Hindawi
Complexity
Volume 2021, Article ID 3473849, 8 pages
https://doi.org/10.1155/2021/3473849

Research Article

Confidence-Aware Embedding for Knowledge Graph Entity Typing

Yu Zhao, Jiayue Hou, Zongjian Yu, Yun Zhang, and Qing Li

Fintech Innovation Center, School of Computer Science, Southwestern University of Finance and Economics, Chengdu, China

Correspondence should be addressed to Yun Zhang; zhangyun@swufe.edu.cn

Received 8 July 2020; Revised 15 October 2020; Accepted 26 March 2021; Published 16 April 2021

Academic Editor: Daniela Paolotti

Knowledge graphs (KGs) entity typing aims to predict the potential types to an entity, that is, (*entity, entity type* = *?*). Recently, several embedding models are proposed for KG entity types prediction according to the existing typing information of the (*entity, entity type*) tuples in KGs. However, most of them unreasonably assume that all existing entity typing instances in KGs are completely correct, which ignore the nonnegligible entity type noises and may lead to potential errors for the downstream tasks. To address this problem, we propose **ConfE**, a novel confidence-aware embedding approach for modeling the (*entity, entity type*) tuples, which takes *tuple confidence* into consideration for learning better embeddings. Specifically, we learn the embeddings of entities and entity types in separate entity space and entity type space since they are different objects in KGs. We utilize an asymmetric matrix to specify the interaction of their embeddings and incorporate the *tuple confidence* as well. To make the *tuple confidence* more universal, we consider only the internal structural information in existing KGs. We evaluate our model on two tasks, including entity type noise detection and entity type prediction. The extensive experimental results in two public benchmark datasets (i.e., FB15kET and YAGO43kET) demonstrate that our proposed model outperforms all baselines on all tasks, which verify the effectiveness of ConfE in learning better embeddings on noisy KGs. The source code and data of this work can be obtained from https://github.com/swufenlp/ConfE.

1. Introduction

Knowledge graphs (KGs) consist of a huge amount of triples, each of which is formally denoted as (*head entity*, *relation*, *tail entity*) (or $(e, r, \tilde{e})$). KGs are effective well-structural relational databases for knowledge acquisition. Beside the triples, KGs usually contain a great number of entity type instances in the form of (*entity, entity type*) (denoted by (e, τ)) [1], which indicate that an entity e is of a certain entity type τ. For example, an entity "*Tom Hanks*" is an instance of a type "*actor*." As an essential part of KGs, they play an important role in KGs and have been widely used in some NLP tasks such as entity linking [2] and relation extraction [3], and question answering (QA) [4]. For instance, the KG-driven QA system could utilize the entity type information in a query: "*Is Tom Hanks an* ***actor*** *?*"

In recent years, many KGs have been built from semi-structured data or free text, such as Freebase [5], YAGO [6], Knowledge Vault [7], and Google Knowledge Graph [7]. However, the large-scale knowledge graph automatic construction inevitably brings noises into KGs due to limited human supervision. For example, the open fine-grained entity typing system [8] even only achieves 58.8% accuracy, which reaffirms the existence of entity type noises in KG construction. Thus, the nonnegligible entity type noise problem extremely impedes the efficient use of KGs [9].

In this work, we focus on dealing with entity type noises located in existing KGs by learning entity type embeddings, which encodes all the entities and entity types into a latent vector space. Since the learning approach depends on the reliability of the existing *(entity, entity type)* tuples, it is crucial to consider the entity type noises for learning embeddings. There are some models proposed for KG entity type embedding learning [10–12]. However, most of the learning methods unreasonably assume that all the existing entity type instances in KGs are true, which may lead to some potential errors for downstream entity type sensitive tasks. To address this issue, we propose **ConfE**, a novel

confidence-aware embedding framework for entity type learning which takes the entity type noises into consideration. Figure 1 shows a simple illustration of our model ConfE, which learns entity type embeddings with the *tuple confidence* on noisy KGs. Such entity type noises are expected to be detected by ConfE and to be ignored in entity type embeddings learning.

Specifically, we build two different entity space and entity type space for learning the embeddings of entities and entity types since they are different objects in the (e, τ) tuple. We utilize a unique "*rdf:type*" relation matrix $\mathbf{M}$ to specify the interaction of their embeddings, that is, $\mathbf{e}^\top \mathbf{M} \tau$, which incorporates the tuple confidence $\mathbf{C}(e, \tau)$ as well. To make the tuple confidence more universal, we only utilize the internal structural information, which makes it more challenging. Accordingly, we propose two kinds of *tuple confidences* that correspondingly consider the local tuple and global triple structural information in KGs. The extensive experimental results on two tasks including entity type noise detection and entity type prediction show that our model achieves the best performance, which demonstrates the effectiveness of ConfE in learning better entity type embeddings in a noisy scenario.

The main contributions are concluded as follows:

(i) We propose ConfE, a novel confidence-aware embedding model for encoding the (*entity, entity type*) tuples to calculate the similarity of an entity and an entity type, which takes the *tuple confidence* into consideration.

(ii) We build two distinguish tuple confidences according to the local and global structural information in existing KGs. The overall confidence of them is utilized in the final energy function for learning better embeddings.

