Article:

Ge, Y., Ma, J., Zhang, L. et al. (2023) Trustworthiness-aware knowledge graph representation for recommendation. Knowledge-Based Systems, 278. 110865. ISSN: 0950-7051

https://doi.org/10.1016/j.knosys.2023.110865

Knowledge-Based Systems 278 (2023) 110865

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Trustworthiness-aware knowledge graph representation for recommendation

Yan Ge [a,*], Jun Ma [b], Li Zhang [c], Xiang Li [a], Haiping Lu [d]

[a] Department of Computer Science, The University of Bristol, Woodland Road, Bristol, BS8 1UB, England, United Kingdom
[b] Amazon.com, Inc., 440 Terry Ave N, Seattle, WA 98109, United States
[c] Oxford-Man Institute, The University of Oxford, Walton Well Road, Oxford, OX1 2JD, England, United Kingdom
[d] Department of Computer Science, The University of Sheffield, 211 Portobello, Sheffield, England, S1 4DP, United Kingdom

ARTICLE INFO

Article history:
Received 15 November 2022
Received in revised form 25 May 2023
Accepted 27 July 2023
Available online 5 August 2023

Keywords:
Recommender systems
Knowledge graph representation
Trustworthiness

ABSTRACT

Incorporating knowledge graphs (KGs) into recommender systems (RS) has recently attracted increasing attention. For large-scale KGs, due to limited labour supervision, noises are inevitably introduced during automatic construction. However, the effects of such noises as untrustworthy information in KGs on RS are unclear, and how to retain RS performing well while encountering such untrustworthy information has yet to be solved. Motivated by them, we study the effects of the trustworthiness of the KG on RS and propose a novel method **trust**worthiness-aware knowledge graph representation (KGR) for **rec**ommendation (TrustRec). TrustRec introduces a trustworthiness estimator into noise-tolerant KGR methods for collaborative filtering. Specifically, to assign trustworthiness, we leverage internal structures of KGs from microscopic to macroscopic levels: motifs, communities and global information, to reflect the true degree of triple expression. Building on this estimator, we then propose trustworthiness integration to learn noise-tolerant KGR and item representations for RS. We conduct extensive experiments to show the superior performance of TrustRec over state-of-the-art recommendation methods.

1. Introduction

Recommender systems (RS) aim to ease information explosion and largely reduce users' effort in finding items of interest. Conventional RS, based on collaborative filtering (CF) [1,2], usually suffer from the sparsity of interactions and the cold-start problem. To address these issues, existing works incorporate auxiliary sources as side information, such as social networks [3].

Knowledge graphs (KGs) as one type of auxiliary source contain rich facts about items in the form of heterogeneous graphs [4], which are successfully applied to many applications such as question answering [5] and text classification [6,7]. Facts in KGs are presented in the form of triples (*head entity, relation, tail entity*). For example, *(Tom Hanks, IsActorOf, Forrest Gump)* indicates that Tom Hanks is an actor in Forrest Gump. However, some untrustworthy information[1] are inevitably introduced when constructing KGs.

Inspired by the success of applying KGs in a variety of tasks, some recent works incorporate KGs into RS via knowledge graph representation (KGR) that aims to learn low-dimensional distributed embedding of entities and relations [8–11]. The usage of KGs within the context of RS can alleviate the item cold-start and sparsity problem of CF. The reason is twofold: (1) KGs introduce extra semantic connections among items, which can provide new items with more interactions to recommendations; (2) KGs consist of a variety of relation types, which helps extend a user's interests reasonably. Therefore, collaborative knowledge-based embedding (CKE) [12] combines CF with KG embedding in a unified Bayesian framework. Knowledge translation-based user preference model [13] transfers relation information from a KG to recommendations for a better understanding the reasons that a user likes an item. Knowledge-aware graph neural network (GNN) with label smoothness regularisation [14] applies GNN architecture to KGs by using a user-specific relation score function and aggregating neighbourhood information with different weights.

However, when incorporating KGs into RS, most existing methods, including the above three state-of-the-art (SOTA) methods, do not consider noises in KGs. In real-world KGs, some noises are inevitably introduced in the process of automatically constructing large-scale KGs due to limited labour supervision [15,16]. For example, to construct KGs, the recent model [17] achieves only around 60% precision when the recall is 20%, which

* Corresponding author.
E-mail address: yan.ge@bristol.ac.uk (Y. Ge).

[1] In this paper, the untrustworthy information is equivalent to the noise in KGs.

https://doi.org/10.1016/j.knosys.2023.110865

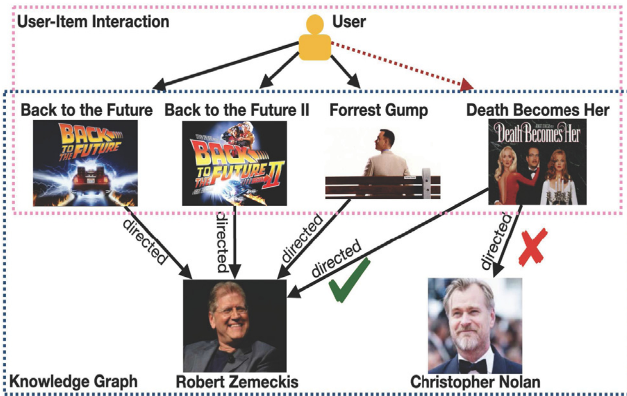

Fig. 1. An example to show that a noisy triple (*Death Becomes Her, IsDirectedBy, Christopher Nolan*) can degrade the recommendation performance (i.e., recommend interesting *Death Becomes Her* to the user). It motivates our model to tolerate such noisy triples.

indicates a large amount of untrustworthy information in KGs. We argue that such noises in KGs as auxiliary data can degrade the performance of RS, which will be verified by our experiments. As illustrated in Fig. 1, the red dashed arrow indicates an interaction to be predicted between a user and a movie *Death Becomes Her*. Assuming that this user has interacted with three similar movies *Back to The Future I & II* and *Forrest Gump* due to the same director *Robert Zemeckis* by using the KG. The correctness of the director of *Death Becomes Her* can determine whether to recommend it to this user. In this case, we fail to recommend the *Death Becomes Her* if a noisy triple (*Death Becomes Her, IsDirectedBy, Christopher Nolan*) exists. Therefore, it is essential to tolerate such noisy triples in the KG incorporated with RS. The other relation type (e.g., 'genre') can affect the recommendations, which shares the same idea about how the 'director' relation affects recommendations. Therefore, we show a single relation 'director' as a representative in this example, and it can be applied to another relation type, such as 'starred' and 'genre'.

In this paper, we aim to estimate noises in KGs, while constructing noise-tolerant KGR to incorporate with RS. However, there remain two challenges: (1) how to estimate noises in **arbitrary** KGs without collecting external information. Some works [18,19] strongly rely on external information (e.g., web content, text) but do not have good generalisation to estimate noises in KGs. (2) Noise estimation integration. Some existing works [12,13,20–22] study an integration between two modules (KGR and RS) through, for example, linearly combining the entity and the corresponding item embeddings. However, building on this two-module integration, introducing another noise estimation module is still unclear.

To address the above challenges, we propose a novel method **trust**worthiness-aware KGR for **rec**ommendations (TrustRec). TrustRec incorporates noise-tolerant translation-based KGR into a CF-based method through a trustworthiness estimator, which gives the degree of certainty of triples. Specifically, to construct this trustworthiness estimator, we firstly leverage internal structural information in KGs from microscopic to macroscopic levels: the motif (co-occurrence in the same type of local connectivity pattern), communities (co-occurrence in the same high association group) and global information (correlation strength on all paths). Then we use a neural network architecture to fuse the structural information, and finally yield a trustworthiness value for every triple. In this way, we can estimate triple trustworthiness in any KG by leveraging internal information to enhance generalisation capacity, which tackles the first challenge. To address the second challenge, building on our estimator, we integrate triple trustworthiness into a proposed neural/weighted pairwise ranking loss functions for noise-tolerant KGR. Meanwhile, we integrate entity trustworthiness as a linear combination ratio of an entity embedding to learn a noise-tolerant item representation for RS. We summarise our contributions as follows:

1. We investigate the effect of untrustworthy information in KGs on recommendations and find that the untrustworthy information degrades the performance of RS.
2. We propose the TrustRec that is trustworthiness-aware RS to learn noise-tolerant KGR and item representations for RS, which retains RS performing well while encountering noises in KGs.
3. We conduct extensive experiments to show the superior performance of our TrustRec over SOTA methods.

