Regularizing Knowledge Graph Embeddings via Equivalence and Inversion Axioms

Pasquale Minervini[1(✉)], Luca Costabello[2], Emir Muñoz[1,2], Vít Nováček[1], and Pierre-Yves Vandenbussche[2]

[1] Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
{pasquale.minervini,emir.munoz,vit.novacek}@insight-centre.org
[2] Fujitsu Ireland Ltd., Galway, Ireland
{luca.costabello,emir.munoz,pierre-yves.vandenbussche}@ie.fujitsu.com

Abstract. Learning embeddings of entities and relations using neural architectures is an effective method of performing statistical learning on large-scale relational data, such as knowledge graphs. In this paper, we consider the problem of regularizing the training of neural knowledge graph embeddings by leveraging external background knowledge. We propose a principled and scalable method for leveraging equivalence and inversion axioms during the learning process, by imposing a set of model-dependent soft constraints on the predicate embeddings. The method has several advantages: *(i)* the number of introduced constraints does not depend on the number of entities in the knowledge base; *(ii)* regularities in the embedding space effectively reflect available background knowledge; *(iii)* it yields more accurate results in link prediction tasks over non-regularized methods; and *(iv)* it can be adapted to a variety of models, without affecting their scalability properties. We demonstrate the effectiveness of the proposed method on several large knowledge graphs. Our evaluation shows that it consistently improves the predictive accuracy of several neural knowledge graph embedding models (for instance, the MRR of TransE on WordNet increases by 11%) without compromising their scalability properties.

1 Introduction

Knowledge graphs are graph-structured knowledge bases, where factual knowledge is represented in the form of relationships between entities: they are powerful instruments in search, analytics, recommendations, and data integration. This justified a broad line of research both from academia and industry, resulting in projects such as DBpedia (Auer et al. 2007), Freebase (Bollacker et al. 2007), YAGO (Suchanek et al. 2012), NELL (Carlson et al. 2010), and Google's Knowledge Graph and Knowledge Vault projects (Dong et al. 2014).

However, despite their size, knowledge graphs are often very far from being complete. For instance, 71% of the people described in Freebase have no known place of birth, 75% have no known nationality, and the coverage for less used relations can be even lower (Dong et al. 2014). Similarly, in DBpedia, 66% of

M. Ceci et al. (Eds.): ECML PKDD 2017, Part I, LNAI 10534, pp. 668–683, 2017.
https://doi.org/10.1007/978-3-319-71249-9_40

the persons are also missing a place of birth, while 58% of the scientists are missing a fact stating what they are known for (Krompaß et al. 2015).

In this work, we focus on the problem of *predicting missing links* in large knowledge graphs, so to discover new facts about the world. In the literature, this problem is referred to as *link prediction* or *knowledge base population*: we refer to Nickel et al. (2016) for a recent survey on machine learning-driven solutions to this problem.

Recently, *neural knowledge graph embedding models* (Nickel et al. 2016) – neural architectures for embedding entities and relations in continuous vector spaces – have received a growing interest: they achieve state-of-the-art link prediction results, while being able to scale to very large and highly-relational knowledge graphs. Furthermore, they can be used in a wide range of applications, including entity disambiguation and resolution (Bordes et al. 2014), taxonomy extraction (Nickel et al. 2016), and query answering on probabilistic databases (Krompaß et al. 2014). However, a limitation in such models is that they only rely on existing facts, without making use of any form of background knowledge. At the time of this writing, how to efficiently leverage preexisting knowledge for learning more accurate neural knowledge graph embeddings is still an open problem (Wang et al. 2015).

Contribution – In this work, we propose a principled and scalable method for leveraging external background knowledge for regularising neural knowledge graph embeddings. In particular, we leverage background axioms in the form $p \equiv q$ and $p \equiv q^-$, where the former denotes that relations p and q are *equivalent*, such as in the case of relations PARTOF and COMPONENTOF, while the latter denotes that the relation p is the *inverse* of the relation q, such as in the case of relations PARTOF and HASPART. Such axioms are used for defining and imposing a set of model-dependent soft constraints on the relation embeddings during the learning process. Such constraints can be considered as regularizers, reflecting available prior knowledge on the distribution of embedding representations of relations.

The proposed method has several advantages: *(i)* the number of introduced constraints is independent on the number of entities, allowing it to scale to large and Web-scale knowledge graphs with millions of entities; *(ii)* relationships between relation types in the embedding space effectively reflect available background schema knowledge; *(iii)* it yields more accurate results in link prediction tasks than state-of-the-art methods; and *(iv)* it is a general framework, applicable to a variety of embedding models. We demonstrate the effectiveness of the proposed method in several link prediction tasks: we show that it consistently improves the predictive accuracy of the models it is applied to, without negative impact on their scalability properties.

2 Preliminaries

Knowledge Graphs – A knowledge graph is a graph-structured knowledge base, where factual information is stored in the form of relationships

between entities. Formally, a knowledge graph $\mathcal{G} \triangleq \{\langle s, p, o\rangle\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of $\langle s, p, o\rangle$ triples, each consisting of a *subject* s, a *predicate* p and an *object* o, and encoding the statement "s has a relationship p with o". The subject and object $s, o \in \mathcal{E}$ are entities, $p \in \mathcal{R}$ is a relation type, and $\mathcal{E}, \mathcal{R}$ respectively denote the sets of all entities and relation types in the knowledge graph.

Example 1. Consider the following statement: *"Ireland is located in Northern Europe, and shares a border with the United Kingdom."* It can be expressed by the following triples:

Subject	Predicate	Object
Ireland	locatedIn	Northern Europe
Ireland	neighborOf	United Kingdom

A knowledge graph can be represented as a labelled directed multigraph, in which each triple is represented as an edge connecting two nodes: the source and target nodes represent the subject and object of the triple, and the edge label represents the predicate.