(iii) We conduct two experimental tasks including entity type noise detection and entity type prediction and utilize two public benchmark datasets (i.e., *FB15kET* and *YAGO43kET*) to verify the effectiveness of our model and the confidence-aware framework.

2. Related Work

2.1. KG Noise Detection. In recent years, the research of KG noise detection attracts wide attention, also known as KG refinement [13]. The noise issue can be roughly classified into two classes, that is, false relationships between entities (*head entity, relationship, tail entity*) and false entity type instances (*entity, entity type*). Most of the existing research concentrates on the deal with the noisy triple facts in KG [9, 14, 14–21]. For example, Jiang et al. [15] present a Markov logic-based system for cleaning an extracted knowledge base. Melo and Paulheim [17] propose an error detection method which relies on path and type features used by a classifier for every relation in the graph exploiting local feature selection. Neil et al. [18] introduce a regularized attention mechanism to GCNNs that not only improves performance on clean datasets but also favorably accommodates noise in KGs. Liang et al. [19] propose a method for graph-based wrong IsA relation detection in a large-scale lexical taxonomy. Pujara et al. [9] propose to improve the quality of knowledge graphs by removing errors and adding missing facts. Xie et al. [20] propose a confidence-aware knowledge representation learning framework that detects possible noises in KGs while learning knowledge representations with confidence simultaneously. Zhao et al. [21] propose a trustiness-aware method for KG noise detection. Despite their success, which focuses on detecting triple fact noises, their goals are different from this paper.

There are a few models of dealing with entity typing noises [22]. However, they mainly concentrate on association rule mining [23], heuristic link-based type inference [24]; therefore, they are constrained by the capability of generalization. Recently, Ren et al. [25] propose a heterogeneous partial-label embedding model for label noise detection. Templemeier et al. [26] propose an approach to predict the missing categories for particular entities that are obtained from noisy and sparse Web markup. Despite their success, their goals are different from KG entity type noises detection, and none of them consider the confidence of the entity type tuples. In this work, we concentrate on knowledge graph entity type noise detection and learn better entity type embeddings with confidence in a noisy scenario. The model illustration of KG noise detection is included in Table 1.

2.2. KG Embedding. Recently, KG embedding has become a hot topic in AI and NLP research field [27]. Most of the existing embedding models concentrate on learning the (*head entity, relationship, tail entity*) triples, such as SE [28], NTN [29], TransE [30], TransH [31], TransR [32], TransG [33], ComplexE [34], SSP [35], ProjE [36], ConvE [37], KBGAT [38], CapsE [39], and ConvKB [40], which pay less attention to the exploration of embedding the (*entity, entity type*) tuples. Recently, Neelakantan and Chang [10] propose a method to infer missing entity type instances, where they embedding the (e, τ) tuple by $\mathbf{e}^\top \tau$. However, they also use external information from Wikipedia besides the information within the existing KG. Moon et al. [11] propose an embedding approach for entity type embedding (ETE), in which they build the energy function as $\|\mathbf{e} - \tau\|_{\ell 1}$. Despite their success, they are lacking enough modeling capability due to their structural simplicity. In this work, we introduce an advanced embedding model with better expressive capability, which considers the structural information of both the (*entity, entity type*) tuples and the (*head entity, relationship, tail entity*) triples in KGs.

3. Methodology

To detect possible entity type noises in KGs and learn better entity type representations, we introduce a novel concept *tuple confidence* for each *(entity, entity type)* tuple. Tuple confidence describes the correctness and significance of a tuple, which could be measured according to local tuple and global triple information. *The novelty of this work is to model the confidence of entity type instances for typing noise*

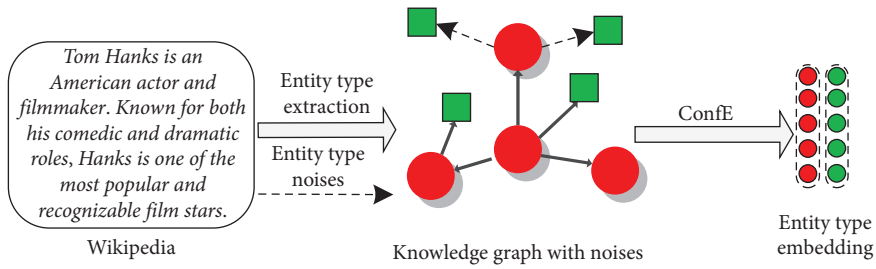

Figure 1: Entity type noises in KGs and learning entity type embeddings with confidence (ConfE).

Table 1: Model illustration of KG Noise Detection.

Models	Data	The method	The goal
Jiang's et al. [15]	Structured information in KG	Markov logic-based	Triple fact noises
PaTyBRED [17]	Path and type features in KG	Classifier	Triple fact noises
Neil's et al. [18]	Structured information in KG	Attention mechanism + GCNNs	Triple fact noises
Liang's et al. [19]	Large-scale lexical taxonomy	Graph-based method	IsA relation detection
Pujara's et al. [9]	Structured information in KG	Embedding techniques	Triple fact noises
TransT [21]	Structured information in KG + data outside KG	Trustiness-aware embedding method	Triple fact noises
CKRL [20]	Structured information in KG	Confidence-aware embedding-based	Triple fact noises
Ma's et al. [23]	Disjointness axioms in KG	Association rule mining	Inconsistency detection
SDType [24]	T-box information from the schema	Heuristic link-based type inference mechanism	Entity type noises
ConfE (this paper)	Structured information in KG	Confidence-aware embedding-based	Entity type noises

detection and propose an embedding method to model tuples. In the following, we first present the confidence-aware embedding learning framework and then describe the embedding model and the methods for calculating the tuple confidences.