2. Preliminary

2.1. Notations

We denote scalars by lowercase italic letters, e.g., a, vectors by lowercase boldface letters, e.g., $\mathbf{a}$, and matrices by uppercase boldface, e.g., $\mathbf{A}$.

We have a KG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$, which is comprised of massive entity–relation–entity triples (e_h, r, e_t), where $e_h \in \mathcal{E}, r \in \mathcal{R}$, $e_h \in \mathcal{E}$ denote the head, relation, tail. We construct a weighted directed graph G from a KG $\mathcal{G}$. Each entity $e \in \mathcal{E}$ is abstracted into a node. If there are relations from the entities e_1 to e_2, a directed edge will exist from node e_1 to e_2, and the weight of the edge is the number of relations. Therefore, a KG with n entities can be mapped as a directed graph G with n nodes. For RS, the user–item interaction matrix $\mathbf{Y}$ is defined according to users' implicit feedback.

2.2. Trustworthiness in KG

Most traditional knowledge graph construction methods usually involve huge human supervision or expert annotation, which are extremely labour-intensive and time-consuming [15]. Recently, large-scale knowledge graphs (e.g., DBpedia [23], Freebase [24]) are productively and automatically constructed from unstructured web text (e.g., NELL [25]). However, some noises and errors are inevitably introduced in the process of automation due to limited labour supervision [26,27].

Existing KG-based tasks (e.g., knowledge completion [19]) or applications (e.g., question answering [28]) assume knowledge in the existing KG is completely correct. To model errors in KGs, Xie et al. [15] proposed a triple confidence awareness knowledge representation learning framework, which detects possible noises in KGs while learning knowledge representations with confidence simultaneously. They introduced triple confidence to conventional translation-based methods for knowledge representation learning. Jia et al. [16] synthetically extracted the 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. [29] proposed a KGs embedding model on uncertain KGs that associate every fact with a confidence score. Dong et al. [18] built a large-scale uncertain knowledge graph, and fused multiple extraction sources with prior knowledge derived from an existing knowledge base. Focusing on rule-based learning, PTrustE [30] integrates a probability logic model, based on correlations, with a knowledge graph representation learning approach, which focuses on the paths of triples. This combination enhances the model's generalisation capability by learning both the matrix of path scores and the trustworthiness of different paths. TKGC [31] operates on data from multiple, noisy sources for the trustworthy completion of knowledge graphs. It incorporates a comprehensive scoring function, which assesses the credibility of both relational and literal facts, regardless of their value types. TrustE [32] employs an innovative structured embedding method specifically designed for different types of entities. Furthermore, it leverages an energy function conscious of trustworthiness, allowing it to establish robust embeddings for various entity types, even in the context of knowledge graphs that contain a lot of noise.

Some works [18,19,33] need to collect external information (e.g., web content, text) to measure the trustworthiness of triples in a KG. To enhance the flexibility and generalisation, the methods [15,16] only rely on internal information to construct reachable paths for trustworthiness estimation. The number of paths is enormous in a large-scale KG. Therefore, instead of constructing paths, our trustworthiness estimator leverage structural information (e.g., motif, community, global) to measure noises in arbitrary KGs without collecting external information.

The definition of untrustworthiness can vary depending on the task and the context of knowledge graphs (KGs). In this paper, we focus on a context of KGs for recommendations and follow the paper [16,29], we formally define the untrustworthy triples below.

Definition 1 (*Untrustworthy Triple*)**.** We have a KG ($\mathcal{G}$) that consists of massive weighted triples $\mathcal{G} = \{e_l, w_l\}$ where w_l reflects a credibility score of the triple l. A triple e_l is an untrustworthy triple if $w_l < c$, where c is a credibility threshold.

2.3. KGR

KGR is used to embed entities and relations into low-dimensional vectors while preserving semantic and structural information [34]. Translational models are popular to exploit distance-based energy functions and a relation is regarded as a translation in the embedding space. TransE [35] follows an assumption that e_h and e_t are connected by r with a low error if a triple (e_h, r, e_t) holds, and thus formulates an energy function

$$g_E = \|\mathbf{e}_h + \mathbf{r} - \mathbf{e}_t\| \tag{1}$$

However, TransE has flaws when dealing with 1-to-N, N-to −1 and N-to-N relations. To address these issues, TransH [36] introduces relation specific hyperplanes, which each relation r as a vector $\mathbf{r}$ on a hyperplane with $\mathbf{w}_r$. The embeddings $\mathbf{e}_h$ and $\mathbf{e}_t$ are first projected to the hyperplane of relation r to obtain vectors

$$\mathbf{e}_h^{\perp} = \mathbf{e}_h - \mathbf{w}_r^{\perp}\mathbf{e}_h\mathbf{w}_r \ , \ \mathbf{e}_t^{\perp} = \mathbf{e}_t - \mathbf{w}_r^{\perp}\mathbf{e}_t\mathbf{w}_r \tag{2}$$

and then $\mathbf{e}_h^{\perp} + \mathbf{r} \approx \mathbf{e}_t^{\perp}$. For TransE and TransH, the embeddings of entities and relations are in the same space. However, entities and relations are different types of objects. It is insufficient to model them in the same space. To address this issue, In TransR [37], e_h and e_t are projected to a new space so that the relation r focuses on through a matrix $\mathbf{M}_r$ and then

$$g_R = \|\mathbf{M}_r\mathbf{e}_h + \mathbf{r} - \mathbf{M}_r\mathbf{e}_t\|. \tag{3}$$

TransD [38] constructs dynamic mapping matrices

$$\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p + \mathbf{I} \ , \ \mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p + \mathbf{I} \tag{4}$$

by the projection vectors $\mathbf{h}_p, \mathbf{t}_p, \mathbf{r}_p \in \mathbb{R}^n$ and an identity matrix $\mathbf{I} \in \mathbb{R}^{n\times n}$, with the formulation as

$$g_D = \| \left(\mathbf{r}_p\mathbf{h}_p^{\top} + \mathbf{I}\right)\mathbf{h} + \mathbf{r} - \left(\mathbf{r}_p\mathbf{t}_p^{\top} + \mathbf{I}\right)\mathbf{t}\|. \tag{5}$$

2.4. Knowledge-aware recommendation

The existing methods on integrating the KG into recommendations can be roughly categorised into embedding-based and path-based methods. Path-based methods encode the connection pattern of user–item pair or item–item pair into latent vectors. For example, to capture the semantics of different paths and distinctive saliency of user preferences, Zhu et al. [39] use recurrent networks to encode different paths. It further determines different path saliency through a pooling operation to generate recommendations. To provide explainability, knowledge-aware path recurrent network [40] generates path representations by composing the semantics of both entities and relations to infer the underlying rationale of a user–item interaction. However, for path-based methods, the number of possible paths in a large-scale knowledge graph can grow countless. Therefore, it may hinder the performance of the recommendation.

Embedding-based methods leverage fruitful triple facts in the KG to enrich the representation of items or users. Knowledge graph attention network for recommendation [41] models the high-order connectivities in KGs and then recursively propagates the node embeddings. To learn both the KG embedding task and the recommendation task, Wang et al. [42] propose a multi-task learning approach. This approach designs a cross&compress unit to associate the two tasks, which can automatically learn high-order interactions [43] of item and entity features and transfer knowledge between the two tasks. Cao et al. [13] model various implicit relations between users and items and transfer knowledge learned from TransH, which reveals the users' preferences in consuming items.

