Enhancing Knowledge Graph Embedding from a Logical Perspective

Jianfeng Du[1(✉)], Kunxun Qi[1], Hai Wan[2], Bo Peng[2], Shengbin Lu[1], and Yuming Shen[1]

[1] School of Computer Science and Technology,
Guangdong University of Foreign Studies, Guangzhou 510006, China
jfdu@gdufs.edu.cn
[2] School of Data and Computer Science,
Sun Yat-sen University, Guangzhou 510006, China
wanhai@mail.sysu.edu.cn

Abstract. Knowledge graph embedding aims to represent entities and relations in a knowledge graph as low-dimensional real-value vectors. Most existing studies exploit only structural information to learn these vectors. This paper studies how logical information expressed as RBox axioms in OWL 2 is used for embedding. The involvement of RBox axioms could prevent existing methods from learning predictive vectors. For example, the symmetric, reflexive or transitive relations can be declared by RBox axioms, but popular translation-based methods are unable to learn distinguishable vectors for multiple these relations in the ideal case. To overcome these limitations introduced by the involvement of RBox axioms, this paper proposes to enhance existing translation-based methods by logical pre-completion and bi-directional projection of entities. Experimental results demonstrate that these enhancements improve the predictive performance in link prediction and triple classification.

1 Introduction

Knowledge graph, a kind of directed graph with vectors labeled by entities and edges labeled by relations, has become a popular formalism for knowledge representation. Meanwhile, knowledge graph embedding, a technology that typically represents entities and relations as low-dimensional real-value vectors, has shown promising results on a number of prediction tasks including link prediction [5,6], triple classification [19,22] and relation extraction [22,23]. Up to date, most existing studies for knowledge graph embedding e.g. [3–7,10–16,18,19,22,25] exploit only structural information of the knowledge graph to learn entity or relation vectors. Seldom work focuses on using other information in the learning process. It remains an open question if other information could be used in the learning process to improve the predictive performance.

Logical information is often attached to a knowledge graph. OWL 2 [8], the newest version of the Web Ontology Language, has become prevalent in

Z. Wang et al. (Eds.): JIST 2017, LNCS 10675, pp. 232–247, 2017.
https://doi.org/10.1007/978-3-319-70682-5_15

knowledge representation. It introduces RBox, expressed as a set of axioms, to declare characteristics of relations and interactions among multiple relations. For example, a relation can be declared to be symmetric, reflexive or transitive, whereas one relation can be declared to be subsumed by another relation. Many existing knowledge graphs such as WordNet [17] and DBPedia [1] come with OWL 2 ontologies that have RBox axioms.

A knowledge graph is usually represented as a set of triples of the form $\langle h, r, t\rangle$, where h is the head entity, r the relation and t the tail entity.[1] When a knowledge graph comes with RBox axioms, the embedding of triples in it should conform to the logical constraints enforced by these RBox axioms. For example, consider a knowledge graph that has a triple $\langle h, r, t\rangle$ where r is declared to symmetric by an axiom attached to the graph. Knowledge graph embedding requires the embedding of $\langle h, r, t\rangle$ to satisfy a certain condition. According to logical inference, the knowledge graph entails the triple $\langle t, r, h\rangle$, thus the embedding of $\langle t, r, h\rangle$ should also satisfy the same condition.

Recently, translation-based methods become popular in knowledge graph embedding due to their high predictive performance and good applicability in sparse knowledge graphs. They represent entities and relations as low-dimensional real-value vectors and typically enforce that $||f_r(\boldsymbol{h})+\boldsymbol{r}-f_r(\boldsymbol{t})||_{L_1/L_2}$ should be less than $||f_r(\boldsymbol{h'})+\boldsymbol{r}-f_r(\boldsymbol{t'})||_{L_1/L_2}$ by a margin, where $\boldsymbol{x}$ denotes the vector representation for x, $||\boldsymbol{x}||_{L_1}$ (resp. $||\boldsymbol{x}||_{L_2}$) denotes the L1-norm (resp. L2-norm) of the vector $\boldsymbol{x}$, $\langle h, r, t\rangle$ (resp. $\langle h', r, t'\rangle$) is a triple in (resp. not in) the given knowledge graph, and $f_r(\cdot)$ is a projection function on entity vectors which is either an identity function (e.g. in TransE [5], TransM [7] and TransA [14]), or a r-specific function (e.g. in TransH [22] and TransR [16]) which may also depend on the given entity (e.g. in TransD [12]) or on the statistical triple distribution (e.g. in TranSparse [13]).

In the ideal case, $||f_r(\boldsymbol{h})+\boldsymbol{r}-f_r(\boldsymbol{t})||_{L_1/L_2}$ declines to 0 for all triples $\langle h, r, t\rangle$ in the given knowledge graph. According to logical inference, $||f_r(\boldsymbol{h})+\boldsymbol{r}-f_r(\boldsymbol{t})||_{L_1/L_2}$ should also decline to 0 for all triples $\langle h, r, t\rangle$ entailed by the knowledge graph. We find that, when the relation r is symmetric, reflexive or transitive, the vector representation $\boldsymbol{r}$ declines to an all-0 vector in the ideal case, thus the vectors for multiple symmetric, reflexive or transitive relations are not distinguishable. Moreover, when the projection function $f_r(\cdot)$ is the identity function, for two relations r_1 and r_2 such that r_1 is subsumed by r_2, the vector representations $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are also not distinguishable in the ideal case. To overcome these limitations, we propose to enhance existing methods that use $||f_r(\boldsymbol{h})+\boldsymbol{r}-f_r(\boldsymbol{t})||_{L_1/L_2}$ as the loss function by logical pre-completion and bi-directional projection of entities. Logical pre-completion enlarges the given knowledge graph by all entailed triples according to the attached RBox, so that learnt vectors conform to the logical constraints enforced by the RBox. Bi-directional projection of entities revises the loss function to

[1] The head entity, relation and tail entity are respectively called the *subject*, *predicate* and *object* in OWL/RDF terminology. We use terminology in the field of knowledge graph embedding rather than OWL/RDF terminology throughout the paper.

$||f_r(\boldsymbol{h}) + \boldsymbol{r} - f'_r(\boldsymbol{t})||_{L_1/L_2}$ for $f_r(\cdot)$ and $f'_r(\cdot)$ two different functions, so that the above limitations are resolved.

We conducted experiments on four benchmark datasets that are widely used in the field of knowledge graph embedding. Experimental results show that the enhancements for TransE, TransH, TransR and TransD improve the predictive performance in two tasks namely link prediction and triple classification.