3.1. Confidence-Aware Embedding Learning Framework. We intend to detect entity type noises and learn better entity type embeddings that take *tuple confidence* into consideration. *Our ConfE model should concentrate more on those tuples with higher confidence*. Similar to [20], we formally design the energy function of a tuple (e, τ) as follows:

$$E(\mathscr{H}) = \sum_{(e,\tau)\in\mathscr{H}} G(e,\tau)\cdot C(e,\tau), \tag{1}$$

where $\mathscr{H}$ denotes the set of all (e, τ) tuples in KGs. The energy function consists of two parts: **(i)** $G(e,\tau) = \mathbf{e}^{\top}\mathbf{M}\tau$ denotes the model score of the tuple (more details are included in Section 3.2), which assigns for an asymmetric matrix $\mathbf{M}$ that specifies the interaction of the latent presentation of entity and entity type. A higher $G(e,\tau)$ indicates better interaction between the latent embeddings of entity and entity type in the tuple. **(ii)** While different from conventional methods, we propose tuple confidence in the framework. $C(e,\tau)$ stands for the overall tuple confidence of the tuple (Section 3.3), whose value comes higher when the current tuple is worth considering. Higher tuple confidence $C(e,\tau)$ implies that the corresponding tuple is more credible and thus should be more considered. Tuple confidence can be calculated both during and after KG construction from different aspects including internal knowledge in KG (such as topological information) and external information (such as textual data). To make our tuple confidence more universal and flexible, we only consider the KG structural information. Accordingly, we propose local and global tuple confidence that are learned iteratively during model training.

3.2. Model Optimization. Following [30], we utilize the margin-based ranking loss function to train our model ConfE. The main idea is that each tuple in the training set $\mathscr{H}^{(i)} = (t^{(i)}, \tau^{(i)})$ should receive a higher score than a corrupt tuple in which a type is replaced with a random entity type. The ranking loss function is defined as follows:

$$\mathscr{L} = \sum_{(e,\tau)\in\mathscr{H}} \sum_{(e',\tau')\in\mathscr{H}'} \max\{0, \gamma_1 - \mathbf{G}(e,\tau) + \mathbf{G}(e',\tau')\}\cdot \mathbf{C}(e,\tau), \tag{2}$$

where $G(e,\tau)$ and $G(e',\tau')$ represent the model score of positive triple and negative triple, respectively. *The tuple confidence* $C(e,\tau)$ *makes our algorithm learning more on those convincing tuples with higher confidence.* γ_1 is a hyperparameter for distinguishing positive instance and negative one. $\mathscr{H}'$ is a set of corrupt tuples built in the following way:

$$\begin{aligned}\mathscr{H}' := &\{(e',\tau) | (e,\tau)\in\mathscr{H} \cap e'\in \mathrm{E} \cap (e',\tau)\notin\mathscr{H}\}\\ &\cup\{(e,\tau') | (e,\tau)\in\mathscr{H}\cap\tau'\in\mathscr{T}\cap(e,\tau')\notin\mathscr{H}\}.\end{aligned} \tag{3}$$

Note that we do not replace both entity and entity type with a random one at the same time.

3.3. Embedding Model. We introduce the embedding model of a (e, τ) tuple in this section. Similar to [11], we treat the (e, τ) tuples as triple facts that only have a unique relationship "*rdf:type*", for example, (*Tom Hanks, rdf:type, actor*). Accordingly, we assign for the "*rdf:type*" relationship an asymmetric matrix $\mathbf{M}$ that specifies the interaction of the latent presentation of entity and entity type, which is inspired by the previous embedding model RESCAL [41]. Formally, the embedding model of a given (e, τ) tuple is designed as follows:

$$\mathbf{G}(e, \tau) = \mathbf{e}^{\top}\mathbf{M}\boldsymbol{\tau}, \tag{4}$$

where $e \in \mathscr{E}, \tau \in \mathscr{T}$, $\mathscr{E}$, and $\mathscr{T}$ are the set of entities and entity types, respectively. Different from the conventional methods that encode entities and types into a common space, we build two distinct latent vector spaces for them, that is, entity space and entity type space, since the entities and entity types are different objects in KGs. $\mathbf{e} \in \mathbb{R}^{\kappa}$ stands for the representation of an entity e in entity space and $\tau \in \mathbb{R}^{\ell}$ is the representation of a type τ in entity type space. $\mathbf{M} \in \mathbb{R}^{\kappa \times \ell}$ denotes the asymmetric matrix. Since the representations of entity types indicate the common knowledge of all their entities, therefore, they usually have fewer parameters, that is, $\ell < \kappa$. The model score is expected to be higher for a positive tuple and lower for a negative one.