2.5. Matrix Factorisation (MF)

To build RS, CF models users' preferences for items based on historical interactions. MF [1] is a popular technique. It is formulated as:

$$\hat{y}_{ui} = \mathbf{p}_u^T\mathbf{q}_i, \tag{6}$$

where $\mathbf{p}_u$ and $\mathbf{q}_i$ denote the latent vector for user u and item i. Bayesian personalised ranking (BPR) optimises the above equation with a pairwise ranking loss [44]

$$\mathcal{L}_r = \sum_{(u,i)\in\mathbf{Y},(u,i')\in\mathbf{Y}'} -\log\delta(\hat{y}_{ui} - \hat{y}_{ui'}), \tag{7}$$

where $\delta(\cdot)$ is a sigmoid function and $\mathbf{Y}'$ contains negative interactions by randomly corrupting an interacted item to a non-interacted one for each user.

3. Methodology

In this section, we propose to estimate trustworthiness of triples through internal structural information: motifs, communities and global information. We then integrate triple trustworthiness into a weighted/neural loss function of KGR to learn noise-tolerant KGR. Meanwhile we integrate entity trustworthiness into RS to learn noise-tolerant item representations for RS.

3.1. Motif-aware trustworthiness

Motifs are fundamental subgraph patterns in graphs and show complex local connectivity patterns beyond a direct relation between nodes [45]. Triangular Motifs (shown in left bottom of Fig. 3) demonstrate very important local structures underlying various complex networks, such as social networks.

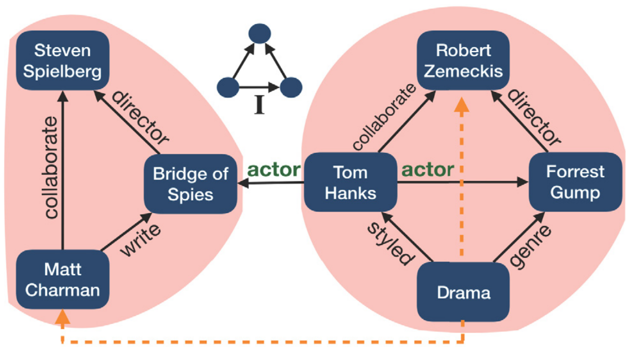

Fig. 2. An example to show (1) a strong tie occurs in the triple *(Tom Hanks, actor, Forrest Gump)* through the motif I; (2) high association in the triple *(Tom Hanks, actor, Forrest Gump)* within a community (shadow); (3) high correlation (an orange dashed arrow) occurs in an entity pair (*Drama*, *Robert Zemeckis*) through all paths.

We use the strength of a tie between head and tail linked by a relation to measure the trustworthiness of triple (e_h, r, e_t). If head e_h and tail e_t have a strong tie, the relation r between head e_h and tail e_t is expected to be strong. Motif modelling is an effective approach to measure the strength of a tie between two entities [45,46]. 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 motif connectivity pattern. Intuitively, if two people have more common friends, the stronger strength of a tie between them can occur. Additionally, considering motifs can capture the rich context of relations to diversify strengthen of ties while direct edges relation cannot. For example, in Fig. 2, if only considering the simple edge relation, triples *(Tom Hanks, actor, Bridge of Spies)* and *(Tom Hanks, actor, Forrest Gump)* have the same strength of a tie. However, when considering a motif type I in Fig. 2, the triple *(Tom Hanks, actor, Forrest Gump)* has a rich context (e.g., with *Robert Zemeckis*) to enhance its strength of a tie. This paper will focus on all triangular motifs as shown in Fig. 3, though our proposed method can be easily extended to other motifs.

Based on the above analysis, we take the input (e_h, r, e_t) from G, and quantify the strength of a tie for it by counting the number of the motif types $\mathcal{M}_i$ containing this triple. Different type of triangular motifs reflect different connectivity patterns. Thus, we construct a feature vector $\mathbf{m}(e_h, r, e_t)$ to consider all, and the ith entry in $\mathbf{m}(e_h, r, e_t)$ are decided by:

$$\mathbf{m}_i(e_h, r, e_t) = \sum_{e_h, e_t \in \mathcal{E}, r \in \mathcal{R}} \mathbb{1}(e_h, r, e_t \text{ occur in } \mathcal{M}_i) \tag{8}$$

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 $\mathbf{m}(e_h, r, e_t)$ where the ith element indicates the number of motif type $\mathcal{M}_i$ containing (e_h, r, e_t). We then compress the motif feature vector $\mathbf{m}(e_h, r, e_t)$ into a value $m(e_h, r, e_t)$ by a trainable weight $\mathbf{w}_m$ as

$$m(e_h, r, e_t) = \mathbf{m}(e_h, r, e_t) \cdot \delta(\mathbf{w}_m^T). \tag{9}$$

We interpret the value $\mathbf{w}_m(i)$ as the importance of motif type $\mathcal{M}_i$. Note that all triples in the KG share the same trainable weight $\mathbf{w}_m$ to largely avoid the increase of the model complexity with the increase of KG size.

For an input triple (e_h, r, e_t), we quantify all 13 different triangular motif types (shown in left bottom of Fig. 3) in a directed graph involving this triple by matrix operation. For example, for the triangular motif with three bidirectional edges, we use $\mathbf{A} \cdot \mathbf{A} \circ \mathbf{A}$

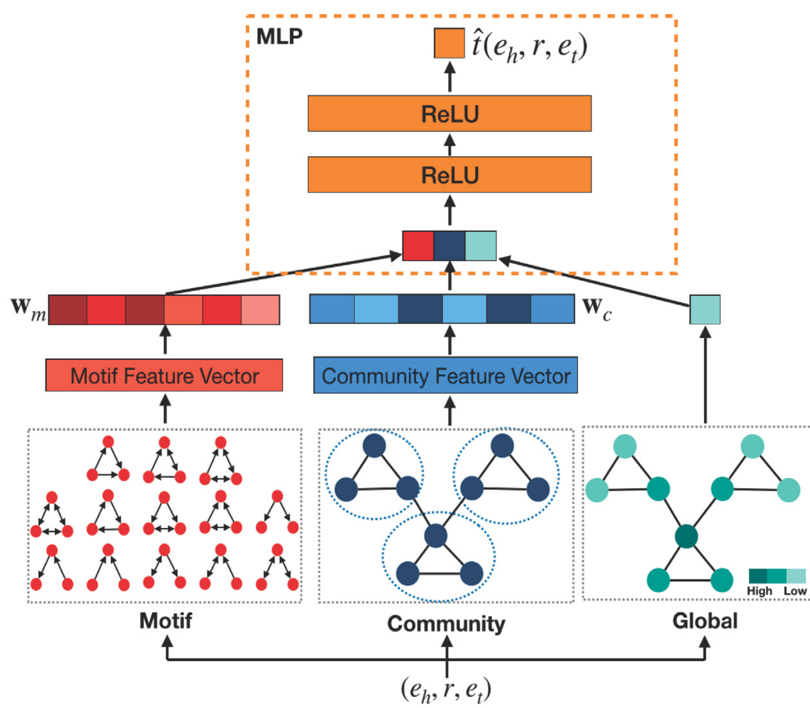

Fig. 3. The framework of the proposed trustworthiness estimator by leveraging internal structure information of KGs: motifs, communities and global information. We utilise internal structural information in KGs, ranging from microscopic to macroscopic levels: motifs, communities, and global information. We extract a motif-aware feature vector indicating the number of motif types involving the input triple. The entry in the community-aware feature vector represents whether the input triple is in the same community. Furthermore, we define the correlation strength for triples by considering all paths throughout the entire graph. We then employ a neural network architecture to integrate the structural information, ultimately producing a trustworthiness value for each triple.

to quantify any triple involving this motif, where $\mathbf{A}$ is the edge adjacency matrix, '$\cdot$' is matrix multiplication, and '$\circ$' is Hadamard product. We set a trainable weight vector $\mathbf{w}_m$ to quantify the influence of different triangular motifs. We consider the value $\mathbf{w}_m(i)$ to represent the significance of motif type $\mathcal{M}_i$. It is important to note that all triples in the KG share the same trainable weight $\mathbf{w}_m$, which effectively prevents the model's complexity from increasing as the KG size grows.

