

Embedding Learning with Triple Trustiness on Noisy Knowledge Graph

Yu Zhao, Huali Feng, Patrick Gallinari

▶ **To cite this version:**

Yu Zhao, Huali Feng, Patrick Gallinari. Embedding Learning with Triple Trustiness on Noisy Knowledge Graph. Entropy, 2019, 21 (11), pp.1083. 10.3390/e21111083 . hal-02430754

HAL Id: hal-02430754

https://hal.sorbonne-universite.fr/hal-02430754

Submitted on 7 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

entropy

MDPI

Article

Embedding Learning with Triple Trustiness on Noisy Knowledge Graph

Yu Zhao [1,*], Huali Feng [1,*] and Patrick Gallinari [2]

[1] Financial Intelligence and Financial Engineering Key Laboratory of Sichuan Province, School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu 611130, China
[2] Laboratoire d'Informatique de Paris 6 (LIP6), Universit Pierre et Marie Curie, 75252 Paris, France; patrick.gallinari@lip6.fr
* Correspondence: zhaoyu@swufe.edu.cn (Y.Z.); fenghuali0723@163.com (H.F.)

Received: 22 October 2019; Accepted: 5 November 2019; Published: 6 November 2019



Abstract: Embedding learning on knowledge graphs (KGs) aims to encode all entities and relationships into a continuous vector space, which provides an effective and flexible method to implement downstream knowledge-driven artificial intelligence (AI) and natural language processing (NLP) tasks. Since KG construction usually involves automatic mechanisms with less human supervision, it inevitably brings in plenty of noises to KGs. However, most conventional KG embedding approaches inappropriately assume that all facts in existing KGs are completely correct and ignore noise issues, which brings about potentially serious errors. To address this issue, in this paper we propose a novel approach to learn embeddings with **triple trustiness** on KGs, which takes possible noises into consideration. Specifically, we calculate the trustiness value of triples according to the rich and relatively reliable information from large amounts of entity type instances and entity descriptions in KGs. In addition, we present a cross-entropy based loss function for model optimization. In experiments, we evaluate our models on KG noise detection, KG completion and classification. Through extensive experiments on three datasets, we demonstrate that our proposed model can learn better embeddings than all baselines on noisy KGs.

Keywords: knowledge graph; embedding learning; cross entropy; noise detection; triple trustiness

> I am convinced that the crux of the problem of learning is recognizing relationships and being able to use them. *Christopher Strachey in a letter to Alan Turing, 1954*

1. Introduction

Knowledge graphs (KGs) provide effective well-structured relational information between entities. A typical KG usually consists of a huge amount of knowledge triples in the form of (*head entity*, *relationship*, *tail entity*) (denoted (h, r, t)), e.g., (*Barack Obama*, *was_born_in*, *Hawaii*). KG embedding aims at learning embeddings of all entities and relationships, which usually are used to promote down-stream knowledge-driven artificial intelligence (AI) and natural language processing (NLP) tasks, such as human-like reasoning, semantic parsing [1], question answering [2,3], relation extraction [4,5], speech generation [6], etc.

The past decade has witnessed great surge in building web-scale KGs, such as Freebase [7], WordNet [8], YAGO [9], DBpedia [10], Google Knowledge Graph [11], and other domain-specific KGs. Recently, open information extraction (Open IE) [12], automatic neural relation extraction [13] and crowd-sourcing mechanism are widely used for KG construction, while these approaches inevitably bring ***noises*** in KG due to insufficient human supervision [14,15]. For instance, the recent open IE model on the benchmark achieves only 24% precision when the recall is 67% [16]. There are some

existing approaches that have been proposed for knowledge graph embedding [17–21]. However, most conventional methods inappropriately assume that all facts in existing KGs are completely correct and ignore noise detection, which can lead to errors as the learning algorithm may treat incorrect facts as true ones. Hence, it is crucial to consider noises in knowledge graph embedding and down-stream tasks. As yet, the basic noise issue on knowledge graph embedding has not attracted enough attention [22]. Recently Xie et al. [23] propose to deal with noisy triples for representation learning. However, the calculation of its confidence value is not straightforward and needs too many intermediate processes with high complexity, especially for global path confidence. Moreover, they discard the external information which could provide rich information for judging triples.

In this paper, we concentrate on learning embeddings on noisy knowledge graphs, which can deal with noises and embed all entries into low dimensional vector space. To address the noise issue, following the translation assumption [17], we propose a novel translating embedding learning approach with ***triple trustiness***, called **TransT**, which takes possible noises into consideration. The trustiness-based framework has been widely studied in the research field such as data mining [24]. Figure 1 demonstrates a brief illustration of our work. KG suffers from noises after automatic construction via OpenIE [12]. Such noises are expected to be considered and detected in learning embeddings with triple trustiness on KGs. For example, there exists a noise <*Hawaii*, *belong_to*, *Indonesia*> brought into the KG by OpenIE, which would be detected and ignored in embedding learning.

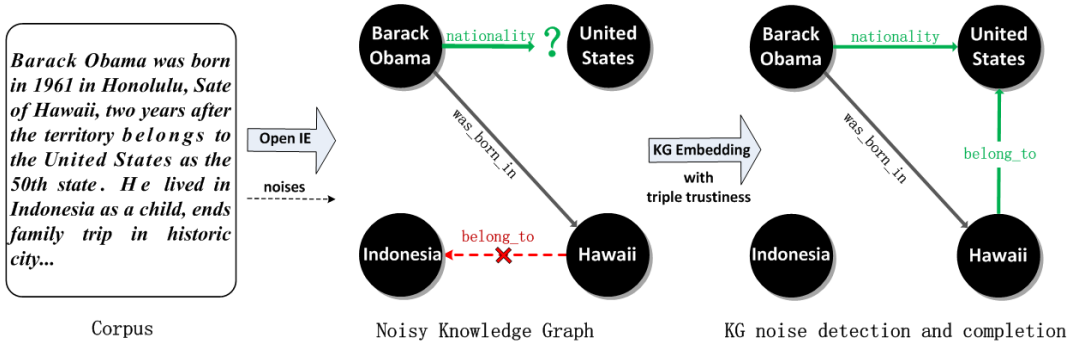


Figure 1. A brief illustration of our work. Knowledge graph faces the noise issue (×) after automatic construction via OpenIE. It's expected to conduct knowledge graph (KG) embeddings learning with triple trustiness for noise detection. For instance, the noise <*Hawaii*, *belong_to*, *Indonesia*> is detected and updated to <*Hawaii*, *belong_to*, *United States*>. Moreover, our noisy KG embedding approach can be used to improve KG completion, such as the incomplete fact <*Barack Obama*, *nationality*, *?*> is completed as a true fact <*Barack Obama*, *nationality*, *United States*> after noise correction, otherwise as a false one <*Barack Obama*, *nationality*, *Indonesia*> according to the noisy triple <*Hawaii*, *belong_to*, *Indonesia*>.

