Adversarial Rule Injection in Knowledge Graph Embeddings

Pasquale Minervini[1] Thomas Demeester[2] Tim Rocktäschel[1]
Sebastian Riedel[1]
[1] University College London, London, UK
{ p.minervini, t.rocktaschel, s.riedel }@cs.ucl.ac.uk
[2] Ghent University - iMinds, Ghent, Belgium
tdmeeste@intec.ugent.be

ABSTRACT

Knowledge Graph embedding models currently hold the state-of-the-art in many link prediction and knowledge base completion tasks. However, a main challenge consists in efficiently incorporating background and common-sense knowledge into such models. Early approaches regularize relation and entity representations by grounding first-order logic rules; however, grounding does not scale to domains with a large number of entities and relations. A recent approach regularize relation representations by imposing ordering relationships over non-negative embeddings; however, it can only model a very limited set of rules. In this paper we propose *Adversarial Rule Injection*, an highly scalable method for incorporating general Horn clauses into distributed representations for Knowledge Graph embedding. Specifically, we use a form of *adversarial training* by iteratively: 1) finding *counter-examples* that violate rules in the embedding space by maximizing a violation loss, and 2) adjusting the model parameters so to minimize the violation loss. Surprisingly we find that ....

1. INTRODUCTION

Knowledge Graphs are graph-structured Knowledge Bases, where facts are represented in the form of relationships between entities: they are powerful instruments in search, analytics and data integration. Developments in the area of Knowledge Graphs were initially driven by academic efforts, such as DBPEDIA [1], FREEBASE [2], YAGO [27] and NELL [4]. At the time of this writing, Knowledge Graphs are also widely popular in industry applications, such as the FREEBASE-based Google's Knowledge Graph [1] and Knowledge Vault [8] projects, which support the company's search and smart assistance services.

However, despite the engineering efforts, real world Knowledge Graphs are often far from being complete. For instance, consider FREEBASE: 71% of the persons described in the knowledge base have no known place of birth, 75% have no known nationality, and coverage for less frequent relations can be even lower [8]. Similarly, in DBPEDIA 66% of the persons have no known place of birth, while 58% of the scientists are missing a fact stating what they are known for [15].

Therefore, in this paper we focus on the problem of *predicting missing links* in Knowledge Graphs, so to discover new facts: in the literature, this problem is also known as *link prediction* and *Knowledge Base completion*. Current successful link prediction methods heavily rely on learned distributed vector representations of entities and relations in the Knowledge Graph [21, 3, 23, 38, 29]. we refer to [20] for a detailed description of such methods. Although such models are able to learn robust representations of entities and relations from large amount of data, they lack common-sense knowledge and reasoning.

1.1 Related Works

Combining neural methods with symbolic common-sense knowledge, *e.g.* in the form of implication rules, is an actively researched area [34, 32, 30, 25, 7] In [26] authors regularize entity-tuple and relation embeddings via First-Order Logic rules: every rule is propositionalized, and a differentiable loss term is added for every propositional rule. However, this approach does not scale to large knowledge bases: even a simple rule such as

$$\forall x, y, z : \textsc{siblingOf}(x, y) \land \textsc{parentOf}(y, z) \implies \textsc{uncleOf}(x, z)$$

would result in a very large number of loss terms. In [7] authors overcome this problem by minimizing an upper bound of the loss that encourages the implication between relations to hold, entirely independent from the number of entities. However, their approach – a variant MODEL F [**?**] – is only able to model a very restricted set of rules in the form:

$$\forall x, y : p(x, y) \implies q(x, y),$$

where p and q are two arbitrary relations.

1.2 Contribution

Adversarial examples are examples crafted for significantly increasing the loss functional of a machine learning model, while *adversarial training* is the process of training a model to correctly classify both training and adversarial examples [28, 12].

[1] https://developers.google.com/knowledge-graph/

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

In this paper we present a method, named *Adversarial Rule Injection* (ARI), for incorporating general First-Order Logic rules in Knowledge Graph embedding models. ARI relies on an *adversarial training architecture*, where: 1) An *adversary* finds counter-examples that maximize a rule violation loss, and 2) A tunes the parameters so to minimize the rule violation loss on such counter-examples.

In particular, in ARI, counter-examples are found in the *continuous embedding space* rather than in the discrete fact space. This has several advantages, namely:

Scalability: It decouples the complexity of the method from the number of entities in the Knowledge Base, which can be in the order of millions or more.