3.2. Community-aware trustworthiness

The motif-aware trustworthiness estimator based on the local neighbours is straightforward but cannot take fully advantage of rich structural information of KGs. To capture a more complete picture of triples, we consider a community structure that consists of a group of entities. Community structure refers to the occurrence of groups of nodes in a graph that are more densely connected than the rest of the graph. Some existing works [47,48] show that entities within a community have relatively higher association than entities in different communities. If head e_h and tail e_t have a higher association, head e_h and tail e_t are more likely to have a trusted relation. For example, in Fig. 2, the same relation *actor* connecting an intra-community entity pair *(Tom Hanks, Forrest Gump)* is more trustful than it in an inter-community entity pair *(Tom Hanks, Bridge of Spies)*.

Inspired by the above, we thus perform community detection task on G. Since our focus is the association of triples, we first convert all directed edges in G to undirected ones and form a graph G_u. For the graph G_u, we then use a spectral clustering (SC) [49] method to cluster G_u into k communities $\mathbb{S} = \{S_1, \ldots, S_k\}$. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be an adjacency matrix of weighted graph G_u where the entry $\mathbf{A}(i,j)$ is the number of relations between e_i and e_j. The degree matrix $\mathbf{D}$ is a diagonal matrix with diagonal entries

$$\mathbf{D}(i,i) = \sum_{j=1}^{n} \mathbf{A}(i,j), \tag{10}$$

where $\mathbf{D}(i,i)$ is the degree of the entity e_i. We then construct a Laplacian matrix $\mathbf{L}$ as follows:

$$\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}, \tag{11}$$

where $\mathbf{I}_n$ is an identity matrix. SC aims to learn a spectral embedding $\mathbf{Z} \in \mathbb{R}^{n\times k}$ by optimising a function as follows:

$$\min_{\mathbf{Z}} \operatorname{tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Y}\right), \ \text{s.t. } \mathbf{Z}^T\mathbf{Z} = \mathbf{I}, \tag{12}$$

where $\operatorname{tr}(\cdot)$ is the trace function. The above function can be solved by eigenvalue decomposition of $\mathbf{L}$, i.e., $\mathbf{Z} = [\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_k}]$ are the eigenvectors corresponding to the smallest k eigenvalues of $\mathbf{L}$. To find clusters, SC then uses $\mathbf{Z}$ as an input to perform k-means.

The changing number of communities can determine the state that whether a triple (e_h, r, e_t) are in the same community. Our model thus contains multiple states to represent community-aware trustworthiness by constructing a community indicator vector $\mathbf{c}$. The ith entry in $\mathbf{c}(e_h, r, e_t)$ is determined by:

$$\mathbf{c}_i^j(e_h, r, e_t) = \begin{cases} 1 & (e_h, r, e_t) \in S_n \text{ with } j \text{ partitions}, \\ 0 & \text{otherwise}, \end{cases} \tag{13}$$

where $1 \leqslant j \leqslant k$. From the Eq. (13), instead of using a single defined number of communities, we use the number of the community from 10 to 100 with an interval of 10. Therefore, instead of using only one community information, we use many by constructing a community-aware feature vector that indicates whether the head and tail entity are in the same community or not. Consistent with compression operation in motif-aware trustworthiness, we have a community-aware trustworthiness

$$c(e_h, r, e_t) = \mathbf{c}(e_h, r, e_t) \cdot \delta(\mathbf{w}_c^T), \tag{14}$$

where $\mathbf{w}_c$ is a shared trainable weight to indicate the importance of the number of communities.

Since community-aware trustworthiness relies on the communities identified within a network, the choice of the community detection method can theoretically impact the results. In this paper, we use spectral clustering (SC) to identify communities because SC is less sensitive to noise and outliers than some other clustering algorithms. By transforming the data into a lower-dimensional space using eigenvectors, spectral clustering can reduce the influence of noise and make the clustering process more robust [49]. It contributes to our TrustRec that aims to retain recommendation performing well while encountering such noise in the KG.

3.3. Global-structure-aware trustworthiness

Motif-aware and community-aware estimators mainly focus on microscopic and mesoscopic structural information. The global structure, one important macroscopic description of the graph structure, is a complementary component to represent the trustworthiness of triples in KGs. Therefore, to consider the global structure, we introduce the concept of correlation strength that captures how difficult to reach a tail entity e_t from a head entity e_h through a sequence of relations in a ***whole*** graph. For example, in Fig. 2, there are dense paths from *Drama* to *Robert Zemeckis* (e.g., *Drama* $\rightarrow$ *Tom Hanks* $\rightarrow$ *Robert Zemeckis*), that is, there is a high correlation between them. By contrast, it is impossible to research from *Drama* to *Matt Charman* following all paths in the graph.

To instantiate the above idea, we adopt source allocation theory in PageRank [50] to characterise the correlation strength for triples. We assume that the trustworthiness between entity pairs (e_h, e_t) will be higher, and more resource is passed from the head e_h through all paths to the tail e_t in a whole graph G. The amount of resource aggregated into e_t indicates the trustworthiness between e_h and e_t. Specifically, starting from e_h each node in the graph should be reached. In the initial state, the resource amount of e_h is 1, and all others are 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 $p(e_t|e_h)$, it is calculated as follows:

$$p(e_t|e_h) = (1-\alpha)\sum_{e_i\in\mathcal{D}} \frac{p(e_i \mid e_h)\cdot w_{e_i t}}{d(e_i)} + \frac{\alpha}{n}, \tag{15}$$

where $\mathcal{D}$ is a set of entities that have outgoing links to the entity e_t, $w_{e_i t}$ is the weight from the e_i to e_t, $d(e_i)$ is the out-degree of the entity e_i. Thus, for each entity e_i in $\mathcal{D}$, the resource flows from e_i to e_t should be $\frac{p(e_i|e_h)\cdot w_{e_i t}}{d(e_i)}$. The entities without outgoing links can cause the absorption of the resource. To prevent it, resource flow from each entity may directly jump to a random entity with the same probability α. This part of the resource that flows to e_t randomly is $\frac{1}{n}$.

3.4. Fusion of estimators

We use a neural network structure multi-layer perceptron (MLP) to extract a final trustworthiness from three estimators. Note that the way of extraction is not limited to MLP, and we can use a more elaborate design of the neural network. For the triple (e_h, r, e_t), we first concatenate the above three-level trustworthiness

$$\mathbf{x}(e_h, r, e_t) = [m(e_h, r, e_t), c(e_h, r, e_t), p(e_t|e_h)]. \tag{16}$$

The vector $\mathbf{x}(e_h, r, e_t)$ will be input into the MLP and transformed passing L hidden layers as follows:

$$\hat{t}(e_h, r, e_t) = \mathcal{M}(\mathcal{M}(\cdots \mathcal{M}(\mathbf{x}(e_h, r, e_t)))) \\ = \mathcal{M}^L(\mathbf{x}(e_h, r, e_t)), \quad (17)$$

where $\mathcal{M}(\mathbf{x}) = \sigma(\mathbf{W}_m\mathbf{x} + \mathbf{b}_m)$ is a fully-connected neural network layer with weight $\mathbf{W}_m$, bias $\mathbf{b}_m$, and nonlinear ReLU activation function $\sigma(\cdot)$. In the output layer of $\mathcal{M}^L(\cdot)$, we use a sigmoid function $\delta(\cdot)$ to ensure the returned $\hat{t}(e_h, r, e_t)$ in the range 0 to 1. The whole framework of the trustworthiness estimator is shown in Fig. 3.