Specifically, TransT calculates the trustiness value of triples by considering two external auxiliary information: entity type instances and entity descriptions, which provide rich pragmatic and semantic information. Correspondingly, we build two novel sub-models for them. Moreover, we present a cross entropy based objective function for training all parameters of our model. We evaluate our model on three tasks including KG noise detection, KG completion, and triple classification. Experimental results demonstrate that our proposed model outperforms all baselines on all tasks, which confirms the capability of TransT in noisy KG embedding. The main work in this paper is concluded as follows:

- We propose a novel translating embedding model, TransT, for learning with triple trustiness on noisy knowledge graph by considering two external information, i.e., entity types and entity descriptions.
- Under this strategy, we propose two sub-models for calculating triple trustiness, one of which is estimated on newly generated entity type triples and another is measured with synthetic entity description triples.

- We present a cross entropy based approach for training model. The experimental results on three noisy datasets including *FB15K-N1, FB15K-N2* and *FB15K-N3* demonstrate the effectiveness of our proposed model.

The structure of the paper is as following. In Section 2, we will provide a brief review of related works. In Section 3, we describe the methodology of our model. In Section 4, we present cross entropy learning method. Section 5 presents experimental results followed by their discussion. Finally, Section 6 gives the conclusion and future directions of this research.

2. Related Work

2.1. Kg Noise Detection

There are various ways of building knowledge graphs, such as edited by crowdsourcing like Freebase [7], extracted from the large-scale semi-structured web like DBpedia [10], and open information extraction methods like Knowledge-Vault [11]. However, all of them inevitably suffer from noise interference due to insufficient human supervision when automatic mechanisms involve. Therefore, noise detection is essential and significant in knowledge automatic construction and knowledge-driven intelligent applications. Most knowledge graph noise detection works happen when constructing knowledge graphs [15,25,26]. These approaches are usually involved with huge human efforts, which are extremely labor-intensive and time-consuming. Recently, there are some works focusing on automatic KG noise detection [27]. Pellissier Tanon et al. [28] select features from contents, users, items [29], and P. et al. [30] propose to judge importance in graphs for nodes and edges. Paulheim and Bizer [31] propose the heuristic link-based type inference mechanism SDType, which can handle noisy and incorrect data. Melo and Paulheim [26] investigate the problem of error detection in relation assertions of knowledge graphs, and 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. Recently, Xie et al. [23] propose an embedding method (CKRL) with confidence to deal with noise detection, however, it ignores the rich semantic information in external nonstructural information which is strong evidence to judge triple quality. In this paper, we propose a knowledge graph embedding learning method with trustiness considering rich auxiliary information.

2.2. Knowledge Graph Embedding

In recent years knowledge graph embedding (see more in this survey [22]) has become a hot research topic. The key idea is to encode all the entities and relations in KG into a latent semantic vector space, so as to predict the probable truth of additional facts purely based on the existing triples in knowledge bases. Various embedding methods have been proposed in recent years. Bordes et al. [32] proposed a structured embedding model (SE), in which the basic idea was to transform the head entity and tail entity into a common latent space by the corresponding left and right projection matrices of the relation and then measured the similarity of the triple by L_1-norm distance in the embedding space. Bordes et al. [33] propose a semantic matching model (SME) for KGC. The main motivation of the model was that entities and predicate relations would share the same form of representation. It mapped all entities and predicate relations into a common latent space to delete the semantic difference between them. Socher et al. [19] propose a neural tensor network model (NTN), which tackled the issue of weak entity vector interaction through replacing a standard linear neural network layer with a bilinear tensor layer that directly relates left entity and right entity across multiple dimensions. The main intuition of the model was that each predicate relation would have different parts of semantic representation. Each slice of the predicate relation tensor was responsible for one class of entity pairs. Bordes et al. [17] propose a translating method (TransE) to model predicate relations by interpreting them as translations operating on the low-dimensional embedding of the left entity and right entity. Wang et al. [18] proposed a translating model (TransH), which builds predicate relation as a hyperplane with a translation operation on it. There are more models to conduct KG

embedding, such as PIDE [34], RESCAL [35], TransG [21], TransR [36], TransD [37], Analogical [38], Convolutional2D [39], ProjE [40], ComplexE [20] and SSP [41], etc.

Moreover, the KG embedding approaches can be roughly classified into two categories according to the information they used: (1) those which learn embeddings only with KG at hand [17,18,36]; (2) those learning embeddings by combining existing KG with external heterogeneous information, e.g., entity hierarchical types [5,42], entity descriptions [41,43,44], plain text [45], and relation paths [46]. However, all these methods assume that all the facts in KG hold without noise, which is unreasonable especially for KGs constructed automatically without sufficient human supervision. In this paper, we concentrate on noisy KG embedding on the basis of the translation-based model (TransE), which is not difficult to be replaced with other enhanced KG embedding model [18,36].

2.3. Knowledge Graph Refinement

Knowledge graph refinement (KGR) is essential after automatic KG construction [11], since the result may never be perfect whichever approach is taken for constructing knowledge graph. Various methods for KGR have been proposed [25], which can differ along three distinct orthogonal dimensions: (i) the overall goal of the method, i.e., completion [17,19] vs. correction [15,26] of KG; (ii) the refinement target (e.g., relations between entities [19], entity types [47]), and (iii) the data used by the approach (i.e., only KG itself [17], or further external information [42,43]). However, most conventional approaches are only used for one goal as yet, while a combination between completion and error detection methods could be of great value [25]. Dong et al. [11] propose a joint approach with both prior knowledge stemmed from KG and external web content to estimate triple quality in KG construction, but lacking flexible ability in scale and reasoning capability without embedding strategy. Jia et al. [48] propose a crisscrossing neural network for KG completion and correction at the same time, while having high complexity and computational cost. In this paper, we introduce the triple trustiness for KGR, by considering the typical external heterogeneous source (i.e., entity type instances and entity descriptions) beyond the KG itself.

3. Methodology

Notation. For each triple (h, r, t), the head entity and tail entity $h, t \in \mathcal{E}$ and the relation $r \in \mathcal{R}$, where $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations respectively. $\mathcal{D} = \{(h, r, t)\}$ stands for the overall training dataset with noises. τ_h and τ_t represent the hierarchical types of head and tail respectively. $\mathcal{T}$ represents the set of all types, $\tau_h, \tau_t \in \mathcal{T}$. d_h and d_t denote the descriptions of head and tail respectively. w represents the keyword in entity descriptions. $\mathcal{W}$ represents the set of keywords, $w \in \mathcal{W}$.

To learn better embeddings on a noisy knowledge graph, we propose a concept triple trustiness for each triple fact. Triple trustiness denotes the evidential reliability of a triple which can be measured with the favor of external nonstructural auxiliary information.

3.1. Translating Embedding Model

To model entity triples, we first present a typical translating embedding model TransE [17], which was proposed to model `relationships` by interpreting them as translations operating on the low-dimensional embedding of the `head entity` and `tail entity`. The scoring function (the lower the better for correct triple) was as follows:

$$\mathbf{M}(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2 , \tag{1}$$

using ℓ_2-norm, $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^\kappa$.