2 Preliminaries

2.1 OWL 2 RBox and Knowledge Graph

OWL 2 [8] is the newest version of the W3C proposed Web Ontology Language. It is a formal language with rigorous syntactics and semantics, underpinned by decidable fragments of first-order logic (FOL). Due to a good balance between expressivity and computational complexity, OWL 2 has become popular in both academy and industry. An OWL 2 ontology usually has an RBox and a TBox so as to express logical information in real-life applications. The relations in OWL 2 are mostly object properties or data properties. The RBox consists of axioms declaring characteristics of relations and interactions among multiple relations, whereas the TBox consists of axioms declaring interactions among multiple classes possibly with the help of relations.

A knowledge graph can often be expressed as a set of triples of the form $\langle h, r, t\rangle$, where h is called the *head entity* (simply *head*), r is called the *relation*, t is called the *tail entity* (simply *tail*). Many publicly available knowledge graphs such as WordNet [17] and DBPedia [1] come with OWL 2 ontologies. In such a scenario, the knowledge graph and the attached OWL 2 ontology constitute a first-order logic program, where a triple $\langle h, r, t\rangle$ is translated to a unary fact $t(h)$ if r is $\mathsf{rdf:type}$, or to a binary fact $r(h, t)$ otherwise. It should be noted that only triples on $\mathsf{rdf:type}$ involve classes; i.e., for a triple $\langle h, r, t\rangle$, t is a class if and only if r is $\mathsf{rdf:type}$. Since the number of triples on $\mathsf{rdf:type}$ is often much smaller than that of other triples in a knowledge graph, in this work we do not consider triples on $\mathsf{rdf:type}$. Accordingly, since the triples that we consider do not involve classes, we do not consider TBox axioms either. We leave the consideration of classes in knowledge graph embedding with OWL 2 as our future work.

A relation in an OWL 2 RBox can be declared as a primary relation or an inverse relation. An inverse relation is of the form $\mathsf{ObjectInverseOf}(r)$ for r a primary relation. We assume that knowledge graphs use only primary relations. The syntax and semantics for RBox axioms are shown in Table 1. The semantics is given by translating axioms to rules in FOL under the *unique name assumption* (*UNA*). UNA ensures that two entities having different names are semantically different. It is often adopted in real-life applications. A rule R in FOL is of the form $\mathbf{B} \rightarrow h$, where $\mathbf{B}$ is called the *body* of R, written $\mathsf{body}(R)$, which is a conjunction of atoms or empty, and h is called the *head* of R, written $\mathsf{head}(R)$), which is an atom or empty. The natural meaning of RBox axioms can easily be seen from their syntactic representation. To name only a few, $\mathsf{SymmetricObjectProperty}(r)$ means that r is symmetric, whereas

Table 1. The syntax and semantics for RBox axioms in OWL 2

Functional-style syntax	Semantics via translation to FOL
$\mathsf{SymmetricObjectProperty}(r)$	$\mathsf{ar}(x,r,y) \rightarrow \mathsf{ar}(y,r,x)$
$\mathsf{AsymmetricObjectProperty}(r)$	$\mathsf{ar}(x,r,y) \wedge \mathsf{ar}(y,r,x) \rightarrow$
$\mathsf{ReflexiveObjectProperty}(r)$	$\rightarrow \mathsf{ar}(x,r,x)$
$\mathsf{IrreflexiveObjectProperty}(r)$	$\mathsf{ar}(x,r,x) \rightarrow$
$\mathsf{TransitiveObjectProperty}(r)$	$\mathsf{ar}(x,r,y) \wedge \mathsf{ar}(y,r,z) \rightarrow \mathsf{ar}(x,r,z)$
$\mathsf{FunctionalObjectProperty}(r)$	$\mathsf{ar}(x,r,y_1) \wedge \mathsf{ar}(x,r,y_2) \wedge y_1 \neq y_2 \rightarrow$
$\mathsf{InverseFunctionalObjectProperty}(r)$	$\mathsf{ar}(x_1,r,y) \wedge \mathsf{ar}(x_2,r,y) \wedge x_1 \neq x_2 \rightarrow$
$\mathsf{DisjointObjectProperties}(r,s)$	$\mathsf{ar}(x,r,y) \wedge \mathsf{ar}(x,s,y) \rightarrow$
$\mathsf{SubObjectPropertyOf}(r,s)$	$\mathsf{ar}(x,r,y) \rightarrow \mathsf{ar}(x,s,y)$
$\mathsf{SubObjectPropertyOf}(\mathsf{ObjectPropertyChain}(p,q),r)$	$\mathsf{ar}(x,p,y) \wedge \mathsf{ar}(y,q,z) \rightarrow \mathsf{ar}(x,r,z)$

Note: x, y and z are universally quantified variables; $\mathsf{ar}(x,r,y)$ denotes $s(y,x)$ if r is $\mathsf{ObjectInverseOf}(s)$, or $r(x,y)$ otherwise.

Table 2. The projection functions in different translation-based methods

Method	Projection function
TransE	$f_r(\boldsymbol{x}) = \boldsymbol{x}$ for $\boldsymbol{x}, \boldsymbol{r} \in \mathbb{R}^n$
TransH	$f_r(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{w}_r^{\mathrm{T}} \boldsymbol{x} \boldsymbol{w}_r$ for $\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{w}_r \in \mathbb{R}^n$
TransR	$f_r(\boldsymbol{x}) = \boldsymbol{M}_r \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{r} \in \mathbb{R}^m$ and $\boldsymbol{M} \in \mathbb{R}^{m \times n}$
TransD	$f_r(\boldsymbol{x}) = (\boldsymbol{p}_r \boldsymbol{p}_x^{\mathrm{T}} + \boldsymbol{D})\, \boldsymbol{x}$ for $\boldsymbol{x}, \boldsymbol{p}_x \in \mathbb{R}^n$, $\boldsymbol{r}, \boldsymbol{p}_r \in \mathbb{R}^m$, and $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ being a diagonal matrix whose principal diagonal is $\mathbf{1}^{\min(m,n)}$
TranSparse (share or separate version)	$f_r(\boldsymbol{x}) = \boldsymbol{M}_{r,\theta_r} \boldsymbol{x}$ (share) or $f_r(\boldsymbol{h}) = \boldsymbol{M}_{r,\theta_{r1}} \boldsymbol{h}$ and $f_r(\boldsymbol{t}) = \boldsymbol{M}_{r,\theta_{r2}} \boldsymbol{t}$ (separate) for $\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^n$, $\boldsymbol{r} \in \mathbb{R}^m$, $\boldsymbol{M}_{r,\theta} \in \mathbb{R}^{m \times n}$ a sparse matrix having $\lfloor \theta mn \rfloor$ non-zero elements, where $\theta_r = 1 - (1 - \theta_{\mathsf{min}}) \frac{\#\{(h',t') \mid \langle h',r,t' \rangle \in \mathcal{G}\}}{\max_{r'} \#\{(h',t') \mid \langle h',r',t' \rangle \in \mathcal{G}\}}$, $\theta_{r1} = 1 - (1 - \theta_{\mathsf{min}}) \frac{\#\{h' \mid \langle h',r,t' \rangle \in \mathcal{G}\}}{\max_{r'} \#\{h' \mid \langle h',r',t' \rangle \in \mathcal{G}\}}$, $\theta_{r2} = 1 - (1 - \theta_{\mathsf{min}}) \frac{\#\{t' \mid \langle h',r,t' \rangle \in \mathcal{G}\}}{\max_{r'} \#\{t' \mid \langle h',r',t' \rangle \in \mathcal{G}\}}$, $\mathcal{G}$ is the given knowledge graph, θ_{min} is a hyper-parameter, and $\#S$ denotes the cardinality of the set S