Generalizability: It enforces rules to hold everywhere in the embedding space, even on previously unseen entity embeddings.

Flexibility: It can be used jointly with any Knowledge Graph embedding model, without making any assumption on its architecture.

In Statistical Relational Learning [9], the proposed approach falls in the category of *lifted inference and learning* [22, 6]: by imposing the desired constraints on the whole entity embedding space, we can efficiently reason about groups of objects as a whole by exploiting symmetries in the relational structure of the model. This allows imposing a large number of First-Order rules while learning the distributed representations of all entities and relations in the Knowledge Graph. Two fundamental advantages with respect to the method in [26] are the drastically lower computation times and the fact that rules are enforced to hold on the whole embedding space, even on previously unseen entity embeddings – thus covering the cases where not all entity embeddings are visible at training time, such as in Row-Less Universal Schema [31]. The method in [7] is also *lifted*, but can only handle a very small subset of First-Order rules.

2. BACKGROUND

In this section we introduce Knowledge Graphs, and describe several models from the literature for learning continuous representations of the entities and relations they describe.

2.1 Knowledge Graphs

Knowledge Graphs represent information in the form of entities and relationships between them [20]. Entities can be anything, including persons, documents, physical objects and abstract concepts. In the following, we assume the Knowledge Graph follows the *Resource Description Framework* (RDF) [36] data model: RDF is a W3C recommended framework for representing information about entities, also referred to as *resources*.

Let $\mathcal{E}$ and $\mathcal{R}$ represent a set of entities and a set of relations, respectively. A RDF Knowledge Base, also referred to as *RDF graph*, is defined as a set of $\langle s, p, o \rangle \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ triples, each consisting of a subject $s \in \mathcal{E}$, a predicate $p \in \mathcal{R}$ and an object $o \in \mathcal{E}$: s and o are entities, while p is a relation type. Each $\langle s, p, o \rangle$ triple encodes the statement "s has a relationship p with o", which can be encoded in predicate logic by the formula $p(s, o)$.

An RDF graph can be represented as a labeled directed multi-graph, in which each triple is represented as an edge connecting two nodes: source and target nodes represent the subject and the object of the triple, and the edge label represents the predicate. Note that RDF and Knowledge Graphs in general adhere to the *Open World Assumption* [14]: a triple not in the graph does not imply that the corresponding statement is false, but rather that its truth value is *unknown*, *i.e.* it cannot be known from the RDF graph.

Example 1. *Consider the following statement:* "Rammstein is a German music band, and KMFDM is also a music band." *It can be expressed by the following RDF triples:*

Subject	***Predicate***	***Object***
⟨ Rammstein,	type,	Music Band ⟩
⟨ Rammstein,	nationality,	German ⟩
⟨ KMFDM,	type,	Music Band ⟩

The fact that the triple ⟨KMFDM, nationality, German⟩ *is not present in the Knowledge Graph does not imply that KMFDM is not German, but rather that we do not know whether KMFDM is German or not.*

In the following, we denote by $\mathcal{S} \triangleq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ the set of *possible triples* that can be represented by using the entities and relations in a Knowledge Graph, each representing a distinct $\langle s, p, o \rangle$ statement.

2.2 Knowledge Graph Embeddings

In Knowledge Graph embedding models, the link prediction score for each $\langle s, p, o \rangle$ triple (fact) is defined as a function of the distributed representations – or *embeddings* – associated with the subject s, the predicate p and the object o of the triple. Each different model is characterized by its scoring function $\phi(\cdot \, ; \Theta)$, where the model parameters Θ correspond to the embedding vectors of all entities and relations in the graph.

Several models have been proposed for solving the link prediction problem. Three largely popular models, due to their effectiveness, are the Translating Embeddings model [3], the Bilinear-Diagonal model [38] and, more recently, the Complex Embeddings model [29]. Such models can scale to very large and Web-scale Knowledge Graphs, thanks to: 1) A space complexity that grows *linearly* with the number of entities and relations in the Knowledge Graph; and 2) Efficient and scalable scoring functions and parameters learning procedures.

2.2.1 The Translating Embeddings Model

In the Translating Embeddings model (or TransE), each entity $e \in \mathcal{E}$ is associated with a unique continuous *embedding vector* $\mathbf{e}_e \in \mathbb{R}^k$, where $k \in \mathbb{N}$ is a user-defined hyperparameter. Each vector $\mathbf{e}_e$ can be interpreted as a collection of continuous *latent features* describing e [20]. Similarly, each predicate $p \in \mathcal{R}$ is associated with a unique continuous embedding vector $\mathbf{r}_p \in \mathbb{R}^k$.