3.5. Trustworthiness integration

After obtaining trustworthiness of triples, TrustRec follows the conventional translation-based KGR to incorporate with CF. To inject auxiliary information from KG to RS, some existing works study the integration between two modules KGR and RS (e.g., linearly combine the entity and the corresponding item embeddings). When considering an additional trustworthiness estimator module, we propose trustworthiness integration with both KGR and RS. Specifically, we propose triple trustworthiness integration to learn noise-tolerant KGR, and entity trustworthiness integration to learn noise-tolerant item representations of RS. For the triple trustworthiness integration, we propose a weighted and a neural margin-based ranking loss (MRL) of KGR.

Weighted MRL. The idea is that a triple with higher trustworthiness should be more important when training KGR. Based on it, we construct a weighted MRL as below

$$\mathcal{L}_k^{(w)} = \sum_{\substack{(e_h,r,e_t)\in\mathcal{G} \\ (e'_h,r,e'_t)\in\mathcal{G}^-}} \hat{t}(e_h, r, e_t) \cdot [\gamma + g_R(e_h, r, e_t) - g_R(e'_h, r, e'_t)]_+, \quad (18)$$

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, and $g_R(\cdot)$ is the energy function of TransR. We choose TransR because TrustRec is equivalent with CKE if our trustworthiness estimator is neglected, which can gain insights about the effect of our estimator. To learn noise-tolerant KGR, trustworthiness $\hat{t}(e_h, r, e_t)$ instructs our model to pay more attention on those more trustful triples.

Neural MRL. The idea is that if $\hat{t}(e_h, r, e_t)$ is involved in a parameterised way to determine the score of the energy function, TrustRec itself will learn to integrate trustworthiness for noise-tolerant KGR. For example, if $\hat{t}(e_h, r, e_t)$ negligibly contributes to the energy score of (e_h, r, e_t), TrustRec can assign very low trustworthiness to it. Thus, we first perform a concatenation operation

$$\mathbf{n}(e_h, r, e_t) = [\hat{t}(e_h, r, e_t), g_D(e_h, r, e_t)]. \quad (19)$$

We then construct a neural MRL as below:

$$\mathcal{L}_k^{(n)} = \sum_{\substack{(e_h,r,e_t)\in\mathcal{G} \\ (e'_h,r,e'_t)\in\mathcal{G}^-}} \left[\gamma + \mathcal{N}^L(\mathbf{n}(e_h, r, e_t)) - g_D(e'_h, r, e'_t)\right]_+, \quad (20)$$

where $\mathcal{N}(\mathbf{x}) = \sigma(\mathbf{W}_n\mathbf{x} + \mathbf{b}_n)$. Here, we use the energy function of TransD because of a consideration of different types of entities in KGs and a study of the diverse KGR methods on TrustRec.

Integration with RS. Some existing works linearly combine the entity and corresponding item embedding as the final item embedding as $\mathbf{i}' = \mathbf{e} + \mathbf{q}_i$. However, the final item embedding $\mathbf{i}'$ contains noises from knowledge. Therefore, to learn noise-tolerant item representations of RS, we assume that if an entity is likely to be involved in triples with high trustworthiness, this entity has high combination ratio to form $\mathbf{i}'$. We propose entity trustworthiness that is an averaged summation of the triple trustworthiness it involves. It is formulated as below:

$$\hat{t}(e) = \frac{\sum_{e'_t\in\mathcal{E},r'\in\mathcal{R}} \hat{t}(e, r', e'_t)}{n_h} + \frac{\sum_{e'_h\in\mathcal{E},r'\in\mathcal{R}} \hat{t}(e'_h, r', e)}{n_t}, \quad (21)$$

where n_h and n_t are the number of triples that the entity e acts as heads and tails. TrustRec treats $\hat{t}(e)$ as an integration ratio of entity e, and formulates

$$\mathbf{q}'_i = \hat{t}(e) \cdot \mathbf{e} + \mathbf{q}_i, \quad (22)$$

where $\mathbf{q}_i$ is a learned latent vector of item i by MF. We then develop two variants of TrustRec depending on the overall loss. TrustRec(W) uses the overall loss

$$\mathcal{L}^{(w)} = \mathcal{L}_k^{(w)} + \mathcal{L}_r, \quad (23)$$

while TrustRec(N) uses the overall loss

$$\mathcal{L}^{(n)} = \mathcal{L}_k^{(n)} + \mathcal{L}_r. \quad (24)$$

In addition, compared to our previous work [51], this paper has two main advancements: non-linearity modelling and experimental verification. To capture the non-linearity of noises, we propose to fuse estimators with fully-connected layers and a neural margin-based ranking loss. Moreover, in this paper, we conduct comprehensive experiments to demonstrate the superior performance of TrustRec in comparison to state-of-the-art recommendation methods using real-world datasets. In contrast, our previous paper [51] does not consider the non-linearity of noises and does not conduct experimental results and analysis.

4. Experiments

In this section, we conduct extensive experiments with the aim of answering the following research questions:

- **RQ1:** How do noisy triples in KGs affect the performance of KG-aware RS?
- **RQ2:** For recommendation, how does TrustRec perform compared with SOTA KG-aware recommendation methods?
- **RQ3:** For KG completion, can TrustRec show superior performance over SOTA KG-aware RS that can learn KGR?

4.1. Dataset

We use two public datasets in the book and movie domains: DBbook2014,[2] MovieLens-1M.[3] Items in these two domains are mapped into DBPedia entities if there is a mapping available, which is released by the paper [13]. Table 1 shows the statistics of datasets.

Following most item recommendation works that models implicit feedback, we treat existing ratings as positive interactions,

Table 1
Statistics of DBbook2014 and MovieLens-1M.

		DBbook2014	MovieLens-1M
Rec	# Users	5576	6040
	# Item	2680	3240
	# Ratings	65,961	998,539
	# Avg. ratings	12	165
	# Completeness	0.4%	5.1%
KG	# Entity	13,882	14,708
	# Relation	13	20
	# Triple	334,511	434,189

[2] http://2014.eswc-conferences.org/important-dates.html.
[3] https://grouplens.org/datasets/movielens/1m/.

and generate negative ones by randomly corrupting items. To study the effect of noisy triples, we generate noisy triples to be 10%, 20%, 30% and 100% of existing triples by the following protocol: for an existing triple (e_h, r, e_t) in a training set, we generate a corresponding noisy one by randomly replacing its head (e_h', r, e_t) or tail (e_h, r, e_t') while ensuring that (1) it cannot be found in the existing KG; (2) it contains at least one item; (3) for noise injection, triple (e_h, r, e_t) is replaced with (e_h', r, e_t) (or (e_h, r, e_t')) to ensure the total number of triples is unchanged. To validate the effect of KG-aware methods, we consider an item cold-start and sparsity scenarios. For both datasets, 25% of items in valid and test sets cannot be found in the train set. We randomly sparsify 70% interactions of MovieLens-1M since from Table 1 its completeness is more than ten times than the completeness of DBBook2014.

4.2. Baselines

We compare TrustRec with the following six SOTA RS methods:

1. Collaborative Filtering with Knowledge Graph (CFKG)[4] [52]: This method constructs an user–item KG and the relation is decided by user behaviours (i.e., review, brand, category, bought-together). This KG will be combined with the item-side KG by shared common items. It then uses translational recommendation to minimise the loss.
2. Collaborative Knowledge Embedding (CKE) [12]: This approach applies matrix-factorisation-based CF to knowledge-base embedding for recommendation, which uses TransR to learn entity and relation embedding.
3. Knowledge Co-Knowledge factorisation model (CoFM) [53]: It studies the effect of knowledge transfer between item recommendations and KG completion via a co-factorisation model which can be seen as a transfer learning model.
4. Knowledge Translation-based User Preference model (KTUP) [13]: KTUP models various implicit relations between users and items and transfer knowledge learned from TransH, which reveals the preferences of users on consuming items. Additionally, it provides explainability via aligned relations and preferences.
5. Knowledge Graph Convolutional Networks (KGCN)[5] [14]: It extends the GCN to the KG by aggregating neighbourhood information selectively and biasedly, which simultaneously learns both structural information and semantic information from the KG as well as users' preferences and potential interests.
6. Knowledge-aware Graph Neural Networks (GNN) with Label Smoothness regularisation (KGNN-LS)[6] [14]: This approach incorporates GNN architecture into KGs after converting KGs to weighted homogeneous graphs. This conversion uses a user-specific relation scoring functions and then aggregates neighbourhood information with different weights. In addition, KGNN-LS proposes label smoothness constraint to provide strong regularisation for learning the edge weights in KGs.
7. Multi-task feature learning approach for knowledge graph enhanced recommendation (MKR) [42]. MKR is a deep end-to-end framework that employs knowledge graph embedding tasks to aid recommendation tasks. This framework has the ability to autonomously share hidden features and acquire advanced interactions between items in recommender systems and entities present in the knowledge graph.