$\mathsf{SubObjectPropertyOf}(r,s)$ means that r is subsumed by s. The union of a knowledge graph $\mathcal{G}$ and an OWL 2 RBox $\mathcal{R}$ can be translated to a function-free Horn logic program, denoted by $\pi(\mathcal{G} \cup \mathcal{R})$, which inherits the standard FOL semantics. We say a triple $\langle h,r,t \rangle$ is *entailed by* by a knowledge graph $\mathcal{G}$ w.r.t. an OWL 2 RBox $\mathcal{R}$ if all models of $\pi(\mathcal{G} \cup \mathcal{R})$ are also models of $r(h,t)$.

2.2 Translation-Based Methods in Knowledge Graph Embedding

There have emerged a number of methods for embedding a knowledge graph into a continuous vector space, called *knowledge graph embedding*, to tackle prediction

problems such as link prediction [5,6], triple classification [19,22] and relation extraction [22,23]. Among these methods, the translation-based methods, which represent entities as points and relations as translation from head entities to tail entities in the vector space, become very popular recently as they work well for sparse knowledge graphs with high predictive performance. TransE [5] is the pioneer translation-based method which defines translation directly on entity vectors. Most subsequent translation-based methods define translation on projection of entity vectors. TransH [22] defines projection on relation specific hyperplanes. TransR [16] defines projection by relation specific matrices. TransD [12] defines projection by relation-entity specific matrices. TranSparse [13] defines projection by sparse matrices that depend on relation specific triple distributions.

All the above methods finally learn entity vectors and relation vectors by minimizing a global margin-based loss function with the help of a loss function for triples. The loss function for a triple $\langle h, r, t\rangle$, written $\mathsf{loss}_r(h, t)$, can be uniformly written as $||f_r(\boldsymbol{h}) + \boldsymbol{r} - f_r(\boldsymbol{t})||_{L_1/L_2}$, where $\boldsymbol{x}$ denotes the vector representation $(x_1, \ldots, x_n)$ for x, $||\boldsymbol{x}||_{L_1}$ is the L1-norm of $\boldsymbol{x}$ defined as $\sum_{i=1}^{n} |x_i|$, $||\boldsymbol{x}||_{L_2}$ is the L2-norm of $\boldsymbol{x}$ defined as $\sqrt{\sum_{i=1}^{n} x_i^2}$, and $f_r(\cdot)$ is a projection function defined using the formulae given by Table 2. The global margin-based loss function to be minimized is defined on the set Δ of triples in the given knowledge graph and a randomly generated set $\overline{\Delta}$ of triples that are not in the given knowledge graph, where $\overline{\Delta}$ is constructed from Δ by randomly corrupting triples in either heads or tails; i.e., all elements of $\overline{\Delta}$ are randomly selected from

$$\{\langle h', r, t\rangle \notin \Delta \mid h' \neq h, \langle h, r, t\rangle \in \Delta\} \cup \{\langle h, r, t'\rangle \notin \Delta \mid t' \neq t, \langle h, r, t\rangle \in \Delta\}.$$

By introducing the margin γ as a hyper-parameter, the global margin-based loss function to be minimized can be uniformly written as

$$\sum_{\langle h,r,t\rangle \in \Delta} \sum_{\langle h',r,t'\rangle \in \overline{\Delta}} \max(0, \gamma + \mathsf{loss}_r(h, t) - \mathsf{loss}_r(h', t')).$$

This function is commonly minimized, with all entity vectors, relation vectors and parameters on the projection function learnt, by stochastic gradient descent (SGD), where hyper-parameters are determined by a validation set.

Some other translation-based methods such as TransM [7] and TransA [14] are slightly extended from TransE. TransM defines the loss function for a triple $\langle h, r, t\rangle$ as $c_r||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_{L_1/L_2}$, where c_r is the r-specific weight for loss functions. TransA directly computes the margin γ according to some statistics on the training set instead of determining γ by a validation set.

3 Enhancements for Translation-Based Methods

When a knowledge graph comes with an OWL 2 RBox, the embedding of triples in it should conform to the logical constraints enforced by the RBox. In other words, what should be satisfied by all triples *in* the knowledge graph should also be satisfied by all triples *entailed by* the knowledge graph w.r.t. the attached

Algorithm. `ComputeEntailedTripleSet`$(\mathcal{G}, \mathcal{R})$
Input: A knowledge graph $\mathcal{G}$ and an attached OWL 2 RBox $\mathcal{R}$.
Output: The set of triples entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$.
1: Let Π be obtained from $\pi(\mathcal{G} \cup \mathcal{R})$ by modifying rules to safe rules;
2: Let $\Delta' = \emptyset$ and $\Delta = \{h \mid\!\rightarrow h \in \Pi\}$;
3: **while** $\Delta \neq \Delta'$ **do**
4: Let $\Delta' = \Delta$;
5: **for** each $R \in \Pi$ and each ground substitution σ for $\mathsf{body}(R)$ such that $\mathsf{head}(R)$ is not empty and $\mathsf{body}(R)\,\sigma \subseteq \Delta'$ **do**
6: Let $\Delta = \Delta \cup \{\mathsf{head}(R)\,\sigma\}$;
7: **return** $\{\langle h, r, t\rangle \mid r(h, t) \in \Delta\}$;

Fig. 1. The algorithm for computing all entailed triples

RBox. As summarized in Subsect. 2.2, a translation-based method requires that $\gamma + \mathsf{loss}_r(h, t) - \mathsf{loss}_r(h', t')$ be as small as possible for a triple $\langle h, r, t\rangle$ in the given knowledge graph and a triple $\langle h', r, t'\rangle$ not in the given knowledge graph. Hence $\gamma + \mathsf{loss}_r(h, t) - \mathsf{loss}_r(h', t')$ should also be as small as possible for a triple $\langle h, r, t\rangle$ entailed by the given knowledge graph and a triple $\langle h', r, t'\rangle$ not entailed by the given knowledge graph w.r.t. the attached RBox.

