Embedding Knowledge Graphs Based on Transitivity and Asymmetry of Rules

Mengya Wang, Erhu Rong, Hankui Zhuo, and Huiling Zhu

Sun Yat-sen University,
No.132, Waihuan East Road, Guangzhou, China,
{wangmy38, rongerhu}@mail2.sysu.edu.cn
{zhuohank, zhuhling6}@mail.sysu.edu.cn

Abstract. Representation learning of knowledge graphs encodes entities and relation types into a continuous low-dimensional vector space, learns embeddings of entities and relation types. Most existing methods only concentrate on knowledge triples, ignoring logic rules which contain rich background knowledge. Although there has been some work aiming at leveraging both knowledge triples and logic rules, they ignore the transitivity and asymmetry of logic rules. In this paper, we propose a novel approach to learn knowledge representations with entities and ordered relations in knowledges and logic rules. The key idea is to integrate knowledge triples and logic rules, and approximately order the relation types in logic rules to utilize the transitivity and asymmetry of logic rules. All entries of the embeddings of relation types are constrained to be non-negative. We translate the general constrained optimization problem into an unconstrained optimization problem to solve the non-negative matrix factorization. Experimental results show that our model significantly outperforms other baselines on knowledge graph completion task. It indicates that our model is capable of capturing the transitivity and asymmetry information, which is significant when learning embeddings of knowledge graphs.

Keywords: Knowledge graph, Logic rules, Non-negative matrix factorization, Transitivity, Asymmetry

1 Introduction

Knowledge graphs (KGs) store rich information of the real world in the form of graphs, which consist of nodes (entities) and labelled edges (relation types between entities) (e.g., *(Trump,PresidentOf,USA)*). This sort of structured data can be interpreted by computers and applied in various fields such as information retrieval [4] and word sense disambiguation [2]. Although powerful in representing structured data, the symbolic nature of relations makes KGs, especially large-scale KGs, difficult to manipulate. Predicting missing entries (known as link prediction) is of great importance in knowledge graph. To do this task, vector space embeddings of knowledge graphs have been widely adopted. The key idea is to embed entities and relation types of a KG into a continuous vector space.

Many approaches have been proposed to learn embeddings of entities and relations, such as TransE [5], NTN [22], HOLE [6], ComplEx [7], and so on. They, however, only focus on knowledge triples, ignoring rich knowledge from logic rules. Logic rules are taken as complex formulae constructed by combining atoms with logical connectives. To leverage logic rules in knowledge graph embeddings, [18–20] propose to utilize both knowledge triples and logic rules for KB completion. In their works, however, logic rules need to be grounded. Each rule needs to be instantiated with concrete entities. A rule can be grounded into plenty of ground rules, since there are many entities connected by the same relation type in logic rules. As a result, the works cannot scale well to larger KGs since rules will be grounded with more entities. And also, they all neglect the properties of transitivity and asymmetry of rules.

*For example, suppose we have two rules: "CapitalOf $\Rightarrow$ LocatedIn" and "LocatedIn $\Rightarrow$ ContainedBy", which indicates the relation that x is a **capital of** y implies another relation that x is **located in** y, and the relation that x is **located in** y implies x is **contained by** y. Provided that we know Paris is a **capital of** France from the knowledge base, we can infer that Paris is **located in** France and **contained by** France as well, according to the **transitivity** of rules. In addition, even though we know Paris is **located in** France, we cannot infer that Paris is a **capital of** France according to the **asymmetry** of rules: "CapitalOf $\Rightarrow$ LocatedIn" $\neq$ "LocatedIn $\Rightarrow$ CapitalOf".*

To leverage the properties of transitivity and asymmetry of rules, we propose a novel knowledge representation learning model to capture the ordering of relations, and infer potential new relations based on the ordering of existing relations and properties of transitivity and asymmetry of rules. We integrate knowledge graphs, existing relations and logic rules together to learn the knowledge graph embeddings. Logic rules are incorporated into relation type representations directly, rather than instantiated with concrete entities. The embeddings learned are therefore compatible not only with triples but also with rules, and the embeddings of relation types are approximately ordered. We call our learning approach `TARE`, short for **E**mbedding knowledge graphs based on **T**ransitivity and **A**symmetry of **R**ules.

In the remainder of the paper, we first review previous work related to our work. Then we formulate our problem and present our learning algorithm in detail. After that we evaluate our approach by comparing our approach with exiting state-of-the-art algorithms. Finally we conclude the paper with future work.