Knowledge graph adhere to the *Open World Assumption* (Hayes and Patel-Schneider 2014): a missing triple does not necessarily imply that the corresponding statement holds false, but rather that its truth value is *unknown*, *i.e.* it cannot be observed in the graph. For instance, the fact that the triple ⟨United Kingdom, neighborOf, Ireland⟩ is missing from the graph in Example 1 does not imply that the United Kingdom does not share a border with Ireland, but rather that we do not know whether this statement is true or not.

Equivalence and Inversion Axioms – Knowledge graphs are usually endowed with additional background knowledge, describing classes of entities and their properties and characteristics, such as equivalence and symmetry. In this work, we focus on two types of logical axioms in the form $p \equiv q$ and $p \equiv q^-$, where $p, q \in \mathcal{R}$ are predicates.

A largely popular knowledge representation formalism for expressing schema axioms is the OWL 2 Web Ontology language (Schneider 2012). According to the OWL 2 RDF-based semantics, the axiom $p \equiv q$ implies that predicates p and q share the same property extension, *i.e.* if $\langle s, p, o\rangle$ is true then $\langle s, q, o\rangle$ is also true (and vice-versa). Similarly, the axiom $p \equiv q^-$ implies that the predicate q is the inverse of the predicate p, *i.e.* if $\langle s, p, o\rangle$ is true then $\langle o, q, s\rangle$ is also true (and vice-versa). It is possible to express that a predicate $p \in \mathcal{R}$ is *symmetric* by using the axiom $p \equiv p^-$. Such axioms can be expressed by the OWL 2 `owl:equivalentProperty` and `owl:inverseOf` constructs.

Example 2. Consider the following statement: *"The relation* locatedIn *is the inverse of the relation* locationOf*, and the relation* neighborOf *is symmetric."* It can be encoded by the axioms locatedIn $\equiv$ locationOf$^-$ and neighborOf $\equiv$ neighborOf$^-$.

Link Prediction – As mentioned earlier, real world knowledge graphs are often largely incomplete. *Link prediction* in knowledge graphs consists in identifying missing triples (facts) in order to discover new facts about a domain of interest. This task is also referred to as *knowledge base population* in literature. We refer to Nickel et al. (2016) for a recent survey on link prediction methods.

The link prediction task can be cast as a *learning to rank* problem, where we associate a *prediction score* ϕ_{spo} to each triple $\langle s, p, o \rangle$ as follows:

$$\phi_{spo} \triangleq \phi(\langle s, p, o \rangle; \Theta),$$

where the score ϕ_{spo} represents the confidence of the model that the statement encoded by the triple $\langle s, p, o \rangle$ holds true, $\phi(\cdot; \Theta)$ denotes a *triple scoring function*, with $\phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$, and Θ represents the parameters of the scoring function and thus of the link prediction model. Triples associated with a higher score by the link prediction model have a higher probability of encoding a true statement, and are thus considered for a completion of the knowledge graph $\mathcal{G}$.

3 Neural Knowledge Graph Embedding Models

Recently, *neural* link prediction models received a growing interest (Nickel et al. 2016). They can be interpreted as simple multi-layer neural networks, where given a triple $\langle s, p, o \rangle$, its score $\phi(\langle s, p, o \rangle; \Theta)$ is given by a two-layer neural network architecture, composed by an *encoding layer* and a *scoring layer*.

Encoding Layer – in the encoding layer, the subject and object entities s and o are mapped to distributed vector representations $\mathbf{e}_s$ and $\mathbf{e}_o$, referred to as *embeddings*, by an encoder $\psi : \mathcal{E} \mapsto \mathbb{R}^k$ such that $\mathbf{e}_s \triangleq \psi(s)$ and $\mathbf{e}_o \triangleq \psi(o)$. Given an entity $s \in \mathcal{E}$, the encoder ψ is usually implemented as a simple embedding layer $\psi(s) \triangleq [\boldsymbol{\Psi}]_s \in \mathbb{R}^k$, where $\boldsymbol{\Psi} \in \mathbb{R}^{|\mathcal{E}| \times k}$ is an embedding matrix (Nickel et al. 2016).
The distributed representations in this layer can be either pre-trained (Baroni et al. 2012) or, more commonly, learnt from data by back-propagating the link prediction error to the embeddings (Bordes et al. 2013; Yang et al. 2015; Trouillon et al. 2016; Nickel et al. 2016).
Scoring Layer – in the scoring layer, the subject and object representations $\mathbf{e}_s$ and $\mathbf{e}_o$ are scored by a predicate-dependent function $\phi_p^\theta(\mathbf{e}_s, \mathbf{e}_o) \in \mathbb{R}$, parametrised by θ.

The architecture of neural link prediction models can be summarized as follows:

$$\begin{aligned} \phi(\langle s, p, o \rangle; \Theta) &\triangleq \phi_p^\theta(\mathbf{e}_s, \mathbf{e}_o) \\ \mathbf{e}_s, \mathbf{e}_o &\triangleq \psi(s), \psi(o), \end{aligned} \tag{1}$$

and the set of parameters Θ corresponds to $\Theta \triangleq \{\theta, \boldsymbol{\Psi}\}$. Neural link prediction model generate distributed embedding representations for all entities in a knowledge graph, as well as a model of determining whether a triple is more likely than

others, by means of a neural network architecture. For such a reason, they are also referred to as *neural knowledge graph embedding models* (Yang et al. 2015; Nickel et al. 2016).

Several neural link prediction models have been proposed in the literature. For brevity, we overview a small subset of these, namely the Translating Embeddings model TransE (Bordes et al. 2013); the Bilinear-Diagonal model DistMult (Yang et al. 2015); and its extension in the complex domain ComplEx (Trouillon et al. 2016). Unlike previous models, such models can scale to very large knowledge graphs, thanks to: (i) a space complexity that grows *linearly* with the number of entities $|\mathcal{E}|$ and relations $|\mathcal{R}|$; and (ii) efficient and scalable scoring functions and parameters learning procedures. In the following, we provide a brief and self-contained overview of such neural knowledge graph embedding models.