3.2. Translating Embedding with Triple Trustiness

In order to detect noises and learn better embeddings with triple trustiness, we concentrate more on those triples with high evidential trustiness value. Following the translating assumption [17], we build the energy function $\mathbf{E}(\cdot)$ of our translating embedding model with triple trustiness (TransT) as follows:

$$\mathbf{E}(\mathcal{D}) = \sum_{(h,r,t)\in\mathcal{D}} \mathbf{M}(h,r,t) \cdot \mathbf{T}(h,r,t). \tag{2}$$

The TransT energy function includes two parts. The first part $\mathbf{M}(h,r,t)$ represent the distance between head and tail with relation under translation assumption. A lower $\mathbf{M}(h,r,t)$ indicates that the embeddings of entities and relationships of this triple comply with the translation assumption well. We design the triple trustiness $\mathbf{T}(h,r,t)$ as the second part of our energy function. A fact with higher trustiness possesses higher quality, therefore, it should be more reasonably considered in learning embedding. Next we introduce two novel methods to measure triple trustiness according to external auxiliary sources, as in Figure 2.

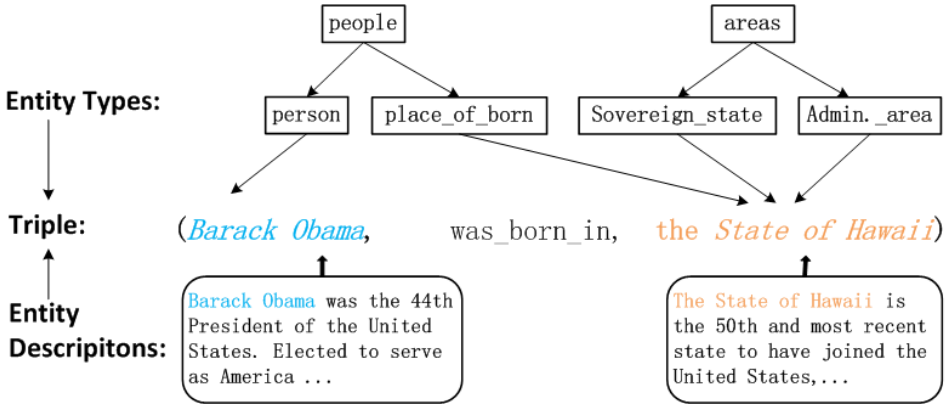


Figure 2. A triple <*Barack Obama, was_born_in, the State of Hawaii*> with its entity hierarchical types and entity descriptions. The left entity "*Barack Obama*" has a hierarchical type: "*/people/person*" and a description: "Barack Obama was the 44h President of the United States...". The right entity "*the State of Hawaii*" possesses similar heterogeneous information.

3.3. Triple Trustiness

In this section, we introduce a novel method to measure triple trustiness with external nonstructural auxiliary information including entity types and entity descriptions.

3.3.1. Triple Trustiness with Entity Types

We first utilize the entity hierarchical types for triple trustiness value estimation. Entity hierarchical types information implies different roles an entity may play in different scenarios [5]. Most typical knowledge graphs (e.g., Freebase [7], DBpedia [10]) have entity type information. Entity types usually consist of hierarchical structures, in which the lower granularity of semantic concepts is considered as the sub-type of entities. Generally, most entities possess more than one hierarchical type. For instance, in Figure 2, *the State of Hawaii* has a variety of types (e.g., */people/place_of_born*, */areas/sovereign_state* and */areas/Administrative_area*) and shows different attributes under different types. The entity hierarchical types are strong evidence to estimate the triple trustiness. For instance, a living thing (Type:*/people/person*) is more credible than a non-living thing (Type: */book/written_work*) when they suppose to be filled in the incomplete triple (*?, was_born_in, the State of Hawaii*). To put it another way, although both triples (*Donald Trump, was_born_in, the State of Hawaii*) and (*Pride and Prejudice, was_born_in, the State of Hawaii*) are not true, but we still believe that the type evidential trustiness of

the former one is higher than the trustiness of the latter due to their distinct types, i.e., the type of *Donald Trump* (*/people/person*) is more reasonable for it.

Entity Type Triple. The key motivation is based on the observation in the research of KG embedding that the learned entity embeddings can be clustered well according to their entity types in the embedding space [34]. For instance, Figure 3 shows that the entity embeddings cluster well according to their entity types represented by different colors [49]. The blue dots indicate the entities with the type: */film/film* and the film entities appear close to each other in the embedding space. Moreover, the more similar between entity types, the more close between corresponding entities in the space, and vice versa. For instance, the group of entities with types: */tv/tv_actor* and */book/author* are closer to each other than entities with other types, and they even show some overlap. These entities share some common types including */person/person*, which is the reason that they are close to each other in the embedding space. Therefore, we believe that one of the premises of a triple (*head entity, relationship, tail entity*) holds is that the corresponding entity types first conform to this relationship. Hence, we build the entity type triple: (*head type, relationship, tail type*) by replacing both head entity and tail entity with their corresponding hierarchical types: $(h, r, t) \rightarrow (\tau_h, r, \tau_t)$.

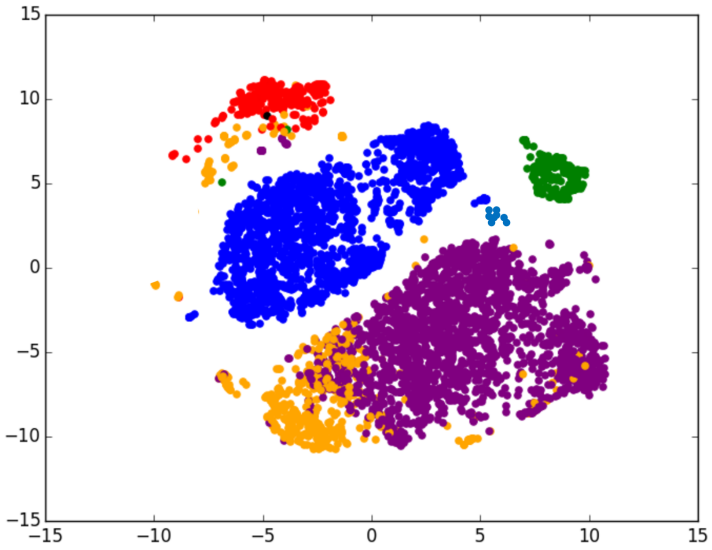


Figure 3. A plot of entities with entity types (Red:/education/educational_institution, Blue:*/film/film*, Purple: */tv/tv_actor*, Orange:/book/author, Green:/tv/tv_program, Black:/music/instrument). Entities with the same entity type tend to appear in well-defined clusters in the embedding space.