2 Related Work

Many works have made great efforts on modelling knowledge graphs. Some works explain triples via latent representation of entities and relations such as tensor factorization [29, 8, 9] and multiway neural networks [22]. The key of relational latent feature models is that the relationship between entities can be derived from interactions of their latent features. Many ways are discovered to model these interactions. RESCAL [29] is a relational latent feature model which explains triples via pairwise interactions of latent features. However, it requires a large number of parameters. TransE [5] translates the latent feature representations via a relation-specific offset. Both entities and relations are projected into the same continuous low-dimensional vector space and relations are interpreted as translating operations between head and tail entities. TransE is efficient when modelling simple relations. To improve the performance of TransE on complicated relations, TransH [10], TransR [11] and TransD [17] are proposed. Unfortunately, these models miss simplicity and efficiency of TransE. To combine the power of tensor product with the efficiency and simplicity of TransE, HOLE [6] uses the circular correlation of vectors to represent pairs of entities. Circular correlation has the advantage, comparing to tensor product, that it does not increase the dimensionality of the composite representation. However, due to the asymmetry of circular correlation, HOLE is unable to deal with symmetric relation. Complex [7] makes use of embeddings with complex value and is able to handle a large number of binary relations, in particular symmetric and antisymmetric relations. Some works such as [30], [12], [31] learn embeddings by sampling long paths($e0 \rightarrow e_1 \rightarrow e_2 ... \rightarrow e_n$) in KG. They learn the transitivity of relation triples but not the transitivity of rules.

These models perform the embedding task based solely on triples contained in a KG. Recent work put growing interest in logic rules. [18] tries to utilize rules via integer linear programming or Markov logical networks. However, rules are modeled separately from embedding models and will not help obtaining better embeddings. [19] proposes a joint model which injects first-order logic into embeddings. This work focus on the relation extraction task and created vector embeddings for entity pairs rather than individual entities. As a result, relations between unpaired entities cannot be effectively discovered. KALE-Joint [20] proposes a new approach which learns entity and relation embeddings by jointly modelling knowledge triples and logic rules. However, all logic rules need to be grounded in these works. Since each relation type is linked to plenty of entities, each rule can be grounded into plenty of triples. The more original triples KG have, the more triples grounded from rules need to be used for learning, keeping grow exponentially. And thus they do not scale well to larger KGs. Also, the embeddings of the relation types in rules are not ordered, and thus the transitivity and asymmetry of logic rules are missed.

To address above issues, we propose a novel approach which learns embeddings by combining logic rules with knowledge triples. Logic rules are incorporated into relation type representations directly and the embeddings of relation types in logic rules are approximately ordered to leverage the transitivity and asymmetry of rules.

3 Problem Definition

In this section, we give a formal definition of the problem. A knowledge graph $\mathcal{G}$ is defined as a set of triples of the form (s, r, o). $s, o \in \mathcal{E}$ denote the subject and object entity, respectively. $r \in \mathcal{R}$ denotes the relation type. $\mathcal{E}$ denotes the set of all entities and $\mathcal{R}$ denotes the set of all relation types in $\mathcal{G}$.

Create from $\mathcal{G}$ a set of logic rules: $r_a \Rightarrow r_b$ (in form of (r_a, r_b)) denotes that r_a logically implies r_b, which means that any two entities linked by relation r_a should also be linked by relation r_b; $r_a \wedge r_b \Rightarrow r_c$ (in form of (r_a, r_b, r_c)) denotes that the conjunction of r_a and r_b logically implies r_c: if e_0 and e_1 are linked by r_a, e_1 and e_2 are linked by r_b, then e_0 and e_2 are linked by r_c. $r_a, r_b, r_c \in \mathcal{LR}$, where $\mathcal{LR} \subseteq \mathcal{R}$ is the subset of relation types observed in logic rules.

Our objective is to learn embeddings of entities, relations more precisely by approximately ordering the embeddings of relation types in logic rules, to predict relation types between entities. The embeddings are set in $\mathbb{R}^d$ and denoted with the same letters in boldface.

4 Our Model

4.1 Restricted Triple Model(RTM)

In RTM, we aim to embed entities and relation types to capture the correlations between them. The embeddings of relation types are restricted to be non-negative. Given two entities $s, o \in \mathcal{E}$, the log-odd of the probability of the truth of fact (s, r, o) is:

$$P(Y_{sro} = 1|\Theta) = \sigma(\phi(s, r, o)) \tag{1}$$

where $\sigma(x) = 1/(1 + exp(-x))$ denotes the logistic function that maps x to (0,1), which is just the range of probability; $\phi()$ is the energy function which is based on a factorization of the observed knowledges and indicates the correlation of relation r and the entity pair (s, o). $\Theta = \{\mathbf{e}_i\}_{i=1}^{n_e} \cup \{\mathbf{r}_k\}_{k=1}^{n_r}$ denotes the the set of all embeddings $\mathbf{v}_e, \mathbf{v}_r \in \mathbb{R}^d$ of the corresponding model, n_e and n_r is the number of entities and relation types in the given KG respectively. $\{Y_{sro}\}_{(s,r,o)\in\Omega} \in \{-1, 1\}^{|\Omega|}$ is a set of labels (true or false) of the triples, where $\Omega \in \mathcal{E} \otimes \mathcal{R} \otimes \mathcal{E}$. $Y_{sro} = 1$ if (s, r, o) is positive. Otherwise, $Y_{sro} = -1$.