TransE – The scoring layer in TransE is defined as follows:

$$\phi_p(\mathbf{e}_s, \mathbf{e}_o) \triangleq -\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| \in \mathbb{R},$$

where $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$ represent the subject and object embeddings, $\mathbf{r}_p \in \mathbb{R}^k$ is a predicate-dependent translation vector, $\|\cdot\|$ denotes either the L_1 or the L_2 norm, and $\|\mathbf{x} - \mathbf{y}\|$ denotes the distance between vectors $\mathbf{x}$ and $\mathbf{y}$. In TransE, the score $\phi_p(\mathbf{e}_s, \mathbf{e}_o)$ is then given by the *similarity* between the translated subject embedding $\mathbf{e}_s + \mathbf{r}_p$ and the object embedding $\mathbf{e}_o$.

DistMult – The scoring layer in DistMult is defined as follows:

$$\phi_p(\mathbf{e}_s, \mathbf{e}_o) \triangleq \langle \mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o \rangle \in \mathbb{R},$$

where, given $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^k$, $\langle \mathbf{x}, \mathbf{y}, \mathbf{z} \rangle \triangleq \sum_{i=1}^{k} \mathbf{x}_i \mathbf{y}_i \mathbf{z}_i$ denotes the standard component-wise multi-linear dot product, and $\mathbf{r}_p \in \mathbb{R}^k$ is a predicate-dependent vector.

ComplEx – The recently proposed ComplEx is related to DistMult, but uses complex-valued embeddings while retaining the mathematical definition of the dot product. The scoring layer in ComplEx is defined as follows:

$$\begin{aligned}\phi_p(\mathbf{e}_s, \mathbf{e}_o) &\triangleq \mathrm{Re}\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right) \\ &= \langle \mathrm{Re}\left(\mathbf{r}_p\right), \mathrm{Re}\left(\mathbf{e}_s\right), \mathrm{Re}\left(\mathbf{e}_o\right) \rangle + \langle \mathrm{Re}\left(\mathbf{r}_p\right), \mathrm{Im}\left(\mathbf{e}_s\right), \mathrm{Im}\left(\mathbf{e}_o\right) \rangle \\ &\quad + \langle \mathrm{Im}\left(\mathbf{r}_p\right), \mathrm{Re}\left(\mathbf{e}_s\right), \mathrm{Im}\left(\mathbf{e}_o\right) \rangle - \langle \mathrm{Im}\left(\mathbf{r}_p\right), \mathrm{Im}\left(\mathbf{e}_s\right), \mathrm{Re}\left(\mathbf{e}_o\right) \rangle \in \mathbb{R},\end{aligned}$$

where given $\mathbf{x} \in \mathbb{C}^k$, $\overline{\mathbf{x}}$ denotes the complex conjugate of $\mathbf{x}$[1], while $\mathrm{Re}\left(\mathbf{x}\right) \in \mathbb{R}^k$ and $\mathrm{Im}\left(\mathbf{x}\right) \in \mathbb{R}^k$ denote the real part and the imaginary part of $\mathbf{x}$, respectively.

4 Training Neural Knowledge Graph Embedding Models

In neural knowledge graph embedding models, the parameters Θ of the embedding and scoring layers are learnt from data. A widely popular strategy for

[1] Given $x \in \mathbb{C}$, its complex conjugate is $\overline{x} \triangleq \mathrm{Re}\left(x\right) - i\mathrm{Im}\left(x\right)$.

Algorithm 1. Learning the model parameters Θ via Projected SGD

Require: Batch size n, epochs τ, learning rates $\boldsymbol{\eta} \in \mathbb{R}^{\tau}$
Ensure: Optimal model parameters $\hat{\Theta}$
for $i = 1, \ldots, \tau$ **do**
  $\mathbf{e}_e \leftarrow \mathbf{e}_e / \|\mathbf{e}_e\|, \ \forall e \in \mathcal{E}$
  {Sample a batch of positive and negative examples $\mathcal{B} = \{(t, \tilde{t})\}$}
  $\mathcal{B} \leftarrow \textsc{SampleBatch}(\mathcal{G}, n)$
  {Compute the gradient of the loss function $\mathcal{J}$ on examples $\mathcal{B}$}
  $g_i \leftarrow \nabla \sum_{(t,\tilde{t}) \in \mathcal{B}} \left[\gamma - \phi(t; \Theta_{i-1}) + \phi(\tilde{t}; \Theta_{i-1})\right]_+$
  {Update the model parameters via gradient descent}
  $\Theta_i \leftarrow \Theta_{i-1} - \eta_i g_i$
end for
return Θ_τ

learning the model parameters is described in Bordes et al. (2013); Yang et al. (2015); Nickel et al. (2016). In such works, authors estimate the optimal parameters by minimizing the following pairwise margin-based ranking loss function $\mathcal{J}$ defined on parameters Θ:

$$\mathcal{J}(\Theta) \triangleq \sum_{t^+ \in \mathcal{G}} \sum_{t^- \in \mathcal{C}(t^+)} \left[\gamma - \phi(t^+; \Theta) + \phi(t^-; \Theta)\right]_+ \tag{2}$$

where $[x]_+ = \max\{0, x\}$, and $\gamma \geq 0$ specifies the width of the margin. Positive examples t^+ are composed by all triples in $\mathcal{G}$, and negative examples t^- are generated by using the following *corruption process*:

$$\mathcal{C}(\langle s, p, o \rangle) \triangleq \{\langle \tilde{s}, p, o \rangle \mid \tilde{s} \in \mathcal{E}\} \cup \{\langle s, p, \tilde{o} \rangle \mid \tilde{o} \in \mathcal{E}\},$$