Let $\mathcal{G}$ denote the given knowledge base, $\mathcal{R}$ the attached RBox and Δ^+ the set of triples entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$. To enhance translation-based methods with RBox axioms, we only need to modify the global margin-based loss function to

$$\sum_{\langle h,r,t\rangle \in \Delta^+} \sum_{\langle h',r,t'\rangle \in \overline{\Delta^+}} \max(0, \gamma + \mathsf{loss}_r(h, t) - \mathsf{loss}_r(h', t')) \tag{1}$$

without modifying the loss function for triples, where $\overline{\Delta^+}$ is a set of triples not entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, constructed from Δ^+ by randomly corrupting triples in either heads or tails; i.e., all its elements are randomly selected from

$$\{\langle h', r, t\rangle \notin \Delta^+ \mid h' \neq h, \langle h, r, t\rangle \in \Delta^+\} \cup \{\langle h, r, t'\rangle \notin \Delta^+ \mid t' \neq t, \langle h, r, t\rangle \in \Delta^+\}.$$

To accomplish this enhancement, we give an algorithm in Fig. 1 for computing Δ^+, the set of triples entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$. Simply speaking, the algorithm modifies all rules in $\pi(\mathcal{G} \cup \mathcal{R})$ to *safe rules*, which are rules such that all variables appearing in the head also appear in the body, then iteratively computes newly entailed facts according to these safe rules and previously entailed facts until a fix-point is reached, and finally returns the set of triples translated from entailed binary facts. A safe rule is computed from the original rule by adding the atom $\mathsf{dom}(x)$ to the body for every variable x in the head but not in the body, and by adding the fact $\mathsf{dom}(e)$ to $\pi(\mathcal{G} \cup \mathcal{R})$ for every entity e in $\mathcal{G}$, where dom is a new globally unique predicate symbol.

Since this enhancement computes the set of entailed triples once before applying a translation-based method, we call it *logical pre-completion*. The following theorem shows that `ComputeEntailedTripleSet`$(\mathcal{G}, \mathcal{R})$ is correct.

Theorem 1. *Given a knowledge graph $\mathcal{G}$ and an OWL 2 RBox $\mathcal{R}$,* `ComputeEntailedTripleSet`*$(\mathcal{G}, \mathcal{R})$ returns the set of triples entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$.*

Proof (sketch). Let Π be obtained from $\pi(\mathcal{G} \cup \mathcal{R})$ by modifying rules to safe rules. Since $\pi(\mathcal{G} \cup \mathcal{R})$ is a function-free Horn logic program, Π has a unique minimal model M_{min} while `ComputeEntailedTripleSet`$(\mathcal{G}, \mathcal{R})$ returns $\{\langle h, r, t\rangle \mid r(h, t) \in M_{\mathsf{min}}\}$. Thus, a triple $\langle h, r, t\rangle$ is entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$ if and only if $r(h, t) \in M$ for all models M of Π if and only if $r(h, t) \in M_{\mathsf{min}}$. □

Although logical pre-completion can guarantee knowledge graph embedding to meet the logical constraints enforced by the attached RBox, it also has some side effects. We find that this enhancement leads to undistinguishable learnt vectors in the ideal case where the loss value $\mathsf{loss}_r(h, t)$ declines to 0 for all entailed triples $\langle h, r, t\rangle$. The ideal case amounts to a sufficient condition under which the revised global margin-based loss function given by Formula (1) is minimized to 0.

The first side effect is that when multiple symmetric, reflexive, or transitive relations exist, the learnt vectors of these relations must decline to all-0 vectors and cannot be distinguished from each other in the ideal case, as shown in the following theorem.

Theorem 2. *For a knowledge graph $\mathcal{G}$, an OWL 2 RBox $\mathcal{R}$ and a relation r appearing in $\mathcal{G}$ such that (1)* $\mathsf{SymmetricObjectProperty}(r) \in \mathcal{R}$ *and there exists $\langle h, r, t\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, or (2)* $\mathsf{ReflexiveObjectProperty}(r) \in \mathcal{R}$*, or (3)* $\mathsf{TransitiveObjectProperty}(r) \in \mathcal{R}$ *and there exist $\langle e_1, r, e_2\rangle$ and $\langle e_2, r, e_3\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, if $\mathsf{loss}_r(h, t) = ||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_{L_1/L_2} = 0$ for all triples $\langle h, r, t\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, then $\boldsymbol{r} = \mathbf{0}^m$ where $\boldsymbol{r}$ is the vector representation for r and m is the dimension of relation vectors.*

Proof. (1) Consider the case where $\mathsf{SymmetricObjectProperty}(r) \in \mathcal{R}$. For every triple $\langle h, r, t\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, $\langle t, r, h\rangle$ is also entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$. Thus $f_r(\boldsymbol{h}) + \boldsymbol{r} - f_r(\boldsymbol{t}) = f_r(\boldsymbol{t}) + \boldsymbol{r} - f_r(\boldsymbol{h}) = \mathbf{0}^m$. It follows that $\boldsymbol{r} = \mathbf{0}^m$. (2) Consider the case where $\mathsf{ReflexiveObjectProperty}(r) \in \mathcal{R}$. The triple $\langle e, r, e\rangle$ must be entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$ for every entity e in $\mathcal{G}$. Thus $f_r(\boldsymbol{e}) + \boldsymbol{r} - f_r(\boldsymbol{e}) = \mathbf{0}^m$, i.e., $\boldsymbol{r} = \mathbf{0}^m$. (3) Consider the case where $\mathsf{TransitiveObjectProperty}(r) \in \mathcal{R}$. For any entities e_1, e_2 and e_3 such that $\langle e_1, r, e_2\rangle$ and $\langle e_2, r, e_3\rangle$ are entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, $\langle e_1, r, e_3\rangle$ is also entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$. Hence $f_r(\boldsymbol{e_1}) + \boldsymbol{r} - f_r(\boldsymbol{e_2}) = f_r(\boldsymbol{e_2}) + \boldsymbol{r} - f_r(\boldsymbol{e_3}) = f_r(\boldsymbol{e_1}) + \boldsymbol{r} - f_r(\boldsymbol{e_3}) = \mathbf{0}^m$. It follows that $f_r(\boldsymbol{e_1}) + 2\,\boldsymbol{r} - f_r(\boldsymbol{e_3}) = f_r(\boldsymbol{e_1}) + \boldsymbol{r} - f_r(\boldsymbol{e_3}) = \mathbf{0}^m$ and thus $\boldsymbol{r} = \mathbf{0}^m$. □

Since TransE [5], TransH [22], TransR [16], TransD [12], TranSparse(share) [13], TransM [7] and TransA [14] use projection function of the form $f_r(\cdot)$, these methods have the first side effect when enhanced by logical pre-completion.

The second side effect is that when a relation is subsumed by another relation, the vector representations for these two relations learnt by a translation-based method that uses the identity function as the projection function are not distinguishable in the ideal case, as shown in the following theorem.