The energy function $\phi(s, r, o; \Theta)$ in our model is based on existing model Complex [7], in which complex vectors $\mathbf{v}_e, \mathbf{v}_r \in \mathbb{R}^d$ are learned for each entity $e \in \mathcal{E}$ and each relation type $r \in \mathcal{R}$. It models the score of a triple as:

$$\begin{aligned}
\phi(s, r, o) &= Re(\langle \mathbf{r}\mathbf{s}\bar{\mathbf{o}} \rangle) \\
&= Re(\sum_{i=0}^{d-1} \mathbf{r}_i \mathbf{s}_i \bar{\mathbf{o}}_i) \\
&= \sum_{i=0}^{d-1} Re(\mathbf{r}_i)Re(\mathbf{s}_i)Re(\mathbf{o}_i) + \sum_{i=0}^{d-1} Re(\mathbf{r}_i)Im(\mathbf{s}_i)Im(\mathbf{o}_i) \\
&\quad + \sum_{i=0}^{d-1} Im(\mathbf{r}_i)Re(\mathbf{s}_i)Im(\mathbf{o}_i) - \sum_{i=0}^{d-1} Im(\mathbf{r}_i)Im(\mathbf{s}_i)Re(\mathbf{o}_i)
\end{aligned} \tag{2}$$

where **r**, **s**, **o** are complex vector embeddings for relation types, subject and object respectively. $Re(\mathbf{x})$ and $Im(\mathbf{x})$ represent the real part and imaginary part of the complex vector embedding **x**, respectively. It separates the embedding vector into real part and imaginary part to obtain the symmetric and antisymmetric relations. This function is antisymmetric when **r** is purely imaginary (i.e.its real part is zero), and symmetric when **r** is real.

To approximately order the relation types in logic rules, we constrain the real part and imaginary part of the vector embeddings of relation types to be non-negative and reduce the problem to Non-negative Matrix Factorization (**NMF**). And also, non-negative constraints make the embeddings interpretable. For most embedding methods, a critical issue is that, we are unaware of what each dimension represent in embeddings. Hence, the dimensions are difficult to interpret. This makes embeddings like a black-box, and prevents them from being human-readable and further manipulation. NMF learn embeddings with good interpretabilities. There are many ways to solve Non-negative Matrix Factorization such as Multiplicative Update [25], Gradient based Update [23] and Alternating Non-negative Least Squares [26, 27].

In our model, we adopt the approach which updates the embeddings based on gradient. Translate the general constrained optimization problem into an unconstrained optimization problem. The embeddings of relation types are translated into unconstrained complex vectors as follows:

$$Re(\mathbf{r}) = Re(\mathbf{q}_a)^{(2)}, \quad Im(\mathbf{r}) = Im(\mathbf{q}_b)^{(2)} \tag{3}$$

where $\mathbf{q}_a$ and $\mathbf{q}_b$ denote the vectors which are initialized randomly(not constrained to be non-negative) and updated during learning ,$\mathbf{x}^{(2)}$ denotes the element-wise square of the vector $\mathbf{x}$. In other words, $Re(\mathbf{r}_i) = Re(\mathbf{q}_{ai})^2$ and $Im(\mathbf{r}_i) = Im(\mathbf{q}_{bi})^2$. Plug eq.(3) into eq.(2) , we get a new energy function:

$$\begin{aligned}\phi(s,r,o) = &\sum_{i=0}^{d-1} Re(\mathbf{q}_{ai})^2 Re(\mathbf{s}_i) Re(\mathbf{o}_i) + \sum_{i=0}^{d-1} Re(\mathbf{q}_{ai})^2 Im(\mathbf{s}_i) Im(\mathbf{o}_i) \\ &+ \sum_{i=0}^{d-1} Im(\mathbf{q}_{bi})^2 Re(\mathbf{s}_i) Im(\mathbf{o}_i) - \sum_{i=0}^{d-1} Im(\mathbf{q}_{bi})^2 Im(\mathbf{s}_i) Re(\mathbf{o}_i)\end{aligned} \tag{4}$$

We train the triples by minimizing the negative log-likelihood of the logistic model with L^2 regularization on the parameters Θ:

$$\mathcal{L}_K = \min_{\Theta} \sum log(1 + exp(-Y_{sro}\phi(s,r,o))) + \lambda_1 \|\Theta\|^2 \tag{5}$$

where λ_1 is the regularization parameter.