which, given a triple, generates a set of corrupt triples by replacing its subject and object with all other entities in $\mathcal{G}$. This method of sampling negative examples is motivated by the *Local Closed World Assumption* (LCWA) (Dong et al. 2014). According to the LCWA, if a triple $\langle s, p, o \rangle$ exists in the graph, other triples obtained by corrupting either the subject or the object of the triples not appearing in the graph can be considered as negative examples. The optimal parameters can be learnt by solving the following minimization problem:

$$\begin{aligned} \underset{\Theta}{\text{minimize}} \quad & \mathcal{J}(\Theta) \\ \text{subject to} \quad & \forall e \in \mathcal{E} : \ \|\mathbf{e}_e\| = 1, \end{aligned} \tag{3}$$

where Θ denotes the parameters of the model. The norm constraints on the entity embeddings prevent to trivially solve the optimization problem by increasing the norm of the embedding vectors (Bordes et al. 2014). The loss function in Eq. (2) will reach its global minimum 0 iff, for each pair of positive and negative examples t^+ and t^-, the score of the (true) triple t^+ is higher with a margin of at least γ than the score of the (missing) triple t^-. Following Yang et al. (2015), we use the Projected Stochastic Gradient Descent (SGD) algorithm (outlined in

Algorithm 1) for solving the loss minimization problem in Eq. (3), and AdaGrad (Duchi et al. 2011) for automatically selecting the optimal learning rate η at each iteration.

5 Regularizing via Background Knowledge

We now propose a method for incorporating background schema knowledge, provided in the form of equivalence and inversion axioms between predicates, in neural knowledge graph embedding models. Formally, let $\mathcal{A}_1$ and $\mathcal{A}_2$ denote the following two sets of equivalence and inversion axioms between predicates:

$$\mathcal{A}_1 \triangleq \{p_1 \equiv q_1, \ldots, p_m \equiv q_m\} \qquad \mathcal{A}_2 \triangleq \{p_{m+1} \equiv q_{m+1}^-, \ldots, p_n \equiv q_n^-\} \tag{4}$$

where $1 \leq m \leq n$, and $\forall i \in \{1, \ldots, n\} : p_i, q_i \in \mathcal{R}$. Recall that each axiom $p \equiv q$ encodes prior knowledge that predicates p and q are equivalent, *i.e.* they share the same extension. Similarly, each axiom $p \equiv q^-$ encodes prior knowledge that the predicate p and the *inverse* of the predicate q are equivalent.

Equivalence Axioms – Consider the case in which predicates $p \in \mathcal{R}$ and $q \in \mathcal{R}$ are equivalent, as encoded by the axiom $p \equiv q$. This implies that a model with scoring function $\phi(\cdot\,; \Theta)$ and parameters Θ should assign the same scores to the triples $\langle s, p, o \rangle$ and $\langle s, q, o \rangle$, for all entities $s, o \in \mathcal{E}$:

$$\phi(\langle s, p, o \rangle; \Theta) = \phi(\langle s, q, o \rangle; \Theta) \quad \forall s, o \in \mathcal{E}. \tag{5}$$

A simple method for enforcing the constraint in Eq. (5) during the parameter learning process consists in solving the loss minimization problem in Eq. (3) under the additional equality constraints in Eq. (5). However, this solution results in introducing $\mathcal{O}(|\mathcal{E}|^2)$ constraints in the optimization problem in Eq. (3), a quantity that grows *quadratically* with the number of entities $|\mathcal{E}|$. This solution may not be feasible for very large knowledge graphs, which typically contain millions of entities or more, while $|\mathcal{R}|$ is usually several orders of magnitude lower. A more efficient method consists in enforcing the model to associate *similar embedding representations* to both p and q, *i.e.* $\mathbf{r}_p = \mathbf{r}_q$. This solution can be encoded by *a single constraint*, satisfying all identities in Eq. (5).

Inversion Axioms – Consider the case in which the predicate p (*e.g.* PARTOF) and the inverse of the predicate q (*e.g.* HASPART) are equivalent, as encoded by the axiom $p \equiv q^-$. This implies that a model with scoring function $\phi(\cdot\,; \Theta)$ and parameters Θ should assign the same scores to the triples $\langle s, p, o \rangle$ and $\langle o, q, s \rangle$, for all entities $s, o \in \mathcal{E}$:

$$\phi(\langle s, p, o \rangle; \Theta) = \phi(\langle o, q, s \rangle; \Theta) \quad \forall s, o \in \mathcal{E}. \tag{6}$$

Also in this case we can enforce the identity in Eq. (6) through a single constraint on the embeddings of predicates p and q. In the following, we derive the constraints for the models TRANSE, DISTMULT and COMPLEX. The constraints

rely on a function $\Phi(\cdot)$ that applies a model-dependent transformation to the predicate embedding $\mathbf{r}_q$.

TransE: We want to enforce that, for any pair of s and o embedding vectors $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$, the score associated to the triples $\langle s, p, o\rangle$ and $\langle o, q, s\rangle$ are the same. Formally:

$$\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| = \|\mathbf{e}_o + \mathbf{r}_q - \mathbf{e}_s\|, \quad \forall \mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k \tag{7}$$

where $\|\cdot\|$ denotes either the L_1 or the L_2 norm.