Given a triple $\langle s, p, o \rangle$, it prediction score $\phi_{spo}^{\textsc{TransE}}$ is defined by the embedding vectors $\mathbf{e}_s, \mathbf{r}_p, \mathbf{e}_o \in \mathbb{R}^k$, respectively associated with the subject s, the predicate p and the object o of the triple. Specifically, the score for a triple $\langle s, p, o \rangle$ is defined as follows:

$$\phi_{\textsc{TransE}}(\langle s, p, o \rangle; \Theta) \triangleq - \|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| \, ,$$

where $\| \cdot \|$ denotes either the L_1 or the L_2 norm, and $\|\mathbf{x}_1 - \mathbf{x}_2\|$ denotes the distance between vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.

2.2.2 The Bilinear-Diagonal Model

The Bilinear-Diagonal model (or DistMult) is a variant of TransE, where: 1) The predicate embedding defines a *scaling* operation, and 2) The *dot product* is used for assessing the similarity between two vectors. In this model, the score of a triple $\langle s, p, o \rangle$ is defined as follows:

$$\phi_{\textsc{DistMult}}(\langle s, p, o \rangle; \Theta) \triangleq \langle \mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o \rangle,$$

where, given $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^k$, $\langle \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \rangle \triangleq \sum_{i=1}^{k} \mathbf{x}_{1,i} \mathbf{x}_{2,i} \mathbf{x}_{3,i}$ is the standard component-wise multi-linear dot product.

2.2.3 The Complex Embeddings Model

The Complex Embeddings model (or ComplEx) is related to DistMult, but uses complex-valued embeddings while retaining the mathematical definition of the dot product. In this model, the score of a triple $\langle s, p, o \rangle$ is defined as follows:

$$\begin{aligned}\phi_{\textsc{ComplEx}}(\langle s, p, o \rangle; \Theta) &\triangleq \mathrm{Re}\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right) \\ &= \langle \mathrm{Re}\left(\mathbf{r}_p\right), \mathrm{Re}\left(\mathbf{e}_s\right), \mathrm{Re}\left(\mathbf{e}_o\right) \rangle \\ &\quad + \langle \mathrm{Re}\left(\mathbf{r}_p\right), \mathrm{Im}\left(\mathbf{e}_s\right), \mathrm{Im}\left(\mathbf{e}_o\right) \rangle \\ &\quad + \langle \mathrm{Im}\left(\mathbf{r}_p\right), \mathrm{Re}\left(\mathbf{e}_s\right), \mathrm{Im}\left(\mathbf{e}_o\right) \rangle \\ &\quad - \langle \mathrm{Im}\left(\mathbf{r}_p\right), \mathrm{Im}\left(\mathbf{e}_s\right), \mathrm{Re}\left(\mathbf{e}_o\right) \rangle,\end{aligned}$$

where $\mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o \in \mathbb{C}^k$ are complex vectors, $\overline{\mathbf{x}}$ denotes the complex conjugate of $\mathbf{x}$ [2], 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.

2.2.4 Learning the Model Parameters

In [3, 38, 20], authors estimate the model parameters by minimizing the a ranking loss, where negative examples are obtained by *corrupting* the triples in the Knowledge Graph. More formally, given a triple $\langle s, p, o \rangle$, negative examples are generated by the corruption process defined by $\delta(\,\cdot\,)$:

$$\delta(\langle s, p, o \rangle) \triangleq \{\langle \tilde{s}, p, o \rangle \mid \tilde{s} \in \mathcal{E}\} \cup \{\langle s, p, \tilde{o} \rangle \mid \tilde{o} \in \mathcal{E}\}, \qquad (1)$$

i.e. given a triple, the process generates a set of 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) [8]: 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 ranking loss is then defined as follows:

$$\mathcal{J}(\Theta) \triangleq \sum_{t \in \mathcal{G}} \sum_{\tilde{t} \in \delta(t)} \left[\gamma - \phi(t; \Theta) + \phi(\tilde{t}; \Theta)\right]_+, \qquad (2)$$

where $[x]_+ \triangleq \max\{0, x\}$. Note that the loss function in Eq. 2 will reach its global minimum 0 iff, for each pair of positive and negative examples t and $\tilde{t}$, the score of the (true) triple t is higher with a margin of at least γ than the score of the (missing) triple $\tilde{t}$. The optimal parameters can be learned by solving the following minimization problem:

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

The norm constraints on the entity embeddings prevent to trivially solve the optimization problem by increasing the norm of the embedding vectors.