Entity Type Embedding. We encode the entity hierarchical type information into representation learning with a general form. Suppose an entity e has hierarchical type: $/\tau_{\mathbf{e}}^{(1)}/\tau_{\mathbf{e}}^{(2)}/.../\tau_{\mathbf{e}}^{(m)}$, m is the number of layers in the hierarchical structure, we utilize the weighted hierarchical embedding (WHE) method, considering that different granularities of sub-type in hierarchical structures may vary in significance in type representation, to build the entity type representation $\boldsymbol{\tau}_e$ as follows:

$$\boldsymbol{\tau}_e = \sum_{i=1}^{m} \beta_i \cdot \boldsymbol{\tau}_e^{(i)} = \beta_1 \cdot \boldsymbol{\tau}_e^{(1)} + \cdots + \beta_m \cdot \boldsymbol{\tau}_e^{(m)}, \tag{3}$$

in which $\boldsymbol{\tau}_e^{(i)}$ is the representation of i-th sub-type $\tau_e^{(i)}$, β_i is the corresponding weight of $\boldsymbol{\tau}_e^{(i)}$.

Entity Type Trustiness (TT). As mentioned above, we have entity type triple (τ_h, r, τ_t) by replacing entity with entity type. We assume that the more a type triple fits the translation assumption, the more convincing the corresponding entity triple should be considered. Hence, the distance $\mathcal{G}(\cdot)$ of entity type triple (τ_h, r, τ_t) under translation-framework with entity type embedding (calculated by (3)), as follows:

$$\mathcal{G}(\tau_h, r, \tau_t) = \| \sum_{i=1}^{m_1} \beta_i \cdot \boldsymbol{\tau}_h^{(i)} + \mathbf{r} - \sum_{j=1}^{m_2} \beta_j \cdot \boldsymbol{\tau}_t^{(j)} \|_2^2 \,, \tag{4}$$

where m_1 and m_2 denote the number of layers in the hierarchical type structure of head entity and tail entity respectively. $\boldsymbol{\tau}_h^{(i)}$ is the representation of i-th sub-type $\tau_h^{(i)}$, $\boldsymbol{\tau}_t^{(j)}$ is the representation of j-th sub-type $\tau_t^{(j)}$, β_i and β_j are the corresponding weight of $\boldsymbol{\tau}_h^{(i)}$ and $\boldsymbol{\tau}_t^{(j)}$ respectively.

To measure the entity type trustiness during training, we first judge the current conformity of each entity type triple with translation assumption. Following margin-based training strategy, we design a function to estimate the type triple quality $\mathcal{Q}_\tau(\tau_h, r, \tau_t)$ as follows:

$$\mathcal{Q}_\tau(\tau_h, r, \tau_t) = -(\gamma_\tau + \mathcal{G}(\tau_h, r, \tau_t) - \mathcal{G}(\tau_h', r, \tau_t')), \tag{5}$$

where $\gamma_\tau > 0$ is a hyperparameter. (τ_h', r, τ_t') is a negative entity type triple in which the head type or tail type is replaced by a random one. A higher $\mathcal{Q}_\tau(\tau_h, r, \tau_t)$ value indicates a better entity type triple judged by the translation framework. All entity type triples are supposed to be correct at the beginning of learning, and set the entity type trustiness $TT(h, r, t) = 1$ for all triples. Since the embeddings of both entity type and relation will be updated constantly in the learning process, the current entity type trustiness for each triple should change according to how much this entity type triple comply with the translation framework. Hence, we utilize the strategy for updating the entity type trustiness $TT(h, r, t)$ according to its type triple quality $\mathcal{Q}_\tau(\tau_h, r, \tau_t)$ as follows:

$$TT(h, r, t) = \begin{cases} \mu \cdot TT(h, r, t), & \mathcal{Q}_\tau(\tau_h, r, \tau_t) \leq 0 \\ \min\{TT(h, r, t) + \nu, 1\}, & \mathcal{Q}_\tau(\tau_h, r, \tau_t) > 0 \,, \end{cases} \tag{6}$$

where $\mu \in (0, 1)$ and $\nu > 0$ are hyper-parameters, $TT(h, r, t) \in (0, 1]$. The condition $\mathcal{Q}_\tau(\tau_h, r, \tau_t) \leq 0$ indicates that the current entity type triple doesn't fit the translation rule well, and thus should cut down the corresponding entity type trustiness, otherwise should increase it when $\mathcal{Q}_\tau(\tau_h, r, \tau_t) > 0$ holds. Hence, a higher $TT(h, r, t)$ implies that the triple is more convinced to hold according to entity type constraints.

3.3.2. Triple Trustiness with Entity Descriptions

In the following, we introduce a novel approach to build triple trustiness with entity descriptions.

Entity Description Triple. TT would fail to work if the types of head and tail exactly match but the fact is actually false, such as *(Donald Trump, was_born_in, the State of Hawaii)*. However, the entity textual descriptions can discover semantic relevance and offer precise semantic expression [41]. The semantic relevance between entities is capable to recognize the true triples, and precise semantic expression could promote the discriminative ability between two triples. Here, we design entity description triple to estimate the triple trustiness by replacing both head and tail with their corresponding descriptions: $(h, r, t) \rightarrow (d_h, r, d_t)$.

Entity Description Embedding. From each short description, we generate a set of keywords, which is capable of capturing the main ideas of entities, based on TFIDF. The assumption is that similar entities should have similar descriptions, and correspondingly have similar keywords. Those triple trustiness may be detected in the internal contact of their keywords. We formulate entity descriptions as $d_e := \{w_1, w_2, ..., w_n\}$. $\{w_1, w_2, ..., w_n\}$ is the set of keywords in entity description. n is the size of words set. We take advantage of convolutional neural network (CNN) [43,50] to model entity description d_e. The CNN model can take word orders, i.e., complicated local interactions of keywords in entity description, into consideration. Specifically, the i-th output vector of convolution layer in CNN is calculated as:

$$\mathbf{z}_i^{(\ell)} = \sigma(\mathbf{W}^{(\ell)} \cdot \mathbf{w}_i^{'(\ell)} + \mathbf{b}_i^{(\ell)}) \,, \tag{7}$$

where $\mathbf{W}^{(\ell)}$ is the convolution kernel for all input vectors of ℓ-th convolution layer after window process and $\mathbf{b}_i^{(l)}$ is the optional bias. σ is the activation function such as *tanh* or *ReLU*. $\mathbf{w}_i^{'(\ell)}$ is the i-th vector of $\mathbf{w}^{'(\ell)}$ which is obtained by concatenating κ column vectors in i-th window of the polling output of (ℓ-1)-th layer. The pooling process shrinks the parameter space of CNN and filter noises after every convolution layer. We use n-max-pooling and mean-pooling strategies respectively in different pooling layers. After the last pooling layer, we obtain the representation of entity description d_e.