Theorem 1. *The identity in Eq. (7) is satisfied by imposing:*

$$\mathbf{r}_p = \Phi(\mathbf{r}_q) \quad \textit{such that} \quad \Phi(\mathbf{r}_q) \triangleq -\mathbf{r}_q.$$

Proof. *For any $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$, the following result holds:*

$$\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| = \|\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s\|,$$

where $\|\cdot\|$ is a norm on $\mathbb{R}^k$. Because of the absolute homogeneity property *of norms we have that, for any $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^k$:*

$$\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|.$$

It follows that:

$$\begin{aligned}\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| &= \|-1\,(\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s)\| \\ &= |-1|\|\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s\| \qquad \textit{(absolute homogeneity property)} \\ &= \|\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s\|.\end{aligned}$$

DistMult: We want to enforce that:

$$\langle \mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o\rangle = \langle \mathbf{r}_q, \mathbf{e}_o, \mathbf{e}_s\rangle, \quad \forall \mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k \tag{8}$$

A limitation in DistMult, addressed by ComplEx, is that its scoring function is *symmetric*, *i.e.* it assigns the same score to $\langle s, p, o\rangle$ and $\langle o, p, s\rangle$, due to the commutativity of the element-wise product.

The identity in Eq. (8) is thus satisfied by imposing $\mathbf{r}_p = \Phi(\mathbf{r}_q)$ such that $\Phi(\mathbf{r}_q) \triangleq \mathbf{r}_q$.

ComplEx: We want to enforce that:

$$\mathrm{Re}\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o}\rangle\right) = \mathrm{Re}\left(\langle \mathbf{r}_q, \mathbf{e}_o, \overline{\mathbf{e}_s}\rangle\right), \quad \forall \mathbf{e}_s, \mathbf{e}_o \in \mathbb{C}^k. \tag{9}$$

The identity in Eq. (9) can be satisfied as follows:

Theorem 2. *The identity in Eq. (9) is satisfied by imposing:*

$$\mathbf{r}_p = \Phi(\mathbf{r}_q) \quad \textit{such that} \quad \Phi(\mathbf{r}_q) \triangleq \overline{\mathbf{r}_q}.$$

Proof. *For any* $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{C}^k$*, the following result holds:*

$$Re\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right)) = Re\left(\langle \overline{\mathbf{r}_p}, \mathbf{e}_o, \overline{\mathbf{e}_s} \rangle\right)).$$

Consider the following steps:

$$\begin{aligned} Re\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right) &= Re\left(\overline{\langle \overline{\mathbf{r}_p}, \overline{\mathbf{e}_s}, \mathbf{e}_o \rangle}\right) && \textit{(since} \overline{(\overline{\mathbf{x}})} = \mathbf{x}\textit{)} \\ &= Re\left(\overline{\langle \overline{\mathbf{r}_p}, \mathbf{e}_o, \overline{\mathbf{e}_s} \rangle}\right) && \textit{(commutative property)} \\ &= Re\left(\langle \overline{\mathbf{r}_p}, \mathbf{e}_o, \overline{\mathbf{e}_s} \rangle\right) && \textit{(since} Re\left(\overline{\mathbf{x}}\right) = Re\left(\mathbf{x}\right)\textit{).} \end{aligned}$$

Similar procedures for deriving the function $\Phi(\cdot)$ can be used in the context of other knowledge graph embedding models.

5.1 Regularizing via Soft Constraints

One solution for integrating background schema knowledge consists in solving the loss minimization problem in Eq. (3) under additional hard equality constraints on the predicate embeddings, for instance by enforcing $\mathbf{r}_p = \mathbf{r}_q$ for all $p \equiv q \in \mathcal{A}_1$, and $\mathbf{r}_p = \Phi(\mathbf{r}_q)$ for all $p \equiv q^- \in \mathcal{A}_2$. However, this solution does not cover cases in which two predicates are not strictly equivalent but still share very similar semantics, such as in the case of predicates MARRIEDWITH and PARTNEROF.

A more flexible solution consists in relying on *soft constraints* (Meseguer et al. 2006), which are used to formalize *desired properties* of the model rather than requirements that cannot be violated: we propose relying on weighted soft constraints for encoding our background knowledge on latent predicate representations.

Formally, we extend the loss function $\mathcal{J}$ described in Eq. (2) with an additional penalty term $\mathcal{R}_\mathcal{S}$ for enforcing a set of desired relationships between the predicate embeddings. This process leads to the following novel loss function $\mathcal{J}_\mathcal{S}$:

$$\begin{aligned} \mathcal{R}_\mathcal{S}(\Theta) &\triangleq \sum_{p \equiv q \in \mathcal{A}_1} D\left[\mathbf{r}_p \| \mathbf{r}_q\right] + \sum_{p \equiv q^- \in \mathcal{A}_2} D\left[\mathbf{r}_p \| \Phi(\mathbf{r}_q)\right] \\ \mathcal{J}_\mathcal{S}(\Theta) &\triangleq \mathcal{J}(\Theta) + \lambda \mathcal{R}_\mathcal{S}(\Theta), \end{aligned} \tag{10}$$

where $\lambda \geq 0$ is the weight associated with the soft constraints, and $D\left[\mathbf{x} \| \mathbf{y}\right]$ is a divergence measure between two vectors $\mathbf{x}$ and $\mathbf{y}$. In our experiments, we use the Euclidean distance as divergence measure, *i.e.* $D\left[\mathbf{x} \| \mathbf{y}\right] \triangleq \|\mathbf{x} - \mathbf{y}\|_2^2$.

In particular, $\mathcal{R}_\mathcal{S}$ in Eq. (10) can be thought of as a schema-aware *regularization term*, which encodes our prior knowledge on the distribution of predicate embeddings. Note that the formulation in Eq. (10) allows us to freely interpolate between *hard constraints* ($\lambda = \infty$) and the original models represented by the loss function $\mathcal{J}$ ($\lambda = 0$), permitting to adaptively specify the relevance of each logical axiom in the embedding model.