3. ADVERSARIAL INJECTION OF FIRST-ORDER RULES

In this section we propose *Adversarial Rule Injection* (ARI) an efficient method for incorporating first-order rules in Knowledge Graph embedding models.

An *atom* is a fact (triple) that can have variables at the subject and/or object position. A *First-Order rule* consists of an implication between a *body* and an *head*, where the head is a single atom, and the body is a set of atoms. Let $\mathcal{V}$ be a set of universally quantified set of variables. We denote a rule with head $q(x_1, x_2)$ and body $\{B_1, \ldots, B_n\}$ by an implication:

$$B_1 \wedge B_2 \wedge \ldots \wedge B_m \Rightarrow q(x_1, x_2) \qquad (4)$$

where each $x_1, x_2 \in \mathcal{V}$ are two variables in $\mathcal{V}$, $q \in \mathcal{R}$, and each B_i is an atom $p(x_{i,1}, x_{i,2})$ with $p \in \mathcal{R}$ and $x_{i,1}, x_{i,2} \in \mathcal{V}$. We abbreviate the rule in Eq. 4 by $\vec{B} \Rightarrow q(x_1, x_2)$.

An *instantiation* of a rule is a copy of the rule where all variables in $\mathcal{V}$ have been substituted by entities in $\mathcal{E}$. According to the rule in Eq. 4 is that for any instantiation of the rule, if the body holds, then the head also holds.

Example 2. *Consider the sentence "*An uncle is a sibling of a parent*". It can be encoded by the following rule:*

$$\textsc{siblingOf}(x_1, x_2) \wedge \textsc{parentOf}(x_2, x_3) \Rightarrow \textsc{uncleOf}(x_1, x_3), \qquad (5)$$

where $\mathcal{V} = \{x_1, x_2, x_3\}$ is the set of variables, $\textsc{uncleOf}(x_1, x_3)$ *is the head and* $\textsc{siblingOf}(x_1, x_2) \wedge \textsc{parentOf}(x_2, x_3)$ *the body of the rule.*

Assume $\{\texttt{Mark}, \texttt{John}, \texttt{Paul}\} \subseteq \mathcal{E}$: according to the rule in Eq. 5, if $\textsc{siblingOf}(\texttt{Mark}, \texttt{John})$ *and* $\textsc{parentOf}(\texttt{John}, \texttt{Paul})$ *hold, then* $\textsc{uncleOf}(\texttt{Mark}, \texttt{Paul})$ *also holds.*

3.1 Grounded Loss Formulation

Following [7], an implication rule in the form:

$$p(x_1, x_2) \Rightarrow q(x_1, x_2), \qquad (6)$$

with $\mathcal{V} = \{x_1, x_2\}$, can be enforced by requiring that, for every pair of entities $\langle s, o \rangle \in \mathcal{E} \times \mathcal{E}$, the triple $\langle s, p, o \rangle$ is considered less likely than the triple $\langle s, q, o \rangle$, *i.e.* :

$$\forall s, o \in \mathcal{E} : \phi(\langle s, p, o \rangle) \leq \phi(\langle s, q, o \rangle). \qquad (7)$$

Note that if $\langle s, p, o \rangle$ is a true fact with a high score $\phi(\langle s, p, o \rangle)$, and the fact $\langle s, q, o \rangle$ has an higher score, it must also be true, but not vice-versa. We can thus enforce an implication rule by minimizing a loss term with a separate contribution for every $\langle s, o \rangle \in \mathcal{E} \times \mathcal{E}$, adding up to the loss function in Eq. 2 if the corresponding inequality is not satisfied. The implication loss for the rule in Eq. 6 can be formulated as:

$$\mathcal{J}_G(\Theta) = \sum_{s \in \mathcal{E}} \sum_{o \in \mathcal{E}} \left[\phi(\langle s, p, o \rangle; \Theta) - \phi(\langle s, q, o \rangle; \Theta)\right]_+. \qquad (8)$$

Note that the loss function $\mathcal{J}_G(\Theta)$ in Eq. 8 is reaches its global minimum 0 when parameters Θ match the constraints in Eq. 7, and it is strictly positive otherwise.