Entity Description Trustiness (DT). Under translation-assumption, we build the distance $\mathcal{H}(\cdot)$ of the entity description triple (d_h, r, d_t):

$$\mathcal{H}(d_h, r, d_t) = \|\mathbf{d}_h + \mathbf{r} - \mathbf{d}_t\|_2^2 , \tag{8}$$

where $\mathbf{d}_h$ and $\mathbf{d}_t$ stand for the representation of head descriptions and tail descriptions respectively which are calculated by CNN. To measure the entity description trustiness during training, like the approach in (5), we design a function to estimate the description triple quality $\mathcal{Q}_d(d_h, r, d_t)$ as follows:

$$\mathcal{Q}_d(d_h, r, d_t) = -(\gamma_d + \mathcal{H}(d_h, r, d_t) - \mathcal{H}(d'_h, r, d'_t)), \tag{9}$$

where $\gamma_d > 0$ is a hyperparameter. (d'_h, r, d'_t) is a negative entity description triple in which the head description or tail description is replaced by a random one. Formally, the entity description trustiness $DT(h, r, t)$ changes with its description triple quality $\mathcal{Q}_d(d_h, r, d_t)$ as follows:

$$DT(h, r, t) = \begin{cases} u \cdot DT(h, r, t), & \mathcal{Q}_d(d_h, r, d_t) \leq 0 \\ \min\{DT(h, r, t) + v, 1\}, & \mathcal{Q}_d(d_h, r, d_t) > 0 , \end{cases} \tag{10}$$

where $DT(h, r, t) \in (0, 1]$. A higher DT implies that the triple is more probable to hold according to entity semantic relevance learned by entity descriptions.

3.3.3. Overall Triple Trustiness Model

Here we introduce the overall triple trustiness. Specifically, the overall triple trustiness model combines with two kinds of trustiness stated above: (1) entity types trustiness $TT(h, r, t)$; (2) entity descriptions trustiness $DT(h, r, t)$. Hence, we have overall triple trustiness model $\mathbf{T}(h, r, t)$ as follows:

$$\mathbf{T}(h, r, t) = \lambda \cdot TT(h, r, t) + (1 - \lambda) \cdot DT(h, r, t), \tag{11}$$

where $\lambda \in (0, 1)$ are hyper-parameters.

4. Cross Entropy Loss Function for Optimization

Cross entropy is an important measurement approach of information entropy (IE) (originally proposed by Shannon in [51]). For training the model parameters, we minimize the following binary cross entropy loss function in this work:

$$\boldsymbol{\mathcal{L}}(\boldsymbol{\Theta}) = \{- \sum_{(h,r,t) \in \mathcal{D}} \log p(h, r, t) - \sum_{(h',r,t') \in \mathcal{D}'} \log(1 - p(h', r, t'))\} \cdot \mathbf{T}(h, r, t),$$

in which we apply the logistic sigmoid function $\sigma(\cdot)$ to the model scores, that is $p(h, r, t) = \sigma(-\mathbf{M}(h, r, t))$. Θ are all the parameters of our model including the embeddings of all entities, relations, sub-types, and keywords, i.e., $\Theta = \{\mathbf{E}, \mathbf{R}, \mathbf{T}, \mathbf{W}\}$, initialized randomly. (h, r, t) are the observed triple fact in the training set $\mathcal{D}$ and (h', r, t') are the negative one, the head or tail of which is replaced by a random one. Note that we do not replace both head and tail with random one at the same time. A triple will not be considered as a negative example if it is already in training set $\mathcal{D}$. Here the triple

trustiness $\mathbf{T}(h,r,t)$ are determined by (11), which instructs our model to pay more attention on those more convincing facts. $\mathcal{D}'$ represents the negative triple set.

$$\begin{aligned}\mathcal{D}' := &\{(h',r,t)|(h,r,t) \in \mathcal{D} \cap h' \in \mathcal{E} \cap h' \neq h\} \\ &\cup \{(h,r,t')|(h,r,t) \in \mathcal{D} \cap t' \in \mathcal{E} \cap t' \neq t\}\,.\end{aligned}$$

It is not absolutely necessary to use a entropy loss function [34]. However, it is very common to use entropy loss for learning embeddings (like ConvE [39], FRN [52], etc) just as our model did.

Optimization. We use mini-batch stochastic gradient descent (SGD) for optimization. We perform the following procedure iteratively for a given number of iterations. First, we sample a small set (minibatch) of triples from the training set $\mathcal{D}$, and then for each positive triple in it, we construct a negative sample by replacing the head or tail with a random one. The parameters are then updated by taking a gradient descent step gradually. Algorithm 1 shows the optimization algorithm in detail. As pointed out by [53,54], it would be uneconomical to save all negative properties of an entity or a concept. Therefore, we further require entities to have non-negative vectorial representations. In fact, the distributed representations can be taken as the feature vectors for entities, with latent semantics encoded in different dimensions. To better compare different entities on the same scale, we further require entity representation to stay within the hypercube of $[0,1]^\kappa$, as approximately Boolean embeddings. In most cases, non-negative will further induce sparsity and interpretability.

Algorithm 1 Learning TransT using cross entropy loss function.

Require: Training set $\mathcal{D} = \{(h,r,t)\}$, the set of entity types, entity descriptions.
Ensure: The embeddings of all entities, relations, sub-types, and keywords: $\Theta = \{\mathbf{E}, \mathbf{R}, \mathbf{T}, \mathbf{W}\}$.

Initialize
  $\mathbf{e} \leftarrow Gaussian(0,1)/10$ for each $e \in \mathcal{E}, \mathbf{e} \in \mathbb{R}^\kappa$
  $\mathbf{r} \leftarrow Gaussian(0,1)/10$ for each $r \in \mathcal{R}, \mathbf{r} \in \mathbb{R}^\kappa$
  $\mathbf{t} \leftarrow Gaussian(0,1)/10$ for each $t \in \mathcal{T}, \mathbf{t} \in \mathbb{R}^\kappa$
  $\mathbf{w} \leftarrow Gaussian(0,1)/10$ for each $w \in \mathcal{W}, \mathbf{w} \in \mathbb{R}^\kappa$
Loop
  $\mathcal{D}_{batch} \leftarrow sample(\mathcal{D}, m)$ //minibatch size m
  $\mathcal{A}_{batch} \in \phi$ //initialize training set as null
  for $(h,r,t) \in \mathcal{D}_{batch}$ **do**
    $(h',r,t') \leftarrow sample\ \mathcal{D}$ //corrupted
    $\mathcal{A}_{batch} \leftarrow \mathcal{A}_{batch} \cup ((h,r,t),(h',r,t'))$
  end for
  Update embeddings w.r.t.

$$\sum_{\mathcal{A}_{batch}} \nabla[-\log p(h,r,t) - \log(1 - p(h',r,t'))] \cdot \mathbf{T}(h,r,t)$$

End Loop

5. Experiments

We present three experiments: KG noise detection, KG completion and triple classification to demonstrate the effectiveness of our proposed model. We first introduce the datasets, experimental settings, and baselines for comparison, and then show the experimental results and discussions.