[4] https://github.com/TaoMiner/joint-kg-recommender.
[5] https://github.com/hwwang55/KGCN.
[6] https://github.com/hwwang55/KGNN-LS.

4.3. Training details

We construct the training set, validation set and testing set by randomly splitting the dataset with the ratio of 7 : 1 : 2. Each experiment is repeated five times, and the average performance is reported. For hyperparameters, the learning rate of all methods is searched in {0.0005, 0.001, 0.005, 0.01}, the embedding size in {16, 32, 64}, We use an open-source PyTorch library to study all methods under the same software framework released by [13]. All trainable parameters are optimised by Adam algorithm. The coefficient of L_2 regularisation is 10^{-5}. The batch size is 512. We perform early stopping strategy on validation sets. All other hyperparameters use default settings. At the beginning of training, we assume all triples are correct, and initialise the triple trustworthiness as 1. All experiments were performed on a Linux machine with 2.4 GHz Intel Core and 8G memory. The code for TrustRec has been uploaded to Supplementary Material.

4.4. Evaluation metrics

In recommendation, we use the trained model to select K items with highest predicted click probability for each user in the test set, and choose $F1@K$, $Precision@K(P@K)$ and $Recall@K$. For KG completion, we use $Hit\ ratio@K$.

1. Hit ratio@K: It is 1 if a correct items are recommended within the top K items, otherwise 0. We compute the mean of all users as the final hit ratio score.
2. F1-score@K: It is the combination mean of precision at rank K and recall at rank K.
3. Precision@K: It is the fraction of the items recommended that are relevant to the user. We compute the mean of all users as the final precision.
4. Recall@K: It is the proportion of the items relevant to the user that have been successfully recommended. We compute the mean of all users as the final recall.

4.5. Effect of noisy triples (RQ1)

Firstly, we study the effect of noisy triples on recommendations. In Fig. 4, we show the performance of three existing KG-aware RS methods CoFM, CKE and CFKG on DBBook2014 w.r.t F1@5. We observe that (1) with the increase of noise ratio the overall performance of all three methods are degraded. It indicates that noisy triples negatively affect the performance of KG-aware methods. (2) The effect of noisy triples is different for different methods. For example, noisy triples have more effect on CFKG than CoFM. This is because the embedding of entity with noise in CoFM is reweighted to determine ratings while CFKG are not.

4.6. Performance for recommendations (RQ2)

For recommendations, we evaluate methods in three scenarios w.r.t KG datasets without noise injection, effect of noisy triples and top-K recommendations.

In Table 2, we show the performance comparison on DBook2014 and MovieLens-1M with DBPedia without noise injection. We observe that (1) our proposed TrustRec(N) consistently achieves the best performance and TrustRec(W) achieves the second best 5 out of 6 settings. (2) TrustRec(N) outperforms TrustRec(W) because it can flexibly learn a proper way to incorporate triple trustworthiness into the energy function. (3) Two GNN-based methods do not show superior performance because their node features are randomly generated and thus such features are not related with information of KGs. (4) TrustRec(W) is

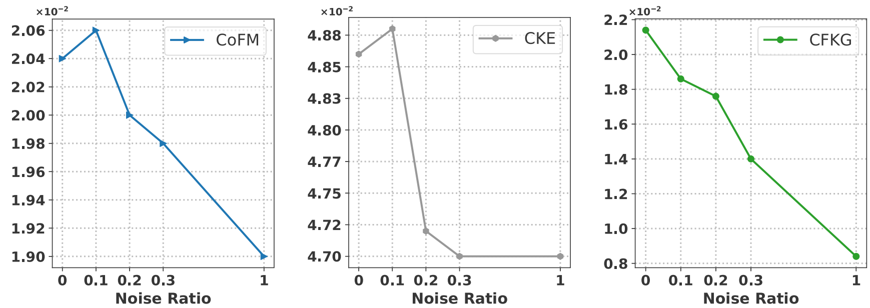

Fig. 4. Effect of noisy triples in DBBook2014 (The y-axis indicates the metric Precision@5). It indicates that noisy triples negatively affect the performance of KG-aware methods.

Table 2
The comparison results about recommendation on the KG dataset without noise injection. The best results are in **bold** and the second best ones are <u>underlined</u>.

	DBBook2014 (@5, %)			MovieLens-1M (@10, %)		
	F1	Precision	Recall	F1	Precision	Recall
CoFM	2.56	2.04	4.39	2.71	2.72	4.01
CKE	6.08	4.86	10.07	3.47	3.46	<u>5.38</u>
CFKG	2.70	2.14	4.55	2.51	2.67	3.54
KTUP	4.72	3.82	7.78	3.54	3.64	5.16
KGCN	2.10	1.54	3.31	2.02	1.89	2.23
KGNN-LS	2.15	1.65	3.11	1.88	1.85	1.94
MKR	3.16	2.36	4.99	1.80	1.73	1.90
TrustRec (N)	**6.33**	**5.04**	**10.64**	**3.80**	**3.84**	**5.57**
TrustRec (W)	<u>6.25</u>	<u>5.02</u>	<u>10.17</u>	<u>3.66</u>	<u>3.80</u>	5.27

superior over CKE, which indicates the efficacy of our trustworthiness estimator.

In Table 3, we show performance comparison over the effect of noisy triples. We observe that (1) with the increase of noisy triples our TrustRec(N) consistently outperforms compared methods. Also our TrustRec(W) can achieve the second best 10 out of 16 settings. It indicates our both methods are noise-tolerant. (2) The overall performance of all methods is degraded with the increase of noisy triples. (3) The trustworthiness estimator is a key component of our model, which aims to learn the degree of trustworthiness of triples. For the ablation study, our proposed TrustRec (W) is equivalent with CKE [12] if our trustworthiness estimator is neglected. To verify the effectiveness of our proposed trustworthiness estimator, our TrustRec (W) outperforms the CKE in five out of six scenarios on three evaluations over two real-world datasets, and improves CKE by 3.4%.

For top-K recommendation, in Table 4, we report the performance of KG-aware methods over precision at $K = \{3, 5, 10, 15, 20\}$. For each K, we report the averaged performance over a range of noise ratio $\{0, 0.1, 0.2, 0.3, 1.0\}$ because noises in KGs can significantly affect performance. We can see that our TrustRec(N) is consistently superior over all baselines.

For the sensitivity analysis for hyper-parameters, we show the learning rate and embedding size of the user and item on both datasets DBbook2014 and MovieLens-1M. From Fig. 5, with the increase of embedding size and learning rate, the performance of TrustRec(N) and TrustRec(W) is generally improved.

We give a significance test and show the $p-$ value for our best TrustRec(N) and the best baseline CKE. From Table 5 in this response, the p-value obtained in our analysis is greater than 0.05. Our TrustRec(N) does not significantly outperform the best baseline CKE, although the performance of TrustRec(N) has practical implications in both large datasets. It can be caused by the small sample size since each experiment is repeated five times. Therefore, in our future work, we will run each experiment more times (e.g., 20 times).

4.7. Performance for KG completion (RQ3)

We evaluate on a KG completion task that predicts the missing entity e_h or e_t. For each missing entity, we take all entities as candidates and rank them according to the scores computed based on entity and relation embeddings. Fig. 6 shows the overall performance with the increase of ratio of noisy triples. We do not show the performance of TrustRec(N) since feeding all unseen triples (more than 100 billion in DBBook2014) to our neural energy function is unfeasible. From Fig. 6, we observe that TrustRec(W) has superior performance over SOTA KG-aware RS. For KG completion, our TrustRec (W) improves CKE by 65.1% in terms of averaging hit ratio in two datasets.

