which both are very different types of objects, we develop a new translation-based RS method to enable the user and item to utilise different mapping functions. In the future, we will elaborately test the performance of the whole TrustRec on real-world datasets.

Acknowledgement

This work is supported by the Amazon Research Awards.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)

2. Benson, A.R., Gleich, D.F., Leskovec, J.: Tensor spectral clustering for partitioning higher-order network structures. In: Proceedings of the 2015 SIAM International Conference on Data Mining. pp. 118–126. SIAM (2015)
3. Benson, A.R., Gleich, D.F., Leskovec, J.: Higher-order organization of complex networks. Science **353**(6295), 163–166 (2016)
4. Bollacker, K., Cook, R., Tufts, P.: Freebase: A shared database of structured general human knowledge. In: AAAI. vol. 7, pp. 1962–1963 (2007)
5. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. Machine Learning **94**(2), 233–259 (2014)
6. 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)
7. Cao, Y., Wang, X., He, X., Hu, Z., Chua, T.S.: Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In: WWW. pp. 151–161 (2019)
8. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Twenty-Fourth AAAI conference on artificial intelligence (2010)
9. Chen, X., Chen, M., Shi, W., Sun, Y., Zaniolo, C.: Embedding uncertain knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3363–3370 (2019)
10. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over freebase with multi-column convolutional neural networks. In: ACL. pp. 260–269 (2015)
11. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: KDD. pp. 601–610 (2014)
12. Granovetter, M.S.: The strength of weak ties. In: Social Networks, pp. 347–367. Elsevier (1977)
13. Heindorf, S., Potthast, M., Stein, B., Engels, G.: Vandalism detection in wikidata. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 327–336 (2016)
14. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters **31**(8), 651–666 (2010)
15. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 135–142 (2010)
16. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL. pp. 687–696 (2015)
17. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition and applications. arXiv preprint arXiv:2002.00388 (2020)
18. Jia, S., Xiang, Y., Chen, X., Wang, K.: Triple trustworthiness measurement for knowledge graph. In: WWW. pp. 2865–2871 (2019)
19. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
20. Li, X., Taheri, A., Tu, L., Gimpel, K.: Commonsense knowledge base completion. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1445–1455 (2016)

21. Liang, J., Xiao, Y., Zhang, Y., Hwang, S.w., Wang, H.: Graph-based wrong isa relation detection in a large-scale lexical taxonomy. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
22. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI conference on artificial intelligence (2015)
23. Lukovnikov, D., Fischer, A., Lehmann, J., Auer, S.: Neural network-based question answering over knowledge graphs on word and character level. In: Proceedings of the 26th international conference on World Wide Web. pp. 1211–1220 (2017)
24. Malliaros, F.D., Vazirgiannis, M.: Clustering and community detection in directed networks: A survey. Physics Reports **533**(4), 95–142 (2013)
25. Nickel, M., Rosasco, L., Poggio, T.A., et al.: Holographic embeddings of knowledge graphs. In: AAAI. vol. 2, pp. 3–2 (2016)
26. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Icml. vol. 11, pp. 809–816 (2011)
27. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
28. Porter, M.A., Onnela, J.P., Mucha, P.J.: Communities in networks. Notices of the AMS **56**(9), 1082–1097 (2009)
29. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: IN: PROCEEDINGS OF THE 25TH CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE (UAI. Citeseer (2009)
30. Tsourakakis, C.E., Pachocki, J., Mitzenmacher, M.: Scalable motif-aware graph clustering. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1451–1460 (2017)
31. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
32. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In: CIKM. pp. 417–426 (2018)
33. Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., Wang, Z.: Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In: KDD. pp. 968–977 (2019)
34. Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M.: Multi-task feature learning for knowledge graph enhanced recommendation. In: WWW. pp. 2000–2010 (2019)
35. Wang, J., Wang, Z., Zhang, D., Yan, J.: Combining knowledge with deep convolutional neural networks for short text classification. In: IJCAI. pp. 2915–2921 (2017)
36. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
37. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Aaai. vol. 14, pp. 1112–1119 (2014)
38. Wang, Z., Huang, J., Li, H., Liu, B., Shao, B., Wang, H., Wang, J., Wang, Y., Wu, W., Xiao, J., et al.: Probase: a universal knowledge base for semantic search. Microsoft Research Asia (2010)
39. Xie, R., Liu, Z., Lin, F., Lin, L.: Does william shakespeare really write hamlet? knowledge representation learning with confidence. In: AAAI (2018)

40. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. ICLR (2015)
41. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: KDD. pp. 353–362 (2016)
42. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR) **52**(1), 1–38 (2019)