5.1. Datasets

Our experiments are conducted on three public benchmark datasets FB15K-N1, FB15K-N2, and FB15K-N3 (The datasets can be accessed at https://github.com/thunlp/CKRL) which are generated based on FB15K with different noise rates (i.e., 10%, 20%, and 40% respectively) to simulate the real-world KG construction with errors [23]. FB15K [17] is a typical experimental dataset extracted from Freebase. FB15K contains 14951 entities and 1345 relationships, in which all entities possess

descriptions. Moreover, we collect 3851 entity types from FB15kET (The FB15kET can be accessed at https://github.ncsu.edu/cmoon2/kg) [49].

Given a positive triple (h, r, t) in KG, the head or tail is randomly replaced to form a negative one (h', r, t) or (h, r, t'). In order to generate harder and more confusing noises, h' (or t') should have appeared in the head (or tail) position with the same relation, which means that the tail entity of relation *was_born_of* in negative triples should also be a place. All three noisy datasets share the same entities, relations, validation and test sets with FB15K, and all generated negative triples fused into the original training set of FB15K. The statistics are listed in Tables 1 and 2.

Table 1. Statistics of FB15K and FB15kET. FB15kET provides entity type information: (<entity, entity type>). We only use the training data of FB15kET, not valid and test data, to estimate type trustiness.

Dataset	**#Entities**	**#Rel**	**#Train**	**#Valid**	**#Test**
FB15k	14,951	1345	483,142	50,000	59,071
Dataset	**#Ent**	**#Type**	**#Train**	**#Valid**	**#Test**
FB15kET	14,951	3851	136,618	16,000	16,000

Table 2. Statistics of the FB15k-N1, FB15K-N2, FB15K-N3 used for experiments. #Negative triples denotes the number of noises in them.

Datasets	**FB15k-N1**	**FB15k-N2**	**FB15k-N3**
#Negative triples	46,408	93,782	187,925
#Training triples	529,550	576,924	671,067
#Valid triples	50,000	50,000	50,000
#Testing triples	59,071	59,071	59,071

5.2. Experimental Settings and Baselines

In the experiment, we evaluate our TransT model with two different combination strategies. TransT (TT) considers entity type trustiness, while TransT(TT+DT) considers both entity type trustiness and entity description trustiness. We choose two models as the baselines for comparison: (1) TransE which is a typical model used for entity prediction [17], and (2) CKRL which is a state-of-the-art model focusing on representation learning on noisy knowledge graph [23]. The results for the baselines are directly taken from original literature. We train our TransT model using mini-batch SGD. We select the learning rate in the stochastic gradient descent among {0.0001, 0.001, 0.01}, the dimension of entity, relation, entity type, and keyword embedding κ in all models in a range of {50, 100} on the validation set. For overall triple trustiness model, the hyperparameter λ set as 0.5, $\gamma_\tau = \gamma_d = 1$, $\mu = 0.95$, $\nu = 0.05$. For CNN, we set the parameters are: #window size=2, #convolution layer = 2, #dimension of feature map = κ. Usually, $m = 2$ and set $\beta_1 = \beta_2 = 1/m$ in FB15K.

5.3. Kg Noise Detection

To verify the capability of our TransT models in identifying noises in KGs, we conduct a comparative experiment – KG noise detection according to their triple scores.

Evaluation Protocol. We utilize translation-assumption method TransE: $\mathbf{M}(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$ as our triple model. Following the triple classification protocol in [19], we rank all triples in training set with their model score. Therefore, the higher the model score, the more likely the triple is noise. We use precision/recall curves to show the performances.

Experimental Results. Figure 4 demonstrates the evaluation results of KG noise detection, from which we can observe that: (1) Our proposed trustiness-aware model TransTs broadly achieves the best performances on all three datasets with different noise rates, which confirms the capability of our TransT models in modeling tripe trustiness and detecting errors in knowledge graphs. (2) TransT (TT+DT) has an impressive improvement in error detection compared to TransT (TT). It indicates

that the triple trustiness with entity descriptions can provide significant help for error detection. (3) In addition, TransT (TT+DT) has 60∼78% in precision with different noise rates when the recall approximately equals to 40%, which demonstrates the triple trustiness strategy could help for noisy KG embedding. (4) With the noises level rising, TransT (TT+DT) performs better regarding to noise detection. We believe the main reason is that the triples in FB15k-N3 has lower confidence than in FB15k-N1, considering the datasets as bipartite graphs. For instance, there are 671,067 training triples and 187,925 noises in FB15k-N3, while FB15k-N1 only has 529,550 triples and 46,408 noisy triples. Due to higher noise rate, the knowledge in FB15k-N3 can be more easily disturbed by noisy data, which can be significantly detected by our models.

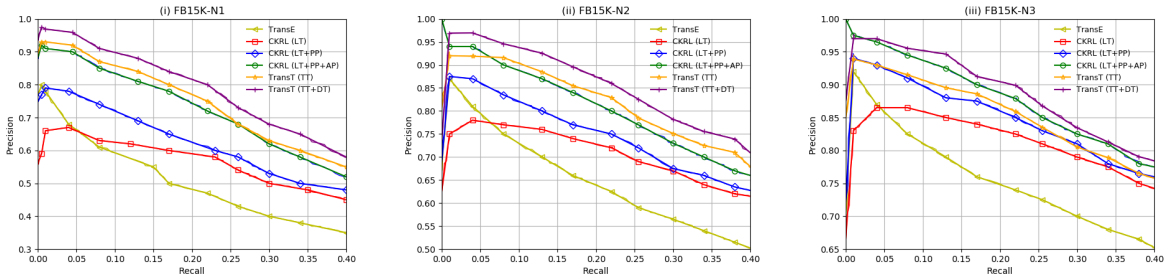


Figure 4. KG noise detection results. Evaluation on FB15K-N1, FB15K-N2, and FB15K-N3.

5.4. Kg Completion

The classical KG completion task concentrates to complete a triple when one of its head, relationship or tail is missing, i.e., to predict how likely some additional triples are held, which aims to verify the capability of our proposed model for KG completion.

Evaluation Protocol. We conduct entity prediction determined by TransE [17]: $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. We use the ranking criteria for evaluation. Firstly for each test triple, we remove the head entity and replace it by each of the entities of the dictionary in turn. The function value $\mathbf{M}(h', r, t)$ of the negative triples would be computed by the related models and then sorted by descending order. We can obtain the exact rank of the correct entity in the candidates. Similarly, we repeat the whole procedure while removing the head entity instead of the tail entity of the test triple. Finally, we use two evaluation metrics for comparison: the mean of those predicted ranks (Mean Rank) and the proportion of correct entities ranked in the top 10 (Hits@10(%)). We also follow the different evaluation settings of "Raw" and "Filter" utilized in [17].

Experimental Results. Table 3 shows the results of entity prediction with different noise rates, from which we observe that: (1) All TransT models achieve better performance compared with the baseline on all noisy datasets, which confirms the capability of our models in KG completion beyond KG noise detection. (2) Our methods achieve more significant improvement as the noise rate increases, compared with basic mode TransE between the three noisy datasets. It verifies that considering the trustiness in noisy KG embedding is very essential especially when KGs have a high rate of noises. Specifically, according to the metrics Mean Rank (Filter) and Hits10(%) (Filter), TrustT (TT+DT) improves (7, 2.5%), (13,3.8%) and (23, 5.3%) on FB15kET-N1, FB15kET-N2, and FB15kET-N3 respectively. (3) TransT (TT+DT) perform better than TransT (TT). It demonstrates that the entity description information could further benefit KG completion especially when TT fails.