Table 3
The comparison results about recommendation on the effect of noisy triples. The best results are in **bold** and the second best ones are <u>underlined</u>.

Datasets	Noise ratio	CoFM	CKE	CFKG	KTUP	KGCN	KGNN-LS	MKR	TrustRec (N)	TrustRec (W)
DBBook (F1@5, %)	0.1	2.62	6.08	2.30	4.29	1.92	2.16	3.04	**6.25**	<u>6.14</u>
	0.2	2.47	5.91	2.17	4.61	2.28	2.12	3.13	**6.23**	<u>6.12</u>
	0.3	2.54	5.84	1.81	4.95	2.01	2.36	3.09	**6.15**	<u>5.96</u>
	1	2.38	<u>5.88</u>	1.03	4.10	2.33	2.26	3.00	**6.02**	5.75
DBBook (P@5, %)	0.1	2.06	4.88	1.86	3.46	1.48	1.61	2.25	**4.98**	<u>4.96</u>
	0.2	2.00	4.72	1.76	3.72	1.68	1.60	2.33	**4.98**	<u>4.86</u>
	0.3	1.98	4.70	1.40	3.98	1.50	1.79	2.31	**4.92**	<u>4.76</u>
	1	1.90	<u>4.70</u>	0.84	3.24	1.74	1.76	2.20	**4.80**	4.56
MovieLens (F1@10, %)	0.1	2.83	3.54	2.13	<u>3.46</u>	2.31	2.05	1.66	**3.63**	3.61
	0.2	2.82	3.53	2.26	3.41	1.82	1.92	1.77	**3.63**	<u>3.46</u>
	0.3	2.71	3.44	1.92	3.37	1.96	2.18	1.88	**3.71**	<u>3.52</u>
	1	2.54	3.31	1.14	3.44	1.75	1.95	1.80	<u>3.45</u>	**3.49**
MovieLens (P@10, %)	0.1	2.89	<u>3.62</u>	2.28	3.51	2.11	1.90	1.62	**3.66**	3.61
	0.2	2.89	<u>3.59</u>	2.38	3.45	1.70	1.84	1.70	**3.61**	3.51
	0.3	2.83	3.47	2.10	3.43	1.84	1.92	1.74	**3.71**	<u>3.53</u>
	1	2.56	3.33	1.27	**3.50**	1.53	1.76	1.69	<u>3.48</u>	3.46

Table 4
The comparison results on top-K recommendation. The best results are in **bold** and the second best ones are underlined.

Top K	DBBook2014 (P, %)					MovieLens-1M (P, %)				
	@3	@5	@10	@15	@20	@3	@5	@10	@15	@20
CoFM	2.62	2.00	1.47	1.35	1.23	3.34	3.05	2.78	2.69	2.57
CKE	5.89	4.77	3.43	2.83	2.46	4.41	4.05	3.49	3.29	3.10
CFKG	1.83	1.60	1.33	1.15	1.04	2.53	2.32	2.14	2.05	2.02
KTUP	4.21	3.64	2.73	2.32	2.02	3.98	3.69	3.51	3.24	3.03
KGCN	2.11	1.59	1.06	0.83	0.68	2.52	2.16	1.81	1.56	1.45
KGNN-LS	2.16	1.68	1.13	0.88	0.72	2.29	2.15	1.85	1.64	1.46
MKR	2.74	2.27	1.59	1.33	1.09	2.16	2.01	1.69	1.50	1.38
TrustRec (N)	**6.07**	**4.94**	**3.59**	**2.92**	**2.53**	**4.48**	**4.09**	**3.66**	**3.39**	**3.19**
TrustRec (W)	5.96	4.83	3.48	2.84	2.44	4.25	4.01	3.58	3.33	3.12

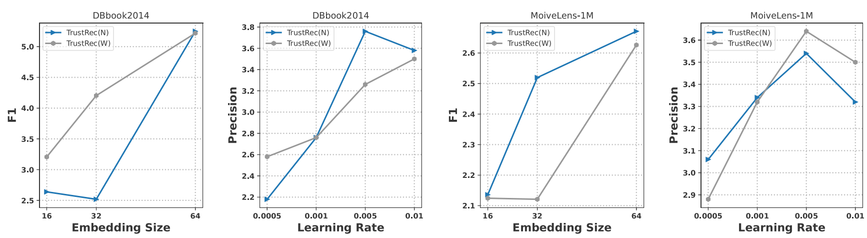

Fig. 5. Sensitivity analysis for the hyper-parameters embedding size of the user and item and learning rate on both datasets DBBook2014 and MovieLens-1M.

Table 5
This table shows the p-value for our best TrustRec(N) and best baseline (CKE).

	DBBook2014 (@5, %)			MovieLens-1M (@10, %)		
	F1	Precision	Recall	F1	Precision	Recall
p−value	0.59	0.65	0.54	0.33	0.26	0.75

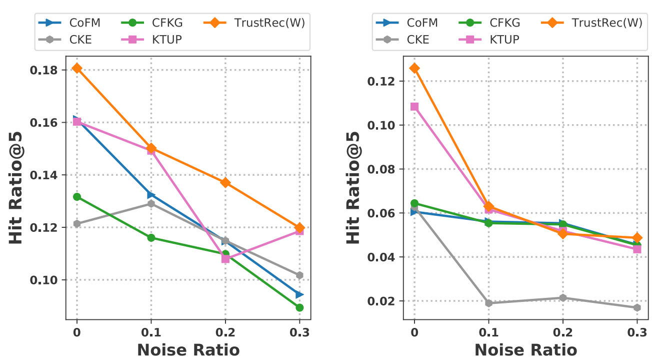

Fig. 6. The comparison results on the KG completion task for DBPedia in DBBook2014 (left) and MovieLens-1M (right). The higher *Hit Ratio*@5 indicates the better performance. TrustRec(W) outperforms the current state-of-the-art KG-aware recommendation system. Specifically, in the task of KG completion, TrustRec(W) shows a significant improvement of 65.1% in terms of the average hit ratio across two datasets compared to CKE.

4.8. Computational time

We show computational time of all compared methods in Table 6. We observe that (1) KGCN and KGNN-LS are not efficient due to personalised relation score function. (2) Our TrustRec(N) and TrustRec(W) are not efficient because both need to train a trustworthiness estimator while all baselines do not need to train it. (3) Our TrustRec(W) is more efficient than TrustRec(N) because TrustRec(W) uses a direct weighted loss function to avoid to train an additional neural network.

5. Conclusion

In this paper, we proposed TrustRec that can estimate trustworthiness of triples in KGs through an estimator that uses motifs, communities and global information. Based on it, we proposed triple trustworthiness integration to learn noise-tolerant KGR, and entity trustworthiness to learn noise-tolerant item representations of RS. We conducted experiments to show that TrustRec outperforms SOTA methods. Graph convolutional neural networks (GCNs) [54,55] have achieved great success in various applications, such as natural language processing, and computer vision, due to their excellent expressive power. Motivated by it, our structure-aware trustworthiness estimator can be further elaborately designed to capture motif, community and global structure for knowledge graph representation with GCNs.

CRediT authorship contribution statement

Yan Ge: Conceptualization, Data curation, Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Jun Ma:** Formal analysis, Funding acquisition, Methodology, Supervision. **Li Zhang:** Formal analysis, Methodology, Writing – review & editing. **Xiang Li:** Formal analysis, Investigation, Methodology, Writing – review & editing. **Haiping Lu:** Formal analysis, Funding acquisition, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

No conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

Data availability

Data will be made available on request.

Acknowledgement

This work is partly supported by the Amazon Research Awards, United States.

Table 6
Computational time (in seconds).