However, a limitation of implication rules in the form provided in Eq. 6 is that the body is limited to one single atom. Following [26], the score for a conjunction of atoms can be computed using *t-norms* [13], for instance:

- The *Minimum t-norm*, also called *Zadeh's t-norm*:

  $$\phi(\langle s_1, p_1, o_1 \rangle \wedge \langle s_2, p_2, o_2 \rangle) = \min\left\{\phi(\langle s_1, p_1, o_1 \rangle), \phi(\langle s_2, p_2, o_2 \rangle)\right\}$$

[2] 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 Solving the minimax problem in Eq. 12 via Stochastic Gradient Descent

```
Require: No. of training epochs $\tau_a, \tau_d, \tau$
Randomly initialise model parameters $\Theta_0$
for $i \in \langle 1, \ldots, \tau \rangle$ do
    $\bar{\mathbf{E}}_i \leftarrow$ FindAversarialExamples($\Theta_{i-1}, \tau_a$)
    $\Theta_i \leftarrow$ TrainModelParameters($\bar{\mathbf{E}}_i, \tau_d$)
end for
return $\Theta_\tau$

function FindAdversarialExamples($\Theta, \tau_a$)
    {Solve the loss maximization problem in Eq. 10}
    Randomly initialise $\bar{\mathbf{E}}_0$
    for $i \in \langle 1, \ldots, \tau_a \rangle$ do
        $\mathbf{e} \leftarrow \mathbf{e} / \|\mathbf{e}\|, \forall \mathbf{e} \in \bar{\mathbf{E}}$
        $g_i \leftarrow \nabla_{\bar{\mathbf{E}}} \mathcal{J}_A(\bar{\mathbf{E}})$
        $\bar{\mathbf{E}}_i \leftarrow \bar{\mathbf{E}}_{i-1} + \eta_i g_i$
    end for
    return $\bar{\mathbf{E}}_\tau$
end function

function TrainModelParameters($\bar{\mathbf{E}}, \tau_d$)
    {Solve the loss minimization problem in Eq. 11}
    Randomly initialise $\Theta_0$
    for $i \in \langle 1, \ldots, \tau_d \rangle$ do
        $\mathbf{e}_e \leftarrow \mathbf{e}_e / \|\mathbf{e}_e\|, \forall e \in \mathcal{E}$
        $\mathcal{B} \leftarrow$ SampleBatch($\mathcal{G}, n$)
        $g_i \leftarrow \nabla_\Theta \sum_{\langle t, \tilde{t} \rangle \in \mathcal{B}} \left[\gamma - \phi(t; \Theta_{i-1}) + \phi(\tilde{t}; \Theta_{i-1})\right]_+$
                $+ \lambda \mathcal{J}_G(\bar{\mathbf{E}}; \Theta)$
        $\Theta_i \leftarrow \Theta_{i-1} - \eta_i g_i$
    end for
    return $\Theta_\tau$
end function
```

- The *Product t-norm*:

$$\phi(\langle s_1, p_1, o_1 \rangle \wedge \langle s_2, p_2, o_2 \rangle) = \phi(\langle s_1, p_1, o_1 \rangle) \cdot \phi(\langle s_2, p_2, o_2 \rangle)$$

Let $\mathcal{V} = \{x_1, \ldots, x_n\}$, and let $x_i \equiv e$ denote the assignment of entity $e \in \mathcal{E}$ to variable $x_i \in \mathcal{V}$. Given an implication rule in the form provided in Eq. 4, its loss can be computed as follows:

$$\mathcal{J}_G(\Theta) = \sum_{\substack{e_1 \in \mathcal{E} \\ x_1 \equiv e_1}} \cdots \sum_{\substack{e_n \in \mathcal{E} \\ x_n \equiv e_n}} \left[\phi(\vec{B}; \Theta) - \phi(\langle x_1, q, x_2 \rangle; \Theta)\right]_+ \tag{9}$$

However, note that the loss in Eq. 9 requires adding $\mathcal{O}(|\mathcal{E}|^n)$ independent terms to the loss function. Even for a small value of n, this can be unfeasible for real world Knowledge Graphs, which may contains millions of entities or more.

3.2 Lifted Adversarial Loss Formulation

The problem mentioned above can be avoided if, instead of enumerating $\mathcal{O}(|\mathcal{E}|^n)$ entity tuples, we iteratively find a set of *adversarial counter-examples* maximizing the loss function in Eq. 9, and then tune the model parameters Θ to minimize a linear combination of the losses in Eq. 2 and Eq. 9.

However, finding a tuple of entities $\langle e_1, \ldots, e_n \rangle \in \mathcal{E}^n$ maximizing the loss in