6 Related Works

How to effectively improve neural knowledge graph embeddings by making use of background knowledge is a largely unexplored field. Chang et al. (2014); Krompass et al. (2014); Krompaß et al. (2015) make use of type information about entities for only considering interactions between entities belonging to the domain and range of each predicate, assuming that type information about entities is complete. In Minervini et al. (2016), authors assume that type information can be incomplete, and propose to adaptively decrease the score of each missing triple depending on the available type information. These works focus on type information about entities, while we propose a method for leveraging background knowledge about relation types which can be used jointly with the aforementioned methods.

Dong et al. (2014); Nickel et al. (2014); Wang et al. (2015) propose combining observable patterns in the form of rules and latent features for link prediction tasks. However, rules are not used *during* the parameters learning process, but rather *after*, in an ensemble fashion. Wang et al. (2015) suggest investigating how to incorporate logical schema knowledge during the parameters learning process as a future research direction. Rocktäschel et al. (2015) regularize relation and entity representations by grounding first-order logic rules. However, as they state in their paper, adding a very large number of ground constraints does not scale to domains with a large number of entities and predicates.

In this work we focus on *2-way* models rather than *3-way* models (García-Durán et al. 2014), since the former received an increasing attention during the last years, mainly thanks to their scalability properties (Nickel et al. 2016). According to García-Durán et al. (2014), 3-way models such as RESCAL (Nickel et al. 2011; 2012) are more prone to overfitting, since they typically have a larger number of parameters. It is possible to extend the proposed model to RESCAL, whose score for a $\langle s, p, o \rangle$ triple is $\mathbf{e}_s^T \mathbf{W}_p \mathbf{e}_o$. For instance, it is easy to show that $\mathbf{e}_s^T \mathbf{W}_p \mathbf{e}_o = \mathbf{e}_o^T \mathbf{W}_p^T \mathbf{e}_s$. However, extending the proposed method to more complex 3-way models, such as the latent factor model proposed by Jenatton et al. (2012) or the ER-MLP model (Dong et al. 2014) can be less trivial.

7 Evaluation

We evaluate the proposed schema-based soft constraints on three datasets: WordNet, DBpedia and YAGO3. Each dataset is composed by a *training*, a *validation* and a *test* set of triples, as summarized in Table 1. All material needed for reproducing the experiments in this paper is available online[2].

WordNet (Miller 1995) is a lexical knowledge base for the English language, where entities correspond to word senses, and relationships define lexical relations between them: we use the version made available by Bordes et al. (2013).

[2] At https://github.com/pminervini/neural-schema-regularization.

Table 1. Statistics for the datasets used in experiments

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#training	#validation	#test
WordNet	40,943	18	141,442	5,000	5,000
DBpedia	32,510	7	289,825	5,000	5,000
YAGO3	123,182	37	1,079,040	5,000	5,000

YAGO3 (Mahdisoltani et al. 2015) is a large knowledge graph automatically extracted from several sources: our dataset is composed by facts stored in the YAGO3 Core Facts component of YAGO3.

DBpedia (Auer et al. 2007) is a knowledge base created extracting structured, multilingual knowledge from Wikipedia, and made available using Semantic Web and Linked Data standards. We consider a fragment extracted following the indications from Krompaß et al. (2014), by considering relations in the music domain[3].

The axioms we used in experiments are simple common-sense rules, and are listed in Table 1.

Evaluation Metrics – For evaluation, for each test triple $\langle s, p, o \rangle$, we measure the quality of the ranking of each test triple among all possible subject and object substitutions $\langle \tilde{s}, p, o \rangle$ and $\langle s, p, \tilde{o} \rangle$, with $\tilde{s}, \tilde{o} \in \mathcal{E}$. Mean Reciprocal Rank (MRR) and Hits@k as described by Bordes et al. (2013); Nickel et al. (2016); Trouillon et al. (2016) are widely adopted evaluation measures for evaluating knowledge graph completion algorithms. The measures are reported in the *raw* and *filtered* settings (Bordes et al. 2013). In the *filtered* setting, metrics are computed after removing all the other positive (true) triples that appear in either training, validation or test set from the ranking, whereas in the *raw* setting these are not removed. The filtered setting is motivated by observing that ranking a positive test triple after another true triple should not be considered a mistake (Bordes et al. 2013).

Evaluation Setting – In our experiments we consider three knowledge graph embedding models – TransE, ComplEx and DistMult, as described in Sect. 3. For evaluating the effectiveness of the proposed method, we train them using both the standard loss function $\mathcal{J}$, defined in Eq. (2), and the proposed schema-aware loss function $\mathcal{J}_{\mathcal{S}}$, defined in Eq. (10). Models trained by using the proposed method are denoted by the R superscript.

For each model and dataset, hyper-parameters were selected on the validation set by grid search. Specifically, we selected the embedding size $k \in \{20, 50, 100, 150\}$, the regularization weight $\lambda \in \{0, 10^{-4}, 10^{-2}, \ldots, 10^{6}\}$ and, in TransE, the norm $\|\cdot\|$ is selected across the L_1 and the L_2 norm.

[3] Following Krompass et al. (2014), such relations are album, associated band, associated musical artist, genre, musical artist, musical band, and record-Label.

Table 2. Link prediction results (Hits@k and Mean Reciprocal Rank, filtered setting) on WordNet, DBpedia and YAGO3.

	WordNet				**DBpedia**				**YAGO3**			
	Hits@N (%)			MRR	Hits@N (%)			MRR	Hits@N (%)			MRR
	3	5	10		3	5	10		3	5	10	
TransE	79.9	87.3	91.1	0.452	44.3	52.6	59.0	0.245	32.4	40.7	50.5	0.214
TransE^R	**86.9**	**91.6**	**93.3**	**0.566**	**47.8**	**54.0**	**60.0**	**0.256**	**33.4**	**42.5**	**52.0**	**0.248**
DistMult	91.7	93.2	94.2	0.840	44.6	**50.6**	55.7	0.371	**29.9**	**37.2**	46.3	**0.260**
DistMult^R	**92.4**	**93.8**	**94.9**	**0.851**	**44.9**	**50.6**	**55.8**	**0.381**	**29.9**	**37.2**	**46.4**	**0.260**
ComplEx	94.2	94.4	94.6	0.939	52.7	54.2	55.8	0.486	**34.8**	41.5	49.9	**0.304**
ComplEx^R	**94.3**	**94.5**	**94.7**	**0.940**	**53.1**	**54.3**	**55.9**	**0.503**	34.7	**41.6**	**50.0**	**0.304**

Similarly to Yang et al. (2015) we set the margin $\gamma = 1$ and, for each combination of hyper-parameters, we train each model for 1000 epochs. The learning rate in Stochastic Gradient Descent was initially set to 0.1, and then adapted during training by AdaGrad.