3.4. Tuple Confidence. In this section, we will introduce the detailed methods of calculating the tuple confidence, which consists of two parts: (i) *local tuple confidence* $LC(e, \tau)$, which only considers the inside structural information of a tuple, and (ii) *global triple confidence* $GC(e, \tau)$, which considers the global triple information in KGs.

3.4.1. Local Tuple Confidence. We first come up with *local tuple confidence* $LC(e, \tau)$ which only concentrates on the inside of a tuple. We assume that the more a tuple fits the interaction assumption, the more convincing this tuple should be considered. The basic idea behind it is that the model score of the positive tuple should be higher than the negative one. *We believe that the more the value of the margin-based objective function, the more convincing the tuple should be considered in training.* To measure the local tuple confidence during training, we first judge the current conformity of each tuple with interaction assumption. Inspire by the margin-based training strategy, we directly utilize it to represent the local tuple quality $\mathbf{Q}_{lt}(e, \tau)$ as follows:

$$\mathbf{Q}_{lt}(e, \tau) = -\left(\gamma_1 - \mathbf{G}(e, \tau) + \mathbf{G}\left(e', \tau'\right)\right). \tag{5}$$

A higher $\mathbf{Q}_{lt}(e, \tau)$ usually indicates a better tuple judged by the interaction assumption. Hence, the local tuple confidence $LC(e, \tau)$ changes with its corresponding tuple quality $\mathbf{Q}_{lt}(e, \tau)$, which is formally built as follows:

$$LC(e, \tau) = \begin{cases} \alpha \cdot LC(e, \tau), & \mathbf{Q}_{lt}(e, \tau) \le 0, \\ \min\{LC(e, \tau) + \beta, 1\}, & \mathbf{Q}_{lt}(e, \tau) > 0. \end{cases} \tag{6}$$

We assume all given tuples are true and set $LC(e, \tau) = 1$ at the beginning, which would be continuously updated during training. $\alpha \in (0, 1)$ and $\beta > 0$ are hyperparameters that control the speed of $LC(e, \tau)$ when updated descendingly and ascendingly, respectively. If $Q_{lt}(e, \tau) \le 0$, it indicates that the interaction between the entity and entity type performs poorly, and thus the local tuple confidence should decrease; otherwise, it should increase it. The local tuple confidence $LC(e, \tau)$ will decrease at a geometric rate and increase with a constant addition. It urges to punish the violations of interaction rule for those tuples which are more likely to be noises; therefore, they should have smaller confidences.

3.4.2. Global Tuple Confidence. Despite the success of *LC*, it only concentrates on the inside of tuples, ignoring valuable triple facts in KGs. We observe that the relational triple information is also helpful to judge tuple qualities. Inspired by the work in [1], we first build the entity type triple (*head type, relationship, tail type*) by replacing both head entity and tail entity with their corresponding entity types, that is, $(e, r, \tilde{e}) \longrightarrow^{\text{replace}} (\tau, r, \tilde{\tau})$, using two entity type tuples (e, τ) and $(\tilde{e}, \tilde{\tau})$. The main idea behind it is that a significant premise of a triple holds is that their corresponding entity types should obey their relationship. Accordingly, we utilize the translating assumption [30] to model the entity type triples, that is, $I(\tau, r, \tilde{\tau}) = \|\tau + \mathbf{r} - \tilde{\tau}\|$. *We believe that the more an entity type triple fits the translation assumption, the more convincing this entity type tuple should be considered.* Therefore, we calculate the global triple quality $Q_{gt}(e, \tau)$ of an entity type tuple (e, τ) as follows:

$$\mathbf{Q}_{gt}(e, \tau) = -\left(\gamma_2 + \mathbf{S}(\tau, r, \tilde{\tau}) - \mathbf{S}\left(\tau, r, \tilde{\tau}'\right)\right), \tag{7}$$

where $\tilde{\tau} \in \{\tilde{\tau} | (e, r, \tilde{e}) \in \mathscr{D}, (\tilde{e}, \tilde{\tau}) \in \mathscr{H}\}$, $\mathscr{D}$ denotes the set of positive triple facts in KGs. $\tilde{\tau}'$ is a random negative entity type. $\gamma_2 > 0$ is a hyperparameter. Hence, the global tuple confidence $GC(e, \tau)$ can be learned during training as follows:

$$GC(e, \tau) = \begin{cases} \alpha \cdot GC(e, \tau), & \mathbf{Q}_{gt}(e, \tau) \le 0, \\ \min\{GC(e, \tau) + \beta, 1\}, & \mathbf{Q}_{gt}(e, \tau) > 0. \end{cases} \tag{8}$$

Here, the iterative learning process of $GC(e, \tau)$ is similar to $LC(e, \tau)$. We assume all tuples are true and $GC(e, \tau) = 1$ at the beginning, which are continuously updated during training. $\alpha \in (0, 1)$ and $\beta > 0$ are hyperparameters that control the speed of $GC(e, \tau)$ $(\in (0, 1])$ updating.

3.4.3. Overall Tuple Confidence. To the end, we build overall tuple confidence for confidence-aware energy function. The overall tuple confidence consists of the following two parts: (i) local tuple confidence $LC(e, \tau)$ and (2) global tuple confidence $GC(e, \tau)$, which is formally designed as follows:

$$\mathbf{C}(e,\tau) = \lambda \cdot LC(e,\tau) + (1-\lambda)\cdot GC(e,\tau), \quad (9)$$

where $\lambda \in (0,1)$ is a parameter for trade-off.