Theorem 3. *For a knowledge graph $\mathcal{G}$, an OWL 2 RBox $\mathcal{R}$ and two relations r_1 and r_2 such that* $\mathsf{SubObjectPropertyOf}(r_1, r_2) \in \mathcal{R}$ *and there exists $\langle h, r_1, t\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, if $\mathsf{loss}_r(h,t) = ||\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}||_{L_1/L_2} = 0$ for all triples $\langle h, r, t\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, then $\boldsymbol{r_1} = \boldsymbol{r_2}$ where $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are respectively the vector representations for r_1 and r_2.*

Proof. For a triple $\langle h, r_1, t\rangle$ entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$, $\langle h, r_2, t\rangle$ is also entailed by $\mathcal{G}$ w.r.t. $\mathcal{R}$. Therefore $\boldsymbol{h} + \boldsymbol{r_1} - \boldsymbol{t} = \boldsymbol{h} + \boldsymbol{r_2} - \boldsymbol{t}$ and thus $\boldsymbol{r_1} = \boldsymbol{r_2}$. □

Since TransE [5], TransM [7] and TransA [14] use the identity function as the projection function, these methods have the second side effect when enhanced by logical pre-completion.

To resolve the above two side effects, we further enhance existing translation-based methods by revising the projection function. We only consider enhancing translation-based methods whose projection function is of the form $f_r(\cdot)$ and is not the identity function. These methods include TransH [22], TransR [16], TransD [12] and TranSparse(share) [13]. It can be seen from the proof of Theorem 2 that, when the projection functions on head entities and on tail entities are different, the first side effect does not exist. Hence we modify the loss function for triples $\langle h, r, t\rangle$ to $\mathsf{loss}_r(h,t) = ||f_r(\boldsymbol{h}) + \boldsymbol{r} - f'_r(\boldsymbol{t})||_{L_1/L_2}$, where $f_r(\cdot)$ and $f'_r(\cdot)$ are two projection functions composed of different sets of parameters to be learnt. For example, to enhance TransH (see Table 2) we define $f_r(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{w}_r^{\mathrm{T}}\boldsymbol{x}\boldsymbol{w}_r$ and $f'_r(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{w'}_r^{\mathrm{T}}\boldsymbol{x}\boldsymbol{w'}_r$, where $\boldsymbol{w}_r$ and $\boldsymbol{w'}_r$ are two different sets of parameters. We call this enhancement *bi-directional projection of entities*.

4 Experiments

We empirically evaluate the two enhancements for four most popular translation-based methods, i.e. TransE [5], TransH [22], TransR [16] and TransD [12], in link prediction and triple classification. By TransX+ we denote the method enhanced from TransX by logical pre-completion, and BiTransX+ the method enhanced from TransX+ by bi-directional projection of entities, where $\mathrm{X} \in \{\mathrm{E}, \mathrm{H}, \mathrm{R}, \mathrm{D}\}$. Since TransE+ employs the identity function as the projection function on entity vectors, it cannot apply the bi-directional projection enhancement, thus we only consider BiTransX+ for $\mathrm{X} \in \{\mathrm{H}, \mathrm{R}, \mathrm{D}\}$. To guarantee a fair comparison, we implemented all TransX, TransX+ and BiTransX+ methods with multi-threads in Java, using standard SGD with fixed mini-batch size 1, and evaluated them in the RapidMiner platform[2] to ensure all methods to be compared in the same environment. The implementations of TransE/H/R were modified from the C++ code available at the KB2E repository[3] [16], whereas the implementation of TransD was based on [12] but employs standard SGD instead. The evaluation results obtained by our reimplementations will be slightly different from those reported in the literature, because a different implementation language was used here and the mini-batch size in SGD was tuned in original implementations.

[2] https://rapidminer.com/.
[3] https://github.com/thunlp/KB2E/.

Table 3. Datasets and their attached RBoxes

Dataset	#relation	#entity	#train	#valid	#test	#sym	#ref	#trans	#subs
WN18	18	40,943	141,442	5,000	5,000	3	0	0	0
FB15k	1,345	14,951	483,142	50,000	59,071	95	33	1,112	566
WN11	11	38,696	112,581	2,609	10,544	0	0	0	0
FB13	13	75,043	316,232	5,908	23,733	0	0	8	0

4.1 Data Sets

We collected four datasets commonly used in evaluating translation-based methods, which were built from two knowledge graphs WordNet [17] and Freebase [2]. Two datasets come from WordNet. They are WN18 provided by [5] for link prediction and WN11 provided by [19] for triple classification. Another two come from Freebase. They are FB13 provided by [19] for triple classification and FB15k provided by [5] for link prediction and triple classification. We composed RBoxes for these datasets by generating SymmetricObjectProperty (**sym**metricity), ReflexiveObjectProperty (**ref**lexivity), TransitiveObjectProperty (**trans**itivity) and SubObjectPropertyOf (**subs**umption) axioms. In more details, a candidate axiom α is generated from a dataset $\mathcal{G}$ if $\mathsf{head}(R)\,\sigma$ is included in $\mathcal{G}$ for all ground substitutions σ for R that make $\mathsf{body}(R)\,\sigma$ included in $\mathcal{G}$, where R is the FOL rule translated from α according to Table 1. All the candidate axioms are then manually filtered by their natural meanings to form an RBox. The statistics for all the datasets and their attached RBoxes are listed in Table 3.[4]

4.2 Link Prediction

Link prediction aims to predict the missing head h or the missing tail t for a triple $\langle h, r, t\rangle$. This task focuses more on ranking a set of candidate entities from the knowledge graph for each missing position rather than computing only one best entity. As set up in previous work [5,7,12–14,16,22], we conduct experiments on two datasets WN18 and FB15k.

Evaluation Protocol. We follow a similar protocol originally given by [5]. For each test triple $\langle h, r, t\rangle$, we replace the tail t with an arbitrary entity e in the knowledge graph and calculate the loss value $\mathsf{loss}_r(h, e)$ on the corrupted triple $\langle h, r, e\rangle$. By ranking the loss values in ascending order, we can get the rank of the original triple. Similarly, we can get another rank for $\langle h, r, t\rangle$ by corrupting the head h. Since a corrupted triple is also correct if it is entailed by the knowledge graph w.r.t. the attached RBox, ranking it before the original triple should not be considered wrong. To eliminate this factor, we remove those corrupted triples that are entailed by the union of the training set, the validation set and the test set w.r.t. the attached RBox before getting the rank of every test triple. In

[4] The test datasets and their attached RBoxes as well as the RapidMiner platform with test processes are available at http://www.dataminingcenter.net/JIST17/.

this setting two metrics are reported: the mean rank averaged by all test triples (denoted as *Mean*) and the proportion of ranks not larger than 10 (denoted as *Hits@10*). A lower Mean is better while a higher Hits@10 is better.