The real part and imaginary part of complex vector $\mathbf{q}$ are both unconstrained. Therefore, the novel objective function can be solved by applying Stochastic Gradient Descent (SGD) directly. The negative set of knowledges is generated by local closed world assumption(LCWA) proposed in [28].

4.2 Approximate Order Logic Model(AOLM)

In AOLM, We aim to embed relation types to capture the ordering between relations. We work on two kinds of logic rules: $r_a \Rightarrow r_b$ and $r_a \wedge r_b \Rightarrow r_c$. Since there is no natural linear ordering on the set of complex numbers, we approximately order the complex vector embeddings by ordering the real part and imaginary part of the embeddings respectively. For their vector representations we require that the component-wise inequality holds:

$$\begin{aligned} r_a \Rightarrow r_b \; if \; and \; only \; if \; &\bigwedge_{i=0}^{d-1} Re(\mathbf{r}_{ai}) \leqslant Re(\mathbf{r}_{bi}) \\ and \; &\bigwedge_{i=0}^{d-1} Im(\mathbf{r}_{ai}) \leqslant Im(\mathbf{r}_{bi}) \end{aligned} \tag{6}$$

and

$$r_a \wedge r_b \Rightarrow r_c \; if \; and \; only \; if \; \bigwedge_{i=0}^{d-1} Re(\mathbf{r}_{ai}) Re(\mathbf{r}_{bi}) \leqslant Re(\mathbf{r}_{ci}) \\ and \bigwedge_{i=0}^{d-1} Im(\mathbf{r}_{ai}) Im(\mathbf{r}_{bi}) \leqslant Im(\mathbf{r}_{ci}) \tag{7}$$

for all vectors with non-negative coordinates, where $\bigwedge$ denotes the conjunction. Smaller coordinates imply higher position: $r_a \Rightarrow r_b$ if and only if all entries of the real part and imaginary part of the vector embedding of r_a are less than or equal to that of r_b.

The penalty for an ordered pair (r_a, r_b) of a given logic rule $r_a \Rightarrow r_b$ is defined as follows:

$$F(r_a, r_b) = \|max(\mathbf{0}, Re(\mathbf{r}_a) - Re(\mathbf{r}_b)) + max(\mathbf{0}, Im(\mathbf{r}_a) - Im(\mathbf{r}_b))\|^2 \tag{8}$$

where $max(\mathbf{0}, \mathbf{x})$ returns the greater one by element between $\mathbf{0}$ and $\mathbf{x}$.

Crucially, if $r_a \Rightarrow r_b$, $F(r_a, r_b)$ = 0. $F(r_a, r_b)$ is positive if $r_a \Rightarrow r_b$ is not satisfied. $F(r_a, r_b)$ = 0 if and only if $max(\mathbf{0}, Re(\mathbf{r}_a) - Re(\mathbf{r}_b)) = \mathbf{0}$ and $max(\mathbf{0}, Im(\mathbf{r}_a) - Im(\mathbf{r}_b)) = \mathbf{0}$. That is, the real part and imaginary part of the embeddings of r_a is less than or equal to that of r_b respectively. This satisfies the condition in eq.(6), and encourages the learned embeddings of relation types to satisfy the order properties of transitivity and asymmetry. For transitivity, if $r_a \Rightarrow r_b$ and $r_b \Rightarrow r_c$, $\bigwedge_{i=0}^{d-1} Re(\mathbf{r}_{ai}) \leqslant Re(\mathbf{r}_{bi}) \leqslant Re(\mathbf{r}_{ci}) \; and \; \bigwedge_{i=0}^{d-1} Im(\mathbf{r}_{ai}) \leqslant Im(\mathbf{r}_{bi}) \leqslant Im(\mathbf{r}_{ci})$, then $F(r_a, r_c) = 0$, and thus $r_a \Rightarrow r_c$ is satisfied. For asymmetry, if $r_a \Rightarrow r_b$, $\bigwedge_{i=0}^{d-1} Re(\mathbf{r}_{ai}) \leqslant Re(\mathbf{r}_{bi}) \; and \; \bigwedge_{i=0}^{d-1} Im(\mathbf{r}_{ai}) \leqslant Im(\mathbf{r}_{bi})$, then $F(r_a, r_b) >= 0$, and thus $r_b \Rightarrow r_a$ is not necessarily satisfied.