4. Experiments

In this section, we evaluate the effectiveness of ConfE on entity type noise detection and entity type prediction.

4.1. Datasets. The two public benchmark datasets for the experiments are directly taken from [42]. The basic statistics of the datasets are in Table 2. Specifically, FB15k [30] and YAGO43k [11] are extracted from Freebase [5] and YAGO [6], respectively. We utilize entity type datasets called FB15kET and YAGO43kET built in [11], which are composed of tuples, in which the entity types are mapped to entities from FB15k and YAGO43k, respectively. Three noisy datasets containing tuples like (e', τ) or (e, τ') based on the training set of *FB15kET* are built in which the noises are 10%, 20%, and 40%, that is, *FB15kET-N1*, *FB15kET-N2*, and *FB15kET-N3*. Similarly, the noisy datasets *YAGO43kET-N1*, *YAGO43kET-N2*, and *YAGO43kET-N3* are built based on YAGO43kET.

4.2. Baselines and Configurations. The parameters we trained for our model are as follows: γ_1, γ_2, the hyperparameters of the margin: {1, 3, 5, 7, 10}; δ-learning rate: {0.1, 0.01, 0.001, 0.0001}; (κ, ℓ), the embedding dimensions of entity and entity type: {(50, 30), (100, 50), (150, 100), (200, 150), (200, 200)}; α-descend controller: {0.96, 0.97, 0.98, 0.99}; β-ascend controller: {0.001, 0.002, 0.005}; and λ-combination weights: {0.1, 0.2, 0.3, 0.7, 0.8, 0.9}.

To confirm the best value of parameters, we train ConfE on the validation dataset. The optimal combination of parameters setting for ConfE is $\gamma_1 = 7, \gamma_2 = 1, \delta = 0.01, \kappa = 50, \ell = 30, \alpha = 0.98, \beta = 0.02, \lambda = 0.7$ on FB15kET; and $\gamma_1 = 7, \gamma_2 = 7, \delta = 0.01, \kappa = \ell = 200, \alpha = 0.98, \beta = 0.02, \lambda = 0.7$ on YAGO43kET. Moreover, we utilize the TransE [30] model to initialize the embeddings of entities and relationships.

We compare our model with the recent baselines: ETE [11], TransE-ET [11], TrustE(LT) [42], and TrustE(LT + GT) [42]. The results of all baselines are directly taken from Zhao et al. [42].

4.3. Entity Type Noise Detection. In this experiment, we conduct entity type noise detection, that is, detecting possible noisy entity types according to their tuple scores.

Evaluation protocol: We consider the model score: $\mathbf{G}(e,\tau) = \mathbf{eM}\tau$ for entity type tuple. Similar to [29], we rank all (*entity, entity type*) tuples in the noisy training set by their scores in descending order. Therefore, the tuples with lower ranking would more likely be noisy ones. We utilize the precision/recall curves to demonstrate the effectiveness of our model.

Experimental results: Figures 2 and 3 show the performances of all models on entity type noise detection, from which we can find that (i) On FB15kET and YAGO43kET, our ConfE model achieves the best performance under different noise rates, which confirms that ConfE could effectively and competently detect entity type noises in KGs. As the recall increases, the improvement introduced by our ConfE model over the baseline grows more insignificant, which reaffirms that the noises greatly impede entity type noise detection. (ii) Compared to YAGO43kET, the ConfE model seems to perform more significantly on FB15kET. Considering that there are 37 relations in YAGO43kET while 1345 in FB15kET, the sparseness of relationships harms the effectiveness of the type-relation-type training set. Such sparseness causes a relation to be connected to too much entity type so that the embedding of relation may not be capable of accurately describing its internal connection with different entity types. The results also verify the effectiveness and robustness of our model in both scenarios.

4.4. Entity Type Prediction. This task aims to verify the effectiveness of the ConfE model in entity type prediction, that is, completing the missing entity type tuple (*entity, entity type = ?*).

Evaluation protocol: For each tuple, we first remove its entity type and fill the resulting vacancies with all the entity types in turn as candidate tuples. Secondly, we compute the score of each candidate tuple based on the function $G(e,\tau)$ and rank them in descending order. Then, we can get the rank of the original tuple. Finally, we use (1) the mean reciprocal rank (MRR) and (2) the proportion of correct entity types ranked in the top 10 (HITS@10(%)) as evaluation metrics for comparison. We follow the method utilized in [30] to define evaluation settings of "Raw" and "Filter":

$$\text{MRR} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{1}{\text{rank}_i}, \quad (10)$$

where C is the collection of all testing (*entity, entity type*) tuples and $rank_i$ is the rank location of the true candidate tuple for the i-th pair.