	CoFM	CKE	CFKG	KTUP	KGCN	KGNN-LS	Trust Rec(N)	Trust Rec(W)
DBBook2014	596	856	654	912	3281	1221	1536	1167
Movie Lens-1M	2105	2955	1986	2230	7556	5393	7219	5368

References

[1] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[2] P. Bai, Y. Ge, F. Liu, H. Lu, Joint interaction with context operation for collaborative filtering, Pattern Recognit. 88 (2019) 729–738.
[3] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: RecSys, 2010, pp. 135–142.
[4] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE TKDE 29 (12) (2017) 2724–2743.
[5] L. Dong, F. Wei, M. Zhou, K. Xu, Question answering over freebase with multi-column convolutional neural networks, in: ACL, 2015, pp. 260–269.
[6] J. Wang, Z. Wang, D. Zhang, J. Yan, Combining knowledge with deep convolutional neural networks for short text classification, in: IJCAI, 2017, pp. 2915–2921.
[7] Y. Ge, P. Peng, H. Lu, Mixed-order spectral clustering for complex networks, Pattern Recognit. 117 (2021) 107964.
[8] B. Wang, H. Xu, C. Li, Y. Li, M. Wang, TKGAT: Graph attention network for knowledge-enhanced tag-aware recommendation system, Knowl.-Based Syst. 257 (2022) 109903.
[9] P.M.T. Do, T.T.S. Nguyen, Semantic-enhanced neural collaborative filtering models in recommender systems, Knowl.-Based Syst. 257 (2022) 109934.
[10] C. Wang, L. Li, H. Zhang, D. Li, Quaternion-based knowledge graph neural network for social recommendation, Knowl.-Based Syst. 257 (2022) 109940.
[11] Y. Zhao, K. Wang, G. Guo, X. Wang, Learning compact yet accurate Generative Adversarial Networks for recommender systems, Knowl.-Based Syst. 257 (2022) 109900.
[12] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: KDD, 2016, pp. 353–362.
[13] Y. Cao, X. Wang, X. He, Z. Hu, T.-S. Chua, Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences, in: WWW, 2019, pp. 151–161.
[14] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, Z. Wang, Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: KDD, 2019, pp. 968–977.
[15] R. Xie, Z. Liu, F. Lin, L. Lin, Does william shakespeare really write hamlet? knowledge representation learning with confidence, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[16] S. Jia, Y. Xiang, X. Chen, K. Wang, Triple trustworthiness measurement for knowledge graph, in: The World Wide Web Conference, 2019, pp. 2865–2871.
[17] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 2124–2133.
[18] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: KDD, 2014, pp. 601–610.
[19] X. Li, A. Taheri, L. Tu, K. Gimpel, Commonsense knowledge base completion, in: ACL, 2016, pp. 1445–1455.
[20] Z. Li, H. Liu, Z. Zhang, T. Liu, N.N. Xiong, Learning knowledge graph embedding with heterogeneous relation attention networks, IEEE Trans. Neural Netw. Learn. Syst. 33 (8) (2021) 3961–3973.
[21] Z. Zhang, Z. Li, H. Liu, N.N. Xiong, Multi-scale dynamic convolutional network for knowledge graph embedding, IEEE Trans. Knowl. Data Eng. 34 (5) (2020) 2335–2347.
[22] H. Liu, C. Zheng, D. Li, X. Shen, K. Lin, J. Wang, Z. Zhang, Z. Zhang, N.N. Xiong, EDMF: Efficient deep matrix factorization with review feature learning for industrial recommender system, IEEE Trans. Ind. Inform. 18 (7) (2021) 4361–4371.
[23] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: The Semantic Web, Springer, 2007, pp. 722–735.
[24] K. Bollacker, R. Cook, P. Tufts, Freebase: A shared database of structured general human knowledge, in: AAAI, Vol. 7, 2007, pp. 1962–1963.
[25] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka, T.M. Mitchell, Toward an architecture for never-ending language learning, in: Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010.
[26] S. Heindorf, M. Potthast, B. Stein, G. Engels, Vandalism detection in wikidata, in: CIKM, 2016, pp. 327–336.
[27] J. Liang, Y. Xiao, Y. Zhang, S.-w. Hwang, H. Wang, Graph-based wrong IsA relation detection in a large-scale lexical taxonomy, in: AAAI, 2017.
[28] D. Lukovnikov, A. Fischer, J. Lehmann, S. Auer, Neural network-based question answering over knowledge graphs on word and character level, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1211–1220.
[29] X. Chen, M. Chen, W. Shi, Y. Sun, C. Zaniolo, Embedding uncertain knowledge graphs, in: Proceedings of the AAAI Conference, Vol. 33, 2019, pp. 3363–3370.
[30] J. Ma, C. Zhou, Y. Wang, Y. Guo, G. Hu, Y. Qiao, Y. Wang, PTrustE: A high-accuracy knowledge graph noise detection method based on path trustworthiness and triple embedding, Knowl.-Based Syst. 256 (2022) 109688.
[31] J. Huang, Y. Zhao, W. Hu, Z. Ning, Q. Chen, X. Qiu, C. Huo, W. Ren, Trustworthy knowledge graph completion based on multi-sourced noisy data, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 956–965.
[32] Y. Zhao, Z. Li, W. Deng, R. Xie, Q. Li, Learning entity type structured embeddings with trustworthiness on noisy knowledge graphs, Knowl.-Based Syst. 215 (2021) 106630.
[33] L. Zhang, Y. Ge, J. Ma, J. Ni, H. Lu, Knowledge-aware neural collective matrix factorization for cross-domain recommendation, 2022, arXiv preprint arXiv:2206.13255.
[34] S. Ji, S. Pan, E. Cambria, P. Marttinen, P.S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, 2020, arXiv preprint arXiv:2002.00388.
[35] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: NeurIPS, 2013, pp. 2787–2795.
[36] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Aaai, Vol. 14, 2014, pp. 1112–1119.
[37] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: AAAI, 2015.
[38] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: ACL, 2015, pp. 687–696.
[39] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, C. Xu, Recurrent knowledge graph embedding for effective recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 297–305.
[40] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, T.-S. Chua, Explainable reasoning over knowledge graphs for recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5329–5336.
[41] X. Wang, X. He, Y. Cao, M. Liu, 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.
[42] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: WWW, 2019, pp. 2000–2010.
[43] L. Zhang, Y. Ge, H. Lu, Hop-hop relation-aware graph neural networks, 2020, arXiv preprint arXiv:2012.11147.
[44] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: UAI, Citeseer, 2009.
[45] A.R. Benson, D.F. Gleich, J. Leskovec, Higher-order organization of complex networks, Science 353 (6295) (2016) 163–166.
[46] C.E. Tsourakakis, J. Pachocki, M. Mitzenmacher, Scalable motif-aware graph clustering, in: WWW, 2017, pp. 1451–1460.
[47] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community preserving network embedding, in: AAAI, 2017.
[48] S. Cavallari, V.W. Zheng, H. Cai, K.C.-C. Chang, E. Cambria, Learning community embedding with community detection and node embedding on graphs, in: CIKM, 2017, pp. 377–386.
[49] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416.
[50] L. Page, S. Brin, R. Motwani, T. Winograd, The Pagerank Citation Ranking: Bringing Order to the Web, Tech. rep, Stanford InfoLab, 1999.
[51] Y. Ge, H. Lu, Trustworthiness-aware knowledge graph representation for explainable recommender systems, in: Graph Embedding and Mining on ECML-PKDD Workshop.
[52] Y. Zhang, Q. Ai, X. Chen, P. Wang, Learning over knowledge-base embeddings for recommendation, in: SIGIR, 2018.
[53] G. Piao, J.G. Breslin, Transfer learning for item recommendations and knowledge graph completion in item related domains via a co-factorization model, in: ESWC, Springer, 2018, pp. 496–511.
[54] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR, 2017.
[55] Z. Li, Y. Zhao, Y. Zhang, Z. Zhang, Multi-relational graph attention networks for knowledge graph completion, Knowl.-Based Syst. 251 (2022) 109262.