For logic rule $r_a \wedge r_b \Rightarrow r_c$ the penalty for (r_a, r_b, r_c) is:

$$F(r_a, r_b, r_c) = \|max(\mathbf{0}, Re(\mathbf{r}_a) * Re(\mathbf{r}_b) - Re(\mathbf{r}_c)) \\ + max(\mathbf{0}, Im(\mathbf{r}_a) * Im(\mathbf{r}_b) - Im(\mathbf{r}_c))\|^2 \tag{9}$$

where $*$ denotes the element-wise multiplication of two vectors: $[a * b]_i = a_i b_i$.

Similarly, if $r_a \wedge r_b \Rightarrow r_c$, $F(r_a, r_b, r_c)$ = 0. $F(r_a, r_b, r_c)$ is positive otherwise. $F(r_a, r_b, r_c)$ = 0 if and only if $max(\mathbf{0}, Re(\mathbf{r}_a) * Re(\mathbf{r}_b) - Re(\mathbf{r}_c)) = \mathbf{0}$ and $max(\mathbf{0}, Im(\mathbf{r}_a) * Im(\mathbf{r}_b) - Im(\mathbf{r}_c)) = \mathbf{0}$. That is, the multiplication of the real part and imaginary part of the embeddings of r_a and r_b is less than or equal to that of r_c respectively. This satisfies the condition in eq.(7).

The set of all relation types in logic rules is the subset of all relation types in the given KG. Therefore, the relation types in logic rules are translated similarly to eq.(3):

$$Re(\mathbf{r}_{a|b|c}) = Re(\mathbf{q}_{a|b|c})^{(2)}, Im(\mathbf{r}_{a|b|c}) = Im(\mathbf{q}_{a|b|c})^{(2)} \tag{10}$$

To learn the approximate order-embedding of relation types in logic rules, we could use a max-margin loss. For rule $r_a \Rightarrow r_b$:

$$\mathcal{L}_R = \min \sum_{(r_a, r_b) \in P} F(r_a, r_b) + \sum_{(r'_a, r'_b) \in N} \max(0, \alpha - F(r'_a, r'_b)) \tag{11}$$

If the rule is $r_a \wedge r_b \Rightarrow r_c$:

$$\mathcal{L}_R = \min \sum_{(r_a, r_b, r_c) \in P} F(r_a, r_b, r_c) + \sum_{(r'_a, r'_b, r'_c) \in N} \max(0, \alpha - F(r'_a, r'_b, r'_c)) \quad (12)$$

where P and N denote the positive and negative sets of logic rules. If $r_a \Rightarrow r_b$, we construct negatives by replacing r_b in the consequent with a random relation $r \in R$. If $r_a \wedge r_b \Rightarrow r_c$, we construct negatives by replacing r_c in the consequent with a random relation $r \in R$. $\alpha \geq 0$ is a hyper-parameter of margin. $F(r_a, r_b), F(r_a, r_b, r_c)$ is the penalty function score of positive logic rule, and $F(r'_a, r'_b), F(r'_a, r'_b, r'_c)$ is that of negative logic rule calculated by eq.(8) or eq.(9). This loss encourages positive examples to have zero penalty, and negative examples to have penalty greater than a margin.

4.3 Global Objective

With both knowledge triples and logic rules modelled, embeddings are learned by minimizing a global loss over this general representation:

$$\mathcal{L} = \mathcal{L}_K + \lambda_2 \mathcal{L}_R \quad (13)$$

where $\mathcal{L}_K$ is calculated by eq.(5) and $\mathcal{L}_R$ is calculated by eq.(11) or eq.(12) . The embeddings of relation types are constrained to be non-negative, and are translated into unconstrained complex vector embeddings in loss function. Therefore, stochastic gradient descent (SGD) in mini-batch mode and AdaGrad [13] for tuning the learning rate can be used to carry out the minimization directly. Embeddings learned are able to be compatible with both triples and logic rules. And the embeddings of relation types in logic rules are approximately ordered to capture the transitivity and asymmetry of rules.