Experimental results: Tables 3 and 4 show the result of all models on entity type prediction, from which we could observe that (i) ConfE consistently and significantly performs better than the baselines on FB15kET noisy testing datasets with all evaluation metrics. It reaffirms the quality of knowledge embedding in our ConfE model, which is also helpful for both KG entity type prediction and entity type noise detection. (ii) Our ConfE model outperforms on MRR in YAGO43kET noisy testing datasets in "raw" setting. Compared with HITS@10, MRR places more importance on the average ranking of the original tuple. We guess that although ConfE may be not as good as baselines, it also has considerable advantages in improving the average prediction accuracy. (iii) In the setting of "filter", ConfE performs better on HITS@10 and has a comparable

Table 2: Statistics of datasets.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
FB15k	14,951	1,345	483,142	50,000	59,071
YAGO43k	42,975	37	331,687	30,000	30,000
Dataset	#Ent	#Type	#Train	#Valid	#Test
FB15kET	14,951	3,851	136,618	16,000	16,000
YAGO43kET	42,975	49,980	378,172	46,000	46,000
Dataset	FB15kET-N1 (10%)	FB15kET-N2 (20%)	FB15kET-N3 (40%)		
#Neg	15200	34150	91000		
Dataset	YAGO43kET-N1 (10%)	YAGO43kET-N2 (20%)	YAGO43kET-N3 (40%)		
#Neg	37817	75634	197753		

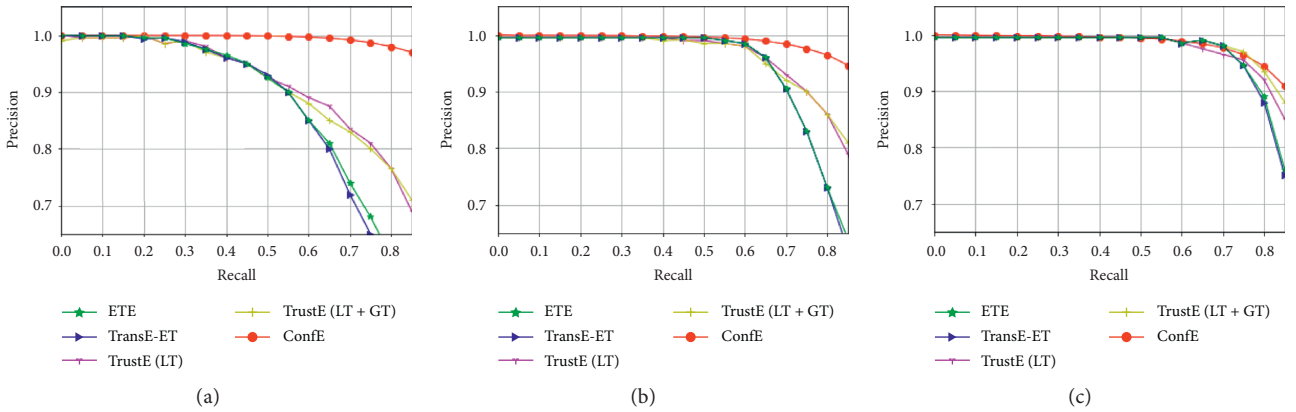

Figure 2: Entity type noise detection results. Evaluation on (a) FB15kET-N1, (b) FB15kET-N2, and (c) FB15kET-N3.

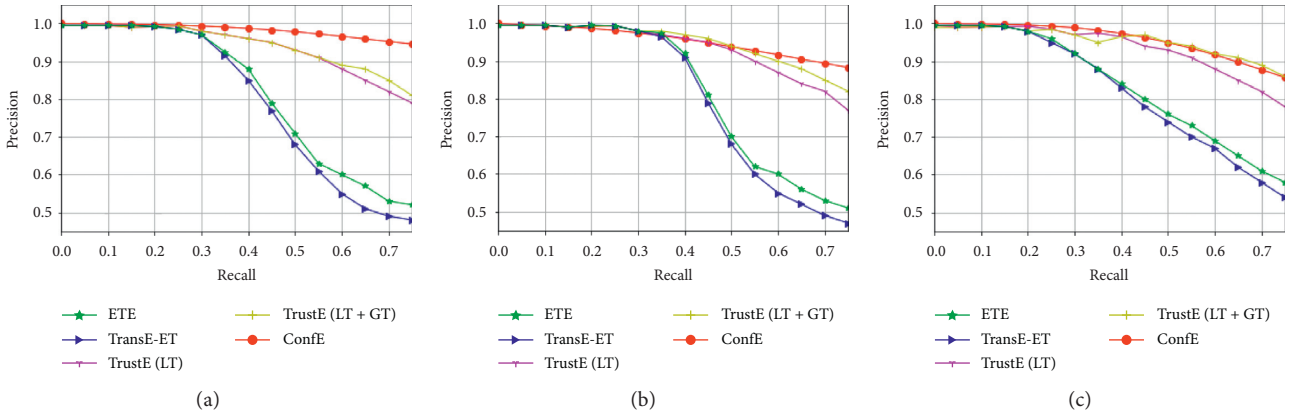

Figure 3: Entity type noise detection results. Evaluation on (a) YAGO43kET-N1, (b) YAGO43kET-N2, and (c) YAGO43kET-N3.

Table 3: Entity type prediction results. Evaluation of different models on FB15kET-N1, FB15kET-N2, and FB15kET-N3.