Table 3. Entity prediction results. Evaluation of different models on FB15K-N1, FB15K-N2, and FB15K-N3.

Dataset	FB15K-N1				FB15K-N2				FB15K-N3			
Metrics	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	240	144	44.9	59.8	250	155	42.8	56.3	265	171	40.2	51.8
CKRL (LT)	237	140	45.5	61.8	243	146	44.3	59.3	244	148	42.7	56.9
CKRL (LT+PP)	236	139	45.3	61.6	241	144	44.2	59.4	245	149	42.8	56.8
CKRL (LT+PP+AP)	236	138	45.3	61.6	240	144	44.2	59.3	245	150	42.8	56.6
TransT (TT)	233	**137**	45.8	61.2	239	143	44.6	58.1	249	153	42.4	55.2
TransT (TT+DT)	**232**	**137**	**45.9**	**62.3**	**237**	**141**	**45.0**	**60.1**	**246**	**148**	**43.4**	**57.1**

5.5. Triple Classification

Triple classification aims to judge whether a triple in test data holds or not, which could be viewed as a binary classification problem, and also can be regarded as a noise detection task in test data.

Evaluation Protocol. Since there are no explicit negative triples in existing KGs, we build negative triples in validation and test set with an equal number of positive and negative examples. Following the same protocol in [19], we use the validation set to find a threshold ζ. If the model score $||\mathbf{h}+\mathbf{r}-\mathbf{t}|| \leq \zeta$ in classification, the triple will be classified to be true, otherwise to be false. The final accuracy is based on how many triples are classified correctly.

Experimental Results. Table 4 shows the accuracy of the evaluation result of different models. We can find that: (1) The TransT models perform better than the baseline on three datasets, and the improvements become more larger with higher noise rates, which prove that triple trustiness can be helpful for relation triple classification as well. (2) Specifically, TransT (TT+DT) model improves 0.7%, 0.9% and 1.8% on FB15K-N1, FB15K-N2, and FB15K-N3 respectively, it reaffirms that our method becomes more significant with higher noise rates. (3) However, the traditional model TransE may also achieve comparable results, and the improvement our proposed model has over them in this task seems to be unobvious. It may be because our proposed models mainly focus on calculating trustiness for triples in training set, but not for negative triples that are generated in the testing set.

Table 4. Triple classification results. Evaluation of different models on FB15K-N1, FB15KN2 and FB15K-N3.

Dataset	FB15K-N1	FB15K-N2	FB15K-N3
TransE	81.3	79.4	76.9
CKRL(LT)	81.8	80.2	78.3
CKRL(LT+PP)	81.9	80.1	78.4
CKRL(LT+PP+AP)	81.7	80.2	78.3
TransT (TT)	82.2	80.8	79.1
TransT (TT+DT)	**82.4**	**81.1**	**80.1**

6. Conclusions and Future Work

In this paper, we concentrate on noisy knowledge graph embedding with triple trustiness. We consider to estimate the triple trustiness according to the conventional external nonstructural auxiliary information, i.e., entity type instances and entity descriptions. Correspondingly, we propose two sub-models for calculating triple trustiness with entity types and entity descriptions respectively. Through extensive experiments on three real-world datasets, we demonstrate TransT's effectiveness over the baselines. In the future, we will explore the following directions: (1) More external resources can further improve our model. We will explore to combine more external heterogeneous information with internal structural information to further enhance the performance. (2) Network embedding also faces the noise issue. We will apply our proposed framework to improve network embedding as well.

(3) Graph Signal Processing (GSP) [55,56], which aims to generalize the classical signal processing to graph signals, could also benefit from KG embedding approaches as this work proposed.

Author Contributions: Investigation, Y.Z. and H.F.; Methodology, Y.Z. and P.G.; Software, Y.Z., H.F.; Supervision, Y.Z.; Writing—original draft, H.F.; Writing—review and editing, Y.Z. and P.G.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant No. 61906159, in part by the Sichuan Science and Technology Program under Grant No. 2018JY0607, in part by the Fundamental Research Funds for the Central Universities under Grant No. JBK1901030, No. JBK1801080.

Acknowledgments: Thanks to the anonymous reviewers for their helpful comments for improving this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13), Seattle, WA, USA, 18–21 October 2013; pp. 1533–1544.
2. Bordes, A.; Weston, J.; Chopra, S. Question Answering with Subgraph Embeddings. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 615–620.
3. Zheng, W.; Cheng, H.; Yu, J.X.; Zou, L.; Zhao, K. Interactive natural language question answering over knowledge graphs. *Inf. Sci.* **2019**, *481*, 141–159. [CrossRef]
4. Weston, J.; Bordes, A.; Yakhnenko, O.; Usunier, N. Connecting language and knowledge bases with embedding models for relation extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1366–1371.
5. Chang, K.W.; tau Yih, W.; Yang, B.; Meek, C. Typed tensor decomposition of knowledge bases for relation extraction. In Proceedings of the EMNLP, Doha, Qatar, 25–29 October 2014; pp. 1568–1579.
6. Kaminska, D. Emotional Speech Recognition Based on the Committee of Classifiers. *Entropy* **2019**, *21*, 920. [CrossRef]
7. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD'08), Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
8. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
9. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the WWW, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
10. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; et al. DBpedia: A largescale, multilingual knowledge base extracted from Wikipedia. *Semant. Web J.* **2015**, *6*, 167–195.
11. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'14), New York, NY, USA, 24–27 August 2014; pp. 601–610.
12. Banko, M.; Cafarella, M.J.; Soderland, S.; Broadhead, M.; Etzioni, O. Open Information Extraction from the Web. In Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, 6–12 January 2007; pp. 2670–2676.
13. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural Relation Extraction with Selective Attention over Instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16), Berlin, Germany, 7–12 August 2016; pp. 2124–2133.
14. Liang, J.; Xiao, Y.; Zhang, Y.; won Hwang, S.; Wang, H. Graph-Based Wrong IsA Relation Detection in a Large-Scale Lexical Taxonomy. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI'17), San Francisco, CA, USA, 4–9 February 2017; pp. 1178–1184.