Table 1. Complexity of our model and some other models

Model	Space	Time
TransE	$n_e d + n_r d$	$n_t d$
TransR	$n_e d + n_r (d^2 + d)$	$n_t d^2$
HOLE	$n_e d + n_r d$	$n_t log d$
Complex	$n_e d + n_r d$	$n_t d$
KALE-Joint	$n_e d + n_r d$	$n_t d + n_g d$
TARE	$n_e d + n_r d$	$n_t d + n_l d$

4.4 Discussions

Complexity. We compare our model with several state-of-the-art models in space and time complexity. Table 1 lists the complexity, where d is the dimension of the embedding vectors, n_e, n_r, n_t, n_g, n_l is the number of entities, relations, triples, ground triples, logical rules respectively. It can be seen that our model does not significantly increase the space or time complexity. Note that KALE-Joint [20] needs to ground rules with entities, which further requires O($n_g d$)=O($n_l n_t / n_r d$) in time complexity, where n_t/n_r is the averaged number of observed triples per relation. Our model only requires O($n_l d$) which is n_r/n_t of O($n_g d$) KALE-Joint required.

5 Experiments

5.1 Datasets and Experiment Settings

Datasets We evaluate our model on knowledge graph completion using two commonly used large-scale knowledge graph datasets and a relational learning dataset:

WN36 WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms called synsets. It provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. The WN18 dataset is a subset of WordNet which contains 40,943 entities, 18 relation types and 151,442 binary triples. Since there are no logic rules among all relation types in WN18, we first add the reversed relations into training set. For example, $*\_hypernym$ is the reversed relation type of $\_hypernym$. We add the triple $(e_1, *\_hypernym, e_0)$ into training set according to the positive triple observed $(e_0, \_hypernym, e_1)$. Then we can find some rules in newly generated training set. We create 14 implication rules.

FB15k Freebase is a large-scale and growing collaborative KG which provides general facts of the real world. For example, the triple (Barack Obama, Spouse, Michelle Obama) describes there is a relation Spouse between Barack Obama and Michelle Obama. The FB15k dataset is a subset of Freebase which contains 14,951 entities, 1,345 relation types, and 592,213 triples. We use original training, validation and test set splits as provided by [5]. We create 200 implication rules.

Countries The countries dataset provided by [14] consists of 244 countries, 22 subregions and 5 regions. Each country is located in exactly one region and subregion, each subregion is located in exactly one region, and each country can have a number of neighbour countries. We construct a set of triple relations from the raw data of two relations $LocatedIn$ and $NeighborOf$. We create 2 conjunction rules.

The statistics of WN36 and FB15k are listed in Table 2. Examples of rules created are shown in Table 3.

Table 2. Statistics of WN36 and FB15k

Model	entities	relations	train	valid	test	rules
Wn36	40,943	36	282,884	5,000	5,000	14
FB15k	14,951	1,345	483,142	50,000	59,071	200

Table 3. Examples of rules created

WN18	_hyponym ⇒ *_hypernym _member_meronym ⇒ *_member_holonym _part_of ⇒ *_has_part
FB15k	/award/award_honor/award_winner ⇒ /award/award_nomination/award_nominee /location/country/administrative_divisions ⇒ /location/location/contains /ice_hockey/hockey_roster_position/position ⇒ /sports/sports_team_roster/position
Countries	NeighborOf ∧ LocatedIn ⇒ LocatedIn LocatedIn ∧ LocatedIn ⇒ LocatedIn

Experiment Settings We use a grid search among the following parameters: $d \in \{20, 50, 100, 150, 200\}$, $n \in \{1, 2, 5, 10\}$, $a \in \{1.0, 0.5, 0.2,\ 0.1, 0.05, 0.02, 0.01\}$, $\lambda_1 \in \{0.1, 0.03, 0.01, 0.003, 0.001, 0.0, 0.0003\}$, $\lambda_2 \in \{0.1, 0.03, 0.01, 0.003, 0.001, 0.0, 0.0003\}$, $m \in \{2.0, 1.0, 0.5, 0.2, 0.05, 0.01\}$ to find the optimal parameters, where d denotes the embedding size of the vectors of the entity and relation type representations; n denotes the number of negatives sampled for per positive triple observed in training set or logic rule; a denotes the initial learning rate which will be tuned during AdaGrad; λ_1 denotes the L^2 regularization parameter, λ_2 is the weight of logic rules, and m is the margin between the positive logic rules and the negative logic rules.

5.2 Knowledge Base Completion

Knowledge base completion aims to complete a triple (s, r, o) when one of s, r, o is missing. In the task of knowledge base completion, we compare our model with several state-of-art models including TransE [5], TransR [11], HOLE [6], ComplEx [7] and KALE-Joint [20]. The former four models only focus on knowledge triples, and KALE-Joint learns embeddings by jointly modelling knowledge triples and logic rules. Rules need to be grounded and are not ordered in KALE-Joint.