Implementation. We tune the hyper-parameters in a similar way presented in [5,12,16,22]. We select the learning rate λ in standard SGD among $\{0.001, 0.005, 0.01\}$, the margin γ among $\{0.25, 0.5, 1, 2\}$, the dimensions of entity embedding k and relation embedding d among $\{20, 50, 100\}$, and the dissimilarity measure as either L1-norm or L2-norm. The optimal hyper-parameters are determined by seeking the lowest Mean in the validation set. Following [22], we adopt two strategies for sampling negative triples in every method. By unif we denote the traditional way of replacing head or tail with equal probability, and by bern we denote the way of replacing head or tail with different probabilities proportional to the average frequency for entities in head or entities in tail. The most prevalent optimal configurations are: $\lambda = 0.005$, $\gamma = 1$, $k = d = 100$ and using L2-norm on WN18 under both unif and bern strategies; $\lambda = 0.001$, $\gamma = 2$, $k = d = 100$ and using L1-norm on FB15k under the unif strategy; $\lambda = 0.001$, $\gamma = 1$, $k = d = 100$ and using L1-norm on FB15k under the bern strategy. The other optimal configurations include: $\lambda = 0.01$, $\gamma = 0.5$, $k = d = 100$ with L2-norm for training TransR+ (bern), BiTransR+ (bern), TransD (bern), TransD+ (bern) and BiTransD+ (bern) on WN18; $\lambda = 0.001$, $\gamma = 1$, $k = d = 100$ with L2-norm for training TransD+ (bern) on FB15k. For training TransE/H, TransE+/H+ and BiTransH+ on both datasets, we traverse all the training triples for 2000 rounds. For training TransR/D (resp. TransR+/D+ and BiTransR+/D+) on both datasets, in order to speed up the convergence and avoid overfitting, we initialize the entity vectors and the relation vectors with the results of TransE (resp. TransE+) and traverse all the training triples for another 500 rounds.

Results. The experimental results for all methods in terms of Mean and Hits@10 are reported in Table 4, where the best performance in each metric on each dataset is highlighted in bold. It can be seen that every TransX+ method consistently outperforms its baseline TransX method except for two cases, namely TransR+ (bern) on WN18 and TransD+ (bern) on FB15k. It shows that the unif strategy for sampling negative triples is more stable than the bern strategy in learning vectors. The performance improvements gained by TransX+ can be explained by that the logical pre-completion enhancement feeds up TransX+ with more training triples than TransX and that in TransX+ none of the entailed triples is treated as negative triples. Furthermore, in most cases the BiTransX+ method achieves at least the same good performance as its former TransX+ method. It shows that the bi-directional projection enhancement is effective in further improving the performance in link prediction.

To further demonstrate the effectiveness of both enhancements in link prediction, we separate the relations in FB15k into four categories, namely **sym**metric, **ref**lexive, **trans**itive and other relations that are not symmetric, reflexive, or transitive. The experimental results on different relation categories of FB15k in terms of macro average Hits@10, calculated as the mean of Hits@10 for every relation in a specific category, are reported in Table 5, where the highest Hits@10

Table 4. The performance for link prediction (H@10 is Hits@10 in percentage)

Method	TransX				TransX+				BiTransX+			
Dataset	WN18		FB15k		WN18		FB15k		WN18		FB15k	
Metric	Mean	H@10	Mean	H@10	Mean	H@10	Mean	H@10	Mean	H@10	Mean	H@10
X = E(unif)	320	90.7	53	75.4	314	91.4	52	75.7	–	–	–	–
X = E(bern)	324	91.1	97	75.7	316	91.8	91	75.8	–	–	–	–
X = H(unif)	324	91.1	52	75.5	323	91.7	**51**	75.6	318	91.7	**51**	75.7
X = H(bern)	314	91.2	99	75.7	300	91.8	97	76.2	273	90.4	97	76.2
X = R(unif)	330	91.4	68	75.5	319	91.6	67	76.1	325	94.0	68	76.2
X = R(bern)	**27**	90.8	98	76.2	143	86.4	64	77.1	118	87.6	60	**77.8**
X = D(unif)	322	90.7	55	75.9	320	91.2	54	76.1	317	91.2	56	76.1
X = D(bern)	395	92.9	96	76.1	46	**98.6**	149	62.9	45	**98.6**	65	76.8

Table 5. The performance for link prediction on FB15k separated by different relation categories (measured by macro average Hits@10 in percentage)

Method	TransX				TransX+				BiTransX+			
Relation Category	sym	ref	trans	other	sym	ref	trans	other	sym	ref	trans	other
X = E(unif)	100	100	74.1	79.3	100	100	74.2	80.1	–	–	–	–
X = E(bern)	100	100	76.5	80.2	100	100	76.7	80.2	–	–	–	–
X = H(unif)	100	100	74.8	78.8	100	100	75.0	79.0	100	100	75.0	80.1
X = H(bern)	100	100	76.5	79.6	100	100	76.9	80.8	100	100	76.9	80.8
X = R(unif)	100	100	75.2	80.0	100	100	76.4	80.4	100	100	75.2	82.1
X = R(bern)	100	100	77.7	80.8	100	100	86.3	86.7	100	100	**87.3**	**87.1**
X = D(unif)	100	100	75.5	80.1	100	100	75.2	80.4	100	100	75.5	80.5
X = D(bern)	100	100	77.8	80.0	100	100	80.0	77.0	100	100	85.7	85.7

on each category is highlighted in bold. This table reveals some wierd cases where the macro average Hits@10 in average category is larger than the mean Hits@10 displayed in Table 4. These cases are possible since the mean Hits@10 in Table 4 is triple-wise calculated and is not a weighted average of the four relation-wise calculated macro average Hits@10s. It can be seen that the TransX+ method generally outperforms its baseline TransX method while the BiTransX+ method also generally outperforms its former TransX+ method in both the transitive category and the other category. For the symmetric category and the reflexive category, all methods achieve the highest (100%) Hits@10.

4.3 Triple Classification

Triple classification aims to judge whether a triple $\langle h, r, t\rangle$ is correct or not. It is a binary classification task. We conduct experiments on all the four datasets.

Evaluation Protocol. We follow a similar protocol originally given by [19]. The evaluation of triple classification requires negative triples that are not entailed by the dataset w.r.t. the attached RBox. Although WN11 and FB13, released by [19], already contain negative triples that were built by corrupting the corresponding observed (positive) triples, these negative triples may be entailed by

the dataset w.r.t. the attached RBox since the procedure for generating negative triples [19] does not consider RBox axioms. To guarantee generated negative triples to be logically valid, we construct negative triples for all the four datasets instead of reusing existing ones provided by [19]. The generation procedure is almost the same as the way presented in [19] except that a generated negative triple is required to be not entailed by the union of the training set, the validation set and the test set w.r.t. the attached RBox. The decision rule for triple classification is simple: given a triple $\langle h, r, t\rangle$, if its loss value $\mathsf{loss}_r(h, e)$ is not larger than a relation specific threshold θ_r, the triple will be classified as positive; otherwise, it will be classified as negative. The relation specific threshold θ_r is determined by maximizing the classification accuracy on the validation set.