Results – We report test results in terms of raw and filtered Mean Reciprocal Rank (MRR), and filtered Hits@k in Table 2. For both the MRR and Hits@k metrics, the higher the results on the test set, the better.

We can see that, in every case, the proposed method – which relies on regularizing relation embeddings by leveraging background knowledge – improves the generalization abilities for each of the models. Results are especially evident for TransE, which largely benefits from the novel regularizer. For instance we can see that, in the WordNet case, the Hits@10 improves from 91.1 to 93.3, while the Mean Reciprocal Rank improves from 0.452 to 0.566. For the remaining models we can only notice marginal improvements, probably because they already able to capture the patterns encoded by the background knowledge.

In Fig. 2 we can see a set of trained WordNet predicate embeddings (using the model TransE), where relationships predicates are described in the axioms in Fig. 1. We can immediately see that, if $p \equiv q^-$, *i.e.* p is the inverse of q, then $\mathbf{r}_p \approx -\mathbf{r}_q$, which means that their embeddings $\mathbf{r}_p$ and $\mathbf{r}_q$ will be similar but will have opposite sign. On the left we set $\lambda = 0$, *i.e.* we do not enforce any soft constraint: we can see that the model is naturally inclined to assign opposite sign embeddings to relations such as PART OF and HAS PART, and HYPONYM and HYPERNYM; however, there is still some error margin in such an assignment, possibly due to the incompleteness of the knowledge graph. On the right we set $\lambda = 10^6$, *i.e.* we enforce the relationships between predicate embeddings via soft constraints: we can see that the aforementioned error margin in modeling the relationships between predicate embeddings is greatly reduced, improving the generalization properties of the model and establishing new state-of-the-art link prediction results on several datasets.

Axioms		
HAS PART	≡	PART OF$^-$
HYPERNYM	≡	HYPONYM$^-$
INSTANCE HYPERNYM	≡	INSTANCE HYPONYM$^-$
M. HOLONYM	≡	S. MERONYM$^-$
M. OF DOMAIN REGION	≡	S DOMAIN REGION OF$^-$
M. OF DOMAIN TOPIC	≡	S. DOMAIN TOPIC OF$^-$
M. OF DOMAIN USAGE	≡	S. DOMAIN USAGE OF$^-$
DER. RELATED FORM	≡	DER. RELATED FORM$^-$
VERB GROUP	≡	VERB GROUP$^-$
ASSOC. BAND	≡	ASSOC. MUSICAL ARTIST
MUSICAL BAND	≡	MUSICALARTIST
ISMARRIEDTO	≡	ISMARRIEDTO$^-$
HASNEIGHBOR	≡	HASNEIGHBOR$^-$

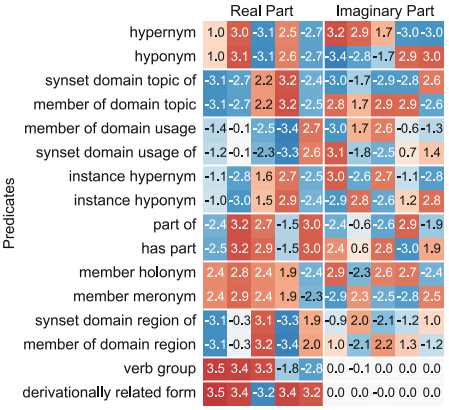

Fig. 1. Axioms used with WORDNET, DBPEDIA and YAGO3 (left) and WORDNET predicate embeddings learned by COMPLEX (right). Note that if $p \equiv q^-$ (*e.g.* PART OF and HAS PART) then $\mathbf{r}_p \approx \overline{\mathbf{r}_q}$, *i.e.* $\mathbf{r}_p$ and $\mathbf{r}_q$ have similar real parts and similar but opposite sign imaginary parts.

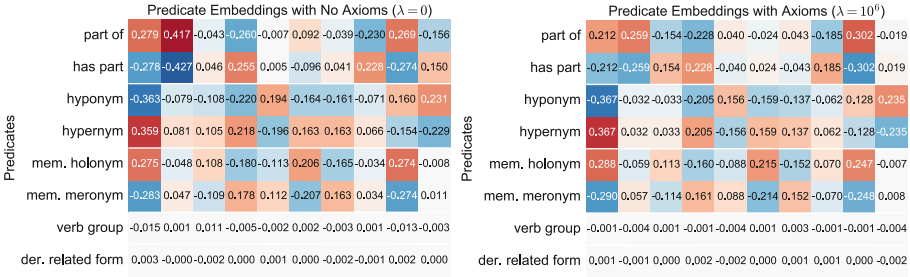

Fig. 2. WORDNET predicate embeddings learned using the TRANSE model, with $k = 10$ and regularization weight $\lambda = 0$ (left) and $\lambda = 10^6$ (right) – embeddings are represented as a heatmap, with values ranging from larger (red) to smaller (blue). Note that, assuming the axiom $p \equiv q^-$ holds, using the proposed method leads to predicate embeddings such that $\mathbf{r}_p \approx -\mathbf{r}_q$. (Color figure online)

A similar phenomenon in Fig. 1 (right), where predicated embeddings have been trained using COMPLEX: we can see that the model is naturally inclined to assign complex conjugate embeddings to inverse relations and, as a consequence, nearly-zero imaginary parts to the embeddings of symmetric predicates – since it is the only way of ensuring $\mathbf{r}_p \approx \overline{\mathbf{r}_p}$. However, we can enforce such relationships explicitly by means of model-specific regularizers, for increasing the predictive accuracy and generalization abilities of the models.