Dataset	FB15kET-N1				FB15kET-N2				FB15kET-N3			
Metrics	MRR		HITS@10		MRR		HITS@10		MRR		HITS@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE-ET	0.04	0.22	45.61	67.65	0.05	0.21	44.96	67.58	*0.05*	0.20	44.95	66.48
ETE	0.04	0.23	45.66	68.54	0.05	0.22	45.21	67.82	*0.05*	0.20	45.52	66.83
TrustE (LT)	0.05	*0.30*	45.76	68.58	0.05	0.29	45.90	68.15	0.04	0.28	44.73	67.75
TrustE (LT + GT)	*0.05*	0.29	*45.80*	*70.29*	0.05	*0.29*	*46.36*	*69.86*	*0.05*	*0.29*	*47.07*	*68.89*
ConfE	**0.07**	**0.74**	**48.15**	**98.62**	**0.07**	**0.73**	**47.03**	**98.03**	**0.07**	**0.72**	**47.16**	**97.64**

The best scores are in given bold, and the second-best ones are given in italics.

Table 4: Entity type prediction results. Evaluation of different models on YAGO43kET-N1, YAGO43kET-N2, and YAGO43kET-N3.

Dataset	YAGO43kET-N1				YAGO43kET-N2				YAGO43kET-N3			
Metrics	MRR		HITS@10		MRR		HITS@10		MRR		HITS@10	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE-ET	*0.04*	0.11	*25.69*	35.99	0.03	0.10	24.54	34.92	0.03	0.09	23.38	33.96
ETE	0.03	0.11	25.56	36.11	0.03	0.11	24.68	35.21	0.03	0.09	23.50	34.32
TrustE (LT)	0.03	*0.12*	25.38	38.49	0.03	*0.12*	*25.11*	36.79	0.03	0.10	*25.24*	35.03
TrustE (LT + GT)	0.03	*0.12*	**25.89**	38.61	0.03	*0.12*	**26.09**	38.45	0.03	*0.11*	**25.87**	*36.28*
ConfE	**0.05**	**0.65**	19.25	**94.63**	**0.04**	**0.61**	18.20	**90.25**	**0.05**	**0.65**	19.84	**93.99**

The best scores are in given bold, and the second-best ones are given in italics.

performance on MRR, which confirms the capability of entity type prediction. Moreover, our model has stronger adaptability in large-scale data modeling than other state-of-art models.

5. Conclusion and Future Work

We propose a novel confidence-aware embedding framework (ConfE) for KG entity typing on a noisy knowledge graph which takes the (*entity, entity type*) *tuple confidence* into consideration. Specifically, we build a bilinear embedding model to model the *(entity, entity type)* tuple. Moreover, we calculate the tuple confidence by considering the internal structural information in KGs. We evaluate our models on two experiments including entity type noise detection and entity type prediction. Empirical experiment results on *FB15kET* and *YAGO43kET* demonstrate the effectiveness of the proposed ConfE model in entity type noise detection. Interesting future work direction includes exploring to detect noises in entity type instances and entity type triples simultaneously.

Data Availability

The data are available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yu Zhao and Jiayue Hou contributed equally.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 61906159, the Sichuan Science and Technology Program under Grant no. 2018JY0607, the Fundamental Research Funds for the Central Universities under Grant no. JBK2003008, and Fintech Innovation Center, and Financial Intelligence and Financial Engineering Key Laboratory of Sichuan Province.

References

[1] Y. Zhao, A. Zhang, R. Xie, K. Liu, and X. Wang, "Connecting embeddings for knowledge graph entity typing," in *Proceedings of ACL*, Seattle, WA, USA, July 2020.

[2] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proceedings of EMNLP*, pp. 2681–2690, Copenhagen, Denmark, September 2017.

[3] P. Jain, P. Kumar, and S. Chakrabarti, "Type-sensitive knowledge base inference without explicit type supervision," in *Proceedings of ACL*, Melbourne, Australia, July 2018.

[4] H. Elsahar, C. Gravier, and F. Laforest, "Zero-shot question generation from knowledge graphs for unseen predicates and entity types," in *Proceedings of NAACL*, pp. 218–228, New Orleans, LA, USA, June 2018.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of SIGMOD*, pp. 1247–1250, Vancouver, Canada, June 2008.

[6] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of WWW*, pp. 697–706, Prague, Czech Republic, June 2007.

[7] X. Dong, E. Gabrilovich, G. Heitz et al., "Knowledge vault: a web-scale approach to probabilistic knowledge fusion," in *Proceedings of SIGKDD*, pp. 601–610, New York, NY, USA, August 2014.

[8] B. Zhou, D. Khashabi, C.-T. Tsai, and D. Roth, "Zero-shot open entity typing as type-compatible grounding," in *Procedddings of EMNLP*, pp. 2065–2076, Brussels, Belgium, October 2018.

[9] J. Pujara, E. Augustine, and L. Getoor, "Sparsity and noise: where knowledge graph embeddings fall short," in *Proceedings of EMNLP*, pp. 1751–1756, Copenhagen, Denmark, September 2017.

[10] A. Neelakantan and M.-W. Chang, "Inferring missing entity type instances for knowledge base completion: new dataset and methods," 2015, http://arxiv.org/abs/1504.06658.

[11] C. Moon, P. Jones, and N. F. Samatova, "Learning entity type embeddings for knowledge graph completion," in *Proceedings of CIKM*, pp. 2215–2218, Singapore, November 2017.