Implementation. We tune the hyper-parameters for all datasets in a similar way presented in [5,12,16,22]. The search space of hyper-parameters is identical to that in link prediction. The optimal hyper-parameters are determined by seeking the highest classification accuracy in the validation set. Following [22], we also adopt two strategies unif and bern for sampling negative triples in every method. The optimal configurations for FB15k are the same as those in link prediction. For other datasets, under both strategies the most prevalent optimal configurations are: $\lambda = 0.01$, $\gamma = 1$, $k = d = 100$ and using L2-norm on WN18; $\lambda = 0.01$, $\gamma = 2$, $k = d = 20$ and using L2-norm on WN11; $\lambda = 0.01$, $\gamma = 2$, $k = d = 100$ and using L2-norm on FB13. The other optimal configurations include: $\lambda = 0.005$, $\gamma = 1$, $k = d = 100$ with L2-norm for training TransE+ (unif), TransR (unif), TransR+ (unif), TransD (unif) and TransD+ (unif) on WN18; $\lambda = 0.005$, $\gamma = 2$, $k = d = 20$ with L2-norm for training TransE (unif/bern), TransE+ (unif/bern), TransR (unif), TransR+ (unif) and BiTransR+ (unif/bern) on WN11; $\lambda = 0.005$, $\gamma = 2$, $k = d = 20$ with L1-norm for training BiTransD+ (unif/bern) on WN11; $\lambda = 0.005$, $\gamma = 2$, $k = d = 100$ with L2-norm for training TransE (bern), TransE+ (unif/bern), TransH+ (unif/bern), BiTransH+ (bern) and BiTransR+ (unif) on FB13; $\lambda = 0.001$, $\gamma = 1$, $k = d = 100$ with L2-norm for training TransR (unif), TransR+ (unif), TransD (unif), TransD+ (unif) and BiTransD+ (unif) on FB13; $\lambda = 0.001$, $\gamma = 2$, $k = d = 100$ with L1-norm for training TransD (bern), TransD+ (bern) and BiTransD+ (bern) on FB13. On every dataset, we traverse all the training triples for the same number of rounds as in link prediction.

Results. The experimental results are reported in Table 6, where the highest accuracy on each dataset is highlighted in bold. It can be seen that every TransX+ method still outperforms its baseline TransX method except for three cases including TransE+ (bern) on WN18, TransE+ (bern) on FB15k, and TransD+ (bern) on FB13. The same accuracy on WN11 is due to the emptiness of the RBox attached to WN11, which implies that the set of entailed triples is the same as the set of observed triples on WN11. Moreover, for 13 out of 24 cases the BiTransX+ method achieves at least the same high accuracy as its former TransX+ method. Also, for three datasets the highest accuracy is achieved by a

Table 6. The performance for triple classification (measured by classification accuracy in percentage)

Method	TransX				TransX+				BiTransX+			
Dataset	WN18	FB15k	WN11	FB13	WN18	FB15k	WN11	FB13	WN18	FB15k	WN11	FB13
X = E(unif)	97.5	86.6	81.6	74.2	97.6	87.2	81.6	74.5	–	–	–	–
X = E(bern)	97.6	85.2	80.6	74.8	97.5	85.2	80.6	77.0	–	–	–	–
X = H(unif)	97.5	86.7	82.7	74.0	97.6	87.5	82.7	74.4	97.6	87.3	**83.1**	74.4
X = H(bern)	97.4	85.3	80.9	75.6	97.6	85.7	80.9	**78.1**	86.3	85.3	82.2	72.4
X = R(unif)	97.6	85.0	82.0	63.4	**97.7**	85.9	82.0	63.5	97.5	**87.8**	82.2	64.5
X = R(bern)	96.0	85.3	81.3	73.7	97.1	85.9	81.3	74.2	96.5	87.1	81.0	56.2
X = D(unif)	97.6	86.5	82.5	66.1	**97.7**	86.9	82.5	66.5	**97.7**	87.0	76.9	66.9
X = D(bern)	97.0	86.6	81.5	65.0	97.3	70.0	81.5	59.6	97.5	87.0	74.7	59.0

Table 7. The performance for triple classification on FB15k separated by different relation categories (measured by macro average accuracy in percentage)

Method	TransX				TransX+				BiTransX+			
Relation Category	sym	ref	trans	other	sym	ref	trans	other	sym	ref	trans	other
X = E(unif)	89.3	89.3	77.4	70.2	91.3	91.3	75.8	70.9	–	–	–	–
X = E(bern)	85.8	85.8	77.5	68.2	87.5	87.5	74.8	68.4	–	–	–	–
X = H(unif)	95.8	95.8	76.5	70.3	86.7	86.7	75.3	71.3	87.5	87.5	75.6	70.8
X = H(bern)	**100**	**100**	75.1	68.3	**100**	**100**	77.4	68.9	86.4	86.4	74.9	68.7
X = R(unif)	64.8	64.8	79.9	67.7	62.8	62.8	73.4	69.6	86.4	86.4	**80.2**	**71.5**
X = R(bern)	66.4	66.4	74.1	68.5	67.6	67.6	75.5	69.5	87.5	87.5	77.1	70.5
X = D(unif)	65.7	65.7	73.2	70.1	69.1	69.1	74.4	70.2	87.2	87.2	75.1	70.4
X = D(bern)	65.9	65.9	73.0	70.3	75.0	75.0	61.7	58.8	86.5	86.5	77.2	70.4

certain BiTransX+ method. These results show that the bi-directional projection enhancement is also effective in improving the classification accuracy.

To further demonstrate the effectiveness of both enhancements in triple classification, we also separate the relations in FB15k into four categories, namely **sym**metric, **ref**lexive, **trans**itive and other relations. The experimental results on different relation categories of FB15k in terms of macro average accuracy, calculated as the mean of classification accuracy for every relation in a specific category, are reported in Table 7, where the best accuracy on each category is highlighted in bold. It can be seen that, the TransX+ method generally outperforms its baseline TransX method while the BiTransX+ method also generally outperforms its former TransX+ method, although the improvement is not so significant as that for Hits@10 in link prediction.

5 Related Work

The translation-based method is a hot research direction for knowledge graph embedding. TransE [5] assumes $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ for all triples $\langle h, r, t \rangle$ in the knowledge graph. To better embed 1-N, N-1 and N-N relations, TransH [22], TransR [16], TransD [12] and TranSparse [13] explore different projection functions on entity

vectors. TransM [7] extends TransE by introducing relation specific weights on loss functions for triples. TransA [14] extends TransE by an automatic way to determine the optimal margin. We refer the interested reader for more details on the above method in Subsect. 2.2. There are also some other translation-based methods. Unstructured method [3,4] is a naive version of TransE where $\boldsymbol{r}$ is fixed to an all-0 vector. PTransE [15] extends TransE by using reliable relation paths as translations between entities, and by developing a path-constraint resource allocation algorithm to measure the reliability of relation paths.