We evaluate the performance of our model with Mean Reciprocal Rank (MRR) and top n (Hits@n) which have been widely used for evaluation in previous works. Replace the subject or object entity of each triple (s, r, o) in the testing set with each entity in the whole dataset: (s', r, o) and (s, r, o'), where $\forall s', \forall o' \in \mathcal{E}$. Afterwards, rank all candidate entities in the dataset according to their scores calculated by eq.(4) in ascending order. Mean Reciprocal Rank (MRR) and the ratio of correct entities ranked in top n (Hits@n) are the standard evaluation measures, which measure the quality of the ranking. They fall into two categories: raw and filtered. The filtered rankings are computed after filtering all other positive triples observed in the whole dataset, whereas the raw rankings do not filter these. We report both filtered and raw MRR, and filtered Hits@10, 3, 1 in Table 4 and Table 5 for the models.

It can be seen that TARE is able to outperform TransE, TransR, HOLE, ComplEx on MRR and Hits@ on WN36 and FB15k. This demonstrates the effectiveness of joint logic rules into knowledges. TARE largely outperforms KALE-Joint, with a filtered MRR of 0.955 and 91.4% of Hits@1, compared to 0.662 and 85.5% for KALE-Joint.

This demonstrates the effectiveness of considering the transitivity and asymmetry of logic rules.

Table 4. KG Completion on WN36

	MRR		Hits@		
Model	Filter	Raw	1	3	10
TransE	0.495	0.351	11.3	88.8	94.3
TransR	0.605	0.427	33.5	87.6	94.0
HOLE	0.938	0.616	93.0	**94.5**	94.9
Complex	0.941	**0.587**	**93.6**	**94.5**	94.7
KALE-Joint	0.662	0.478	85.5	90.1	93.0
TARE	**0.955**	0.545	91.4	94.2	**94.7**

Table 5. KG Completion on FB15k

	MRR		Hits@		
Model	Filter	Raw	1	3	10
TransE	0.463	0.222	29.7	57.8	74.9
TransR	0.346	0.198	21.8	40.4	58.2
HOLE	0.524	0.232	40.2	61.3	73.9
Complex	0.692	0.242	59.9	**75.9**	84.0
TARE	**0.781**	**0.292**	**61.7**	72.8	**84.2**

5.3 Relational Learning

We test the relational learning capabilities of our model on the countries dataset. Most of the test triples in the countries dataset can be inferred by directly applying logic rules on the training set. However, to evaluate our model, we do not use the pure logical inference. we split all countries randomly in train (80%), validation (10%), and test (10%) countries, then training, validation, and test set is composed of the relations which start from all countries in the training validation, and test countries respectively.

Remove all triples of the form $(c, LocatedIn, r)$ for each country c in the validation and test set. In the new set S_1, $(c, LocatedIn, r)$ can be predicted by $LocatedIn \wedge LocatedIn \Rightarrow LocatedIn$.

Based on S_1, remove $(c, LocatedIn, s)$ for all countries in the validation and test set. In the new set S_2, $(c, LocatedIn, r)$ can be predicted by $NeighborOf \wedge LocatedIn \Rightarrow LocatedIn$.

Based on S_2, remove $(c_n, LocatedIn, r)$ for all neighbour countries c_n of all countries in the validation and test set. In the new set S_3 , $(c, LocatedIn, r)$ can be predicted by $NeighborOf \wedge LocatedIn \Rightarrow LocatedIn$ and $LocatedIn \wedge LocatedIn \Rightarrow LocatedIn$.

The prediction quality is measured by the area under the precision-recall curve (AUC-PR), we compute the mean AUC-PR after 10 fold cross-validation. The results are shown in Table 6. It can be seen that our model performs well in this task. It achieves 13.4% improvement on S2 and 19.3% improvement on S3.

6 Conclusion and Future Work

In this paper, we propose TARE model for representation learning of knowledge graphs by integrating existing relations and logic rules together. Logic rules are incorporated

Table 6. Link prediction on Countries dataset

Model	S_1	S_2	S_3
Random	0. 323	0. 323	0. 323
Frequency	0. 323	0. 323	0. 308
ER-MLP	0. 960	0. 745	0. 650
Rescal	**0. 997**	0. 745	0. 650
HOLE	**0. 997**	0. 772	0. 697
TARE	0.994	**0.906**	**0.890**

into relation type representations directly, rather than instantiated with concrete entities. We model logic rules by approximately ordering the relation types in logic rules to leverage the transitivity and asymmetry of rules, and thus obtain better embeddings for entities and relation types. To be ordered, the vector embeddings of relation types are constrained to be non-negative, the general constrained optimization problem is translated into an unconstrained optimization problem in our model. In experiments, we evaluate our models on knowledge base completion and relational learning tasks. Experimental results show that TARE brings significant and consistent improvements over exiting state-of-the-art methods.