[12] R. Zhang, F. Kong, C. Wang, and Y. Mao, "Embedding of hierarchically typed knowledge bases," in *Proceddings of AAAI*, New Orleans, LA, USA, February 2018.

[13] H. Paulheim, "Knowledge graph refinement: a survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, 2017.

[14] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, "Quality assessment for linked open data: a survey," *Semantic Web*, vol. 7, no. 1, pp. 63–93, 2016.

[15] S. Jiang, D. Lowd, and D. Dou, "Learning to refine an automatically extracted knowledge base using markov logic," in *Proceeding of ICDM*, pp. 912–917, Brussels, Belgium, December 2012.

[16] S. Heindorf, M. Potthast, B. Stein, and G. Engels, "Vandalism detection in wikidata," in *Proceedings of the International on*

Conference on Information and Knowledge Management (CIKM'16), ACM, pp. 327–336, Indianapolis, IN, USA, October 2016.

[17] A. Melo and H. Paulheim, "Detection of relation assertion errors in knowledge graphs," in *Proceedings of ACM*, p. 22, Singapore, November 2017.

[18] D. Neil, J. Briody, A. Lacoste, A. Sim, P. Creed, and A. Saffari, "Interpretable graph convolutional neural networks for inference on noisy knowledge graphs," in *Proceedings of NeurIPS*, Long Beach, CA, USA, December 2017.

[19] J. Liang, Y. Xiao, Y. Zhang, S. won Hwang, and H. Wang, "Graph-based wrong isa relation detection in a large-scale lexical taxonomy," in *Proceedings of AAAI*, pp. 1178–1184, San Francisco, CA, USA, February 2017.

[20] R. Xie, Z. Liu, F. Lin, and L. Lin, "Does william shakespeare really write hamlet? knowledge representation learning with confidence," in *Proceedding of AAAI*, Palo Alto, CA, USA, February 2018.

[21] Y. Zhao, H. Feng, and P. Gallinari, "Embedding learning with triple trustiness on noisy knowledge graph," *Entropy*, vol. 21, no. 11, 2019.

[22] H. Paulheim and C. Bizer, "Improving the quality of linked data using statistical distributions," *International Journal on Semantic Web and Information Systems*, vol. 10, no. 2, pp. 63–86, 2014.

[23] Y. Ma, H. Gao, T. Wu, and G. Qi, "Learning disjointness axioms with association rule mining and its application to inconsistency detection of linked data," in *The Semantic Web and Web Science*, pp. 29–41, Springer, Berlin, Germany, 2014.

[24] H. Paulheim and C. Bizer, "Type inference on noisy RDF data," *Advanced Information Systems Engineering*, vol. 8218, pp. 510–525, 2013.

[25] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han, "Label noise reduction in entity typing by heterogeneous partial-label embedding," in *Proceedings of SIGKDD*, pp. 1825–1834, ACM, San Francisco, CA, USA, August 2016.

[26] N. Tempelmeier, E. Demidova, and S. Dietze, "Inferring missing categorical information in noisy and sparse web markup," in *Proceedings of WWW*, pp. 1297–1306, Lyon, France, April 2018.

[27] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: a survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[28] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proceedings of AAAI*, pp. 301–306, San Francisco, CA, USA, August 2011.

[29] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proceedings of NeurIPS*, pp. 926–934, Lake Tahoe, NV, USA, December 2013.

[30] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of NeurIPS*, pp. 2787–2795, Lake Tahoe, NV, USA, December 2013.

[31] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of AAAI*, pp. 1112–1119, Québec City, Canada, December 2014.

[32] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of AAAI*, pp. 2181–2187, Austin, TX, USA, January 2015.

[33] H. Xiao, M. Huang, and X. Zhu, "Transg: a generative model for knowledge graph embedding," in *Proceedings of ACL*, pp. 2316–2325, Berlin, Germany, August 2016.

[34] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of ICML*, pp. 2071–2080, New York, NY, USA, August 2016.

[35] H. Xiao, M. Huang, and X. Zhu, "Ssp: semantic space projection for knowledge graph embedding with text descriptions," in *Proceedings of AAAI*, pp. 3104–3110, San Francisco, CA, USA, February 2017.

[36] B. Shi and T. Weninger, "Proje: embedding projection for knowledge graph completion," in *Proceedings of AAAI*, pp. 1236–1242, San Francisco, CA, USA, February 2017.

[37] T. Dettmers, M. Pasquale, S. Pontus, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of AAAI*, pp. 1811–1818, San Francisco, CA, USA, February 2017.

[38] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proceedings of ACL*, Florence, Italy, July 2019.

[39] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A capsule network-based embedding model for knowledge graph completion and search personalization," in *Proceedings of NAACL*, pp. 2180–2189, Minneapolis, MN, USA, June 2019.

[40] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of NAACL*, pp. 327–333, New Orleansm, LA, USA, June 2018.

[41] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "A three-way model for collective learning on multi-relational data-28th international conference on machine learning," in *Proceedings of the Joint European Conference on Machine Learning*, pp. 809–816, Athens, Greece, September 2011.

[42] Y. Zhao, Z. Li, H. Feng, R. Xie, Q. Li, and P. Gallinari, "Learning entity type structured embeddings with trustworthiness on noisy knowledge graphs," *Knowledge-Based Systems*, vol. 215, 2021.