Another research direction for knowledge graph embedding is the energy-based method, which assigns low energies to plausible triples in the knowledge graph and employs neural network for learning. For example, Structured Embedding (SE) [6] defines two relation specific matrices for head entity and tail entity, and learns entity vectors in a neural network architecture. Other energy-based methods include Semantic Matching Energy (SME) [3,4], Latent Factor Model (LFM) [11], Single Layer Model (SLM) and Neural Tensor Network (NTN) [19].

There also exist some methods for knowledge graph embedding that are not in the above directions. RESCAL [18] is a typical matrix-factorization based method, which treats a knowledge graph as a 3-model tensor and learns the latent representation, namely an entity as a vector and a relation as a matrix, by reconstructing the original graph. KG2E [10] is density-based method which represents entities and relations by Gaussian distributions. TransG [25] is a generative method which leverages a mixture of relation specific component vectors to embed a triple so as to capture multiple relation semantics.

All the above methods only exploit the structure information on the knowledge graph to embed triples. External information such as text information and logical information can also be used to improve the embedding results. One kind of text information namely entity names and their corresponding Wikipedia anchors is used in the probabilistic TransE (pTransE) method [21] to embed both triples and words. Another kind of text information namely entity descriptions is used in the DKRL method [26] to embed both triples and descriptions.

Prior to this work, the logical information considered in knowledge graph embedding is restricted to rules. In [20] four kinds of rules are encoded into an integer linear programming problem to filter triples predicted by an embedding model; i.e., all rules are used in postprocess and will not impact embedding results. In [9] two kinds of rules of the form $b \rightarrow h$ and $b_1 \wedge b2 \rightarrow h$ are incorporated into an enhanced TransE method called KALE. This method instantiates given rules to true ground rules and introduces a loss function for ground rules, so that the global margin-based loss function to be minimized is summed up not only from the margin-based loss difference between positive triples and sampled negative triples but also from the margin-based loss difference between positive ground rules and sampled negative ground rules.

This work is the first work for exploiting OWL 2 RBox as logical information to embed triples. Although RBox axioms can be translated to rules, we enhance translation-based methods by computing all entailed triples rather than instantiating rules as in the KALE method [9]. In KALE a rule could be instantiated to

quadratic number of ground rules in the number of entities since it only requires at least one instantiated body atom to appear in the knowledge graph. The instantiation of rules raises a scalability problem for large rule sets. In addition, the study in [9] has not considered side effects of the enhancement, while we consider side effects and propose solutions to resolve them.

6 Conclusions and Future Work

In order to use logical information represented by an OWL 2 RBox in knowledge graph embedding, we have proposed two enhancements to existing translation-based methods in this paper. The first enhancement called *logical pre-completion* enlarges the given knowledge graph by all entailed triples according to an attached RBox, so that the embedding results conform to the logical constraints enforced by the RBox. The second enhancement called *bi-directional projection of entities* allows the projection function on head entities to be different from the projection function on tail entities, so that the learnt relation vectors are guaranteed to vary with the different relations. Experimental results demonstrate the effectiveness of these enhancements in improving the predictive performance.

For future work, we plan to further extend translation-based methods to embed entity type triples, which are triples of the form $\langle h, \mathsf{rdf:type}, t\rangle$. We will study three sources of information for this extension. The first source is an OWL 2 ontology attached to the given knowledge graph. The second source is the heuristic information studied in the field of ontology learning [24] for predicting entity types. The last source is external links to other knowledge graphs, such as external links from entities to Wordnet [17] synsets.

Acknowledgements. This work was partly supported by National Natural Science Foundation of China under grants 61375056 and 61573386, Natural Science Foundation of Guangdong Province under grant 2016A030313292, Guangdong Province Science and Technology Plan projects under grant 2016B030305007, Sun Yat-sen University Cultivation Project (16lgpy40) and the Undergraduate Innovative Experiment Project in Guangdong University of Foreign Studies (201711846022).

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. J. Web Semant. **7**(3), 154–165 (2009)
2. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of SIGMOD, pp. 1247–1250 (2008)
3. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In: Proceedings of AISTATS, pp. 127–135 (2012)
4. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Mach. Learn. **94**(2), 233–259 (2014)

5. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NIPS, pp. 2787–2795 (2013)
6. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Proceedings of AAAI (2011)
7. Fan, M., Zhou, Q., Chang, E., Zheng, T.F.: Transition-based knowledge graph embedding with relational mapping properties. In: Proceedings of PACLIC, pp. 328–337 (2014)
8. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: the next step for OWL. J. Web Semant. **6**(4), 309–322 (2008)
9. Guo, S., Wang, Q., Wang, L., Wang, B., Guo, L.: Jointly embedding knowledge graphs and logical rules. In: Proceedings of EMNLP, pp. 192–202 (2016)
10. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with Gaussian embedding. In: Proceedings of CIKM, pp. 623–632 (2015)
11. Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A Latent factor model for highly multi-relational data. In: Proceedings of NIPS, pp. 3176–3184 (2012)
12. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of ACL, pp. 687–696 (2015)
13. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Proceedings of AAAI, pp. 985–991 (2016)
14. Jia, Y., Wang, Y., Lin, H., Jin, X., Cheng, X.: Locally adaptive translation for knowledge graph embedding. In: Proceedings of AAAI, pp. 992–998 (2016)
15. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of EMNLP, pp. 705–714 (2015)
16. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of AAAI, pp. 2181–2187 (2015)
17. Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D., Miller, K.: WordNet: an online lexical database. Int. J. Lexicogr. **3**(4), 235–244 (1990)
18. Nickel, M., Tresp, V., Kriegel, H.: Factorizing YAGO: scalable machine learning for linked data. In: Proceedings of WWW, pp. 271–280 (2012)
19. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of NIPS, pp. 926–934 (2013)
20. Wang, Q., Wang, B., Guo, L.: Knowledge base completion using embeddings and rules. In: Proceedings of IJCAI, pp. 1859–1866 (2015)
21. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: Proceedings of EMNLP, pp. 1591–1601 (2014)
22. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI, pp. 1112–1119 (2014)
23. Weston, J., Bordes, A., Yakhnenko, O., Usunier, N.: Connecting language and knowledge bases with embedding models for relation extraction. In: Proceedings of EMNLP, pp. 1366–1371 (2013)
24. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: a look back and into the future. ACM Comput. Surv. **44**(4), 20:1–20:36 (2012)
25. Xiao, H., Huang, M., Zhu, X.: TransG: A generative model for knowledge graph embedding. In: Proceedings of ACL, pp. 992–998 (2016)
26. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of AAAI, pp. 2659–2665 (2016)