For future work, we would like to explore the following research directions: (1)more complex types of logic rules such as $\neg$ and $\vee$ would be modelled to obtain better performance. (2) logic rules can be extracted from text. There is richer information in text than triples, more logic rules can be obtained if we joint the information in text. (3) TARE only consider the order of relation types, the order over entities would also be helpful to obtain better embeddings.

Acknowledgements. We thank the National Key Research and Development Program of China (2016YFB0201900), National Natural Science Foundation of China (U1611262), Guangdong Natural Science Funds for Distinguished Young Scholar (2017A030306028), Pearl River Science and Technology New Star of Guangzhou, and Guangdong Province Key Laboratory of Big Data Analysis and Processing for the support of this research.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. The Semantic Web. ISWC 2007 + Aswc 2007. Busan. Korea. November. DBLP. 722-735(2007)
2. Pritsker, E. W., Cohen, W., Minkov, E.: Learning to Identify the Best Contexts for Knowledge-based WSD. EMNLP, 1662-1667 (2015)
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase:a collaboratively created graph database for structuring human knowledge. SIGMOD Conference, 1247-1250 (2008)
4. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D. S.: Knowledge-based weak supervision for information extraction of overlapping relations. ACL 541-550 (2011)
5. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. NIPS. Curran Associates Inc. 2787-2795 (2013)

6. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. 1955-1961 (2015)
7. Welbl, J., Riedel, S., Bouchard, G.: Complex embeddings for simple link prediction. ICML. JMLR.org. 2071-2080 (2016)
8. Riedel, S., Yao, L., Mccallum, A., Marlin, B. M.: Relation extraction with matrix factorization and universal schemas (2013)
9. Chang, K. W., Yih, W. T., Yang, B., Meek, C.: Typed Tensor Decomposition of Knowledge Bases for Relation Extraction. EMNLP (2014)
10. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. AAAI. 1112-1119 (2014)
11. Lin, Y., Liu, Z., Zhu, X., Zhu, X., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. AAAI. 2181-2187 (2015)
12. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. Computer Science (2015)
13. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. JMLR. 257-269 (2011)
14. Bouchard, G., Singh, S., Trouillon, T.: On approximate reasoning capabilities of low-rank vector spaces. (2015)
15. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. ACM SIGKDD. 601-610 (2014)
16. Neelakantan, A., Roth, B., Mccallum, A.: Compositional vector space models for knowledge base completion. Computer Science. 1-16 (2015)
17. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge Graph Embedding via Dynamic Mapping Matrix. ACL, IJCNLP. 687-696 (2015)
18. Wang, Q., Wang, B., Guo, L.: Knowledge base completion using embeddings and rules. International Conference on Artificial Intelligence. AAAI Press. 1859-1865 (2015)
19. Rocktschel, T., Singh, S., Riedel, S.: Injecting Logical Background Knowledge into Embeddings for Relation Extraction. NAACL. 648-664 (2014)
20. Shu, G., Quan, W., Wang, L., Wang, B., Li, G.: Jointly Embedding Knowledge Graphs and Logical Rules. EMNLP. 192-202 (2016)
21. Vendrov, I., Kiros, R., Fidler, S., Urtasun, R.: Order-embeddings of images and language. (2015)
22. Socher, R., Chen, D., Manning, C. D., Ng, A. Y.: Reasoning with neural tensor networks for knowledge base completion. NIPS. Curran Associates Inc. 926-934 (2013)
23. Lin, C. J.: Projected gradient methods for nonnegative matrix factorization. Neural Computation, 19(10), 2756 (2007)
24. Guan, N., Tao, D., Luo, Z., Yuan, B.: Nenmf: an optimal gradient method for nonnegative matrix factorization. TSP, 60(6), 2882-2898 (2012)
25. Lee, D. D., Seung, H. S.: Algorithms for non-negative matrix factorization. NIPS. MIT Press. 535-541 (2000)
26. Kim, H., Park, H.: Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics, 23(12), 1495. (2007)
27. Cichocki, A., Zdunek, R.: Regularized Alternating Least Squares Algorithms for Non-negative Matrix/Tensor Factorization. Isnn, Nanjing, China. DBLP. 793-802 (2007)
28. Krompa, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. 107(8), 640-655 (2015)
29. Nickel, M., Tresp, V., Kriegel, H. P.: A three-way model for collective learning on multi-relational data. ICML. 809-816 Omnipress (2011)
30. Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. Computer Science (2015)
31. Garca-Durn, A., Bordes, A., Usunier, N.: Composing relationships with translations (2015)