15. Heindorf, S.; Potthast, M.; Stein, B.; Engels, G. Vandalism detection in wikidata. In Proceedings of the International on Conference on Information and Knowledge Management (CIKM'16), Indianapolis, IN, USA, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 327–336.
16. Stanovsky, G.; Michael, J.; Zettlemoyer, L.; Dagan, I. Supervised Open Information Extraction. In Proceedings of the NAACL-HLT 2018, New Orleans, LA, USA, 5 June 2018; pp. 885–895.
17. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational data. In Proceedings of the Advances in Neural Information Processing Systems (NIPS'13), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
18. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
19. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning with neural tensor networks for knowledge base completion. In Proceedings of the Advances in Neural Information Processing Systems (NIPS'13), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 926–934.
20. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the 33rd International Conference on Machine Learning (ICML'16), New York, NY, USA, 19–24 June 2016; pp. 2071–2080.
21. Xiao, H.; Huang, M.; Zhu, X. TransG: A generative model for knowledge graph embedding. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16), Berlin, Germany, 7–12 August 2016; pp. 2316–2325.
22. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
23. Xie, R.; Liu, Z.; Lin, F.; Lin, L. Does William Shakespeare REALLY Write Hamlet? Knowledge Representation Learning with Confidence. In Proceedings of the Association for the Advancement of Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
24. Mitra, S.; Pal, S.K.; Mitra, P. Data Mining in Soft Computing Framework: A Survey. *IEEE Trans. Neural Netw.* **2002**, *13*. [CrossRef] [PubMed]
25. Paulheim, H. Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semant. Web* **2017**, *8*, 489–508. [CrossRef]
26. Melo, A.; Paulheim, H. Detection of Relation Assertion Errors in Knowledge Graphs. In Proceedings of the ACM Conference, Tacoma, WA, USA, 18–20 August 2017; p. 22.
27. Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proc. IEEE* **2016**, *104*, 11–33. [CrossRef]
28. Pellissier Tanon, T.; Vrandecic, D.; Schaffert, S.; Steiner, T.; Pintscher, L. From freebase to wikidata: The great migration. In Proceedings of the WWW, Montreal, QC, Canada, 11–15 April 2016; pp. 1419–1428.
29. Gyöngyi, Z.; Garcia-Molina, H.; Pedersen, J. Combating web spam with trust rank. In Proceedings of the VLDB, Toronto, ON, Canada, 31 August–3 September 2004; pp. 576–587.
30. De Meo, P.; Ferrara, E.; Fiumara, G.; Ricciardello, A. A novel measure of edge centrality in social networks. *Knowl.-Based Syst.* **2012**, *30*, 136–150. [CrossRef]
31. Paulheim, H.; Bizer, C. Type Inference on Noisy RDF Data. In Proceedings of the ISWC, Sydney, Australia, 21–25 October 2013; pp. 510–525.
32. Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y. Learning structured embeddings of knowledge bases. In Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11), San Francisco, CA, USA, 7–11 August 2011; pp. 301–306.
33. Bordes, A.; Glorot, X.; Weston, J.; Bengio, Y. A semantic matching energy function for learning with multi-relational data. *Mach. Learn.* **2014**, *94*, 233–259. [CrossRef]
34. Zhao, Y.; Gao, S.; Gallinari, P.; Guo, J. Knowledge base completion by learning pairwise-interaction differentiated embeddings. *Data Min. Knowl. Discov.* **2015**, *29*, 1486–1504. [CrossRef]
35. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the 28th International Conference on Machine Learning (ICML'11), Bellevue, WA, USA, 28 June–2 June 2011; pp. 809–816.

36. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), Austin, TX, USA, 25–29 January 2015; pp. 2181–2187.
37. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL'15), Beijing, China, 26–31 July 2015; pp. 687–696.
38. Liu, H.; Wu, Y.; Yang, Y. Analogical inference for multi-relational embeddings. In Proceedings of the 34th International Conference on Machine Learning (ICML'17), Sydney, Australia, 6–11 August 2017; pp. 2168–2178.
39. Dettmers, T.; Pasquale, M.; Pontus, S.; Riedel, S. Convolutional 2D knowledge graph embeddings. In Proceedings of the 2017 AAAI Conference on Artificial Intelligence (AAAI'17), San Francisco, CA, USA, 4–9 February 2017; pp. 1811–1818.
40. Shi, B.; Weninger, T. ProjE: Embedding projection for knowledge graph completion. In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17), San Francisco, CA, USA, 4–9 February 2017; pp. 1236–1242.
41. Xiao, H.; Huang, M.; Zhu, X. SSP: Semantic space projection for knowledge graph embedding with text descriptions. In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17), San Francisco, CA, USA, 4–9 February 2017; pp. 3104–3110.
42. Xie, R.; Liu, Z.; Sun, M. Representation learning of knowledge graphs with hierarchical types. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16), New York, NY, USA, 9–15 July 2016; pp. 2965–2971.
43. Xie, R.; Liu, Z.; Jia, J.; Luan, H.; Sun, M. Representation learning of knowledge graphs with entity descriptions. In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16), Phoenix, AZ, USA, 12–17 February 2016; pp. 2659–2665.
44. Zhong, H.; Zhang, J.; Wang, Z.; Wan, H.; Chen, Z. Aligning knowledge and text embeddings by entity descriptions. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15), Lisbon, Portugal, 17–21 September 2015; pp. 267–272.
45. Zhang, D.; Yuan, B.; Wang, D.; Liu, R. Joint semantic relevance learning with text data and graph knowledge. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China, 26–31 July 2015; pp. 32–40.
46. Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; Liu, S. Modeling relation paths for representation learning of knowledge bases. In Proceedings of the 2015 EMNLP, Lisbon, Portugal, 17–21 September 2015; pp. 705–714.
47. Nickel, M.; Tresp, V.; Kriegel, H.P. Factorizing yago: scalable machine learning for linked data. In Proceedings of the 21st international conference on World Wide Web (WWW'12), Lyon, France, 16–20 April 2012; ACM: New York, NY, USA, 2012; pp. 271–280.
48. Jia, S.; Xiang, Y.; Chen, X.; E, S. TTMF: A Triple Trustworthiness Measurement Frame for Knowledge Graphs. *arXiv* **2018**, arXiv:1809.09414.
49. Moon, C.; Jones, P.; Samatova, N.F. Learning Entity Type Embeddings for Knowledge Graph Completion. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 2215–2218.
50. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2267–2273.
51. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
52. Komninos, A.; Manandhar, S. Feature-Rich Networks for Knowledge Base Completion. In Proceedings of the ACL, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 324–329.
53. Brian Murphy, P.T.; Mitchell, T. Learning effective and interpretable semantic models using non-negative sparse embedding. In Proceedings of the COLING 2012, Mumbai, India, 8–15 December 2012; pp. 1933–1950.
54. Ding, B.; Wang, Q.; Wang, B.; Guo, L. Improving Knowledge Graph Embedding Using Simple Constraints. In Proceedings of the 56th Annual Meeting of (ACL'18), Melbourne, Australia, 15–20 July 2018; pp. 110–121.

55. Belda, J.; Vergara, L.; Safont, G.; Salazar, A. Computing the Partial Correlation of ICA Models for Non-Gaussian Graph Signal Processing. *Entropy* **2019**, *21*, 22. [CrossRef]
56. Belda, J.; Vergara, L.; Salazar, A.; Safont, G. Estimating the Laplacian matrix of Gaussian mixtures for signal processing on graphs. *Signal Process.* **2018**, *148*, 241–249. [CrossRef]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).