Table 3. Average number of seconds required for training.

	Plain	Regularized
WORDNET	31.7 s	32.0 s
DBPEDIA	57.9 s	58.5 s
YAGO3	220.7 s	221.3 s

We also benchmarked the computational overhead introduced by the novel regularizers by timing the training time for unregularized (plain) models and for

regularized ones – results are available in Table 3. We can see that the proposed method for leveraging background schema knowledge during the learning process adds a negligible overhead to the optimization algorithm – less than 10^{-1} s per epoch.

8 Conclusions and Future Works

In this work we introduced a novel and scalable approach for leveraging background knowledge into neural knowledge graph embeddings. Specifically, we proposed a set of background knowledge-driven regularizers on the relation embeddings, which effectively enforce a set of desirable algebraic relationships among the distributed representations of relation types. We showed that the proposed method improves the generalization abilities of all considered models, yielding more accurate link prediction results without impacting on the scalability properties of neural link prediction models.

Future Works

A promising research direction consists in leveraging more sophisticated background knowledge – *e.g.* in the form of First-Order Logic rules – in neural knowledge graph embedding models. This can be possible by extending the model in this paper to regularize over subgraph pattern embeddings (such as *paths*), so to leverage relationships between such patterns, rather than only between predicates. Models for embedding subgraph patterns have been proposed in the literature – for instance, see (Niepert 2016; Guu et al. 2015). For instance, it can be possible to enforce an equivalency between the path $\textsc{parentOf} \circ \textsc{parentOf}$ and $\textsc{grandParentOf}$, effectively incorporating a First-Order rule in the model, by regularizing over their embeddings.

Furthermore, a future challenge is also extending the proposed method to more complex models, such as ER-MLP (Dong et al. 2014), and investigating how to mine rules by extracting regularities from the latent representations of knowledge graphs.

Acknowledgements. This work was supported by the TOMOE project funded by Fujitsu Laboratories Ltd., Japan and Insight Centre for Data Analytics at National University of Ireland Galway (supported by the Science Foundation Ireland grant 12/RC/2289).

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52

Baroni, M., Bernardi, R., Do, N-Q., Shan, C.: Entailment above the word level in distributional semantics. In: EACL, pp. 23–32. The Association for Computer Linguistics (2012)

Bollacker, K.D., Cook, R.P., Tufts, P.: Freebase: a shared database of structured general human knowledge. In: AAAI, pp. 1962–1963. AAAI Press (2007)
Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
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)
Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI. AAAI Press (2010)
Chang, K.-W., Yih, W., Yang, B., Meek, C.: Typed tensor decomposition of knowledge bases for relation extraction. In: EMNLP, pp. 1568–1579. ACL (2014)
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: KDD, pp. 601–610. ACM (2014)
Duchi, J.C., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
García-Durán, A., Bordes, A., Usunier, N.: Effective blending of two and three-way interactions for modeling multi-relational data. In: ECML-PKDD, pp. 434–449 (2014)
Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: EMNLP, pp. 318–327. The Association for Computational Linguistics (2015)
Hayes, P., Patel-Schneider, P.: RDF 1.1 semantics. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/
Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: NIPS, pp. 3176–3184 (2012)
Krompass, D., Nickel, M., Tresp, V.: Large-scale factorization of type-constrained multi-relational data. In: DSAA, pp. 18–24. IEEE (2014)
Krompaß, D., Nickel, M., Tresp, V.: Querying factorized probabilistic triple databases. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 114–129. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_8
Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 640–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_37
Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: a knowledge base from multilingual wikipedias. In: CIDR (2015). www.cidrdb.org
Meseguer, P., Rossi, F., Schiex, T.: Soft constraints. In: Handbook of Constraint Programming, of Foundations of Artificial Intelligence, vol. 2, pp. 281–328. Elsevier (2006)
Miller, G.A.: WordNet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
Minervini, P., d'Amato, C., Fanizzi, N., Esposito, F.: Leveraging the schema in latent factor models for knowledge graph completion. In: SAC, pp. 327–332. ACM (2016)
Nickel, M., Tresp, V., Kriegel, H.-P.: A three-way model for collective learning on multi-relational data. In: ICML, pp. 809–816 (2011)
Nickel, M., Tresp, V., Kriegel, H.-P.: Factorizing YAGO: scalable machine learning for linked data. In: WWW, pp. 271–280. ACM (2012)
Nickel, M., Jiang, X., Tresp, V.: Reducing the rank in relational factorization models by including observable patterns. In: NIPS, pp. 1179–1187 (2014)
Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)

Niepert, M.: Discriminative Gaifman models. In: NIPS, pp. 3405–3413 (2016)
Rocktäschel, T., Singh, S., Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction. In: HLT-NAACL, pp. 1119–1129. The Association for Computational Linguistics (2015)
Schneider, M.: OWL 2 web ontology language RDF-based semantics, 2nd edn. W3C recommendation, W3C, December 2012. http://www.w3.org/TR/2012/REC-owl2-rdf-based-semantics-20121211/
Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW, pp. 697–706. ACM (2007)
Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML, of JMLR Workshop and Conference Proceedings, vol. 48, pp. 2071–2080. JMLR. org (2016)
Wang, Q., Wang, B., Guo, L.: Knowledge base completion using embeddings and rules. In: IJCAI, pp. 1859–1866. AAAI Press (2015)
Yang, B., Yih, W-t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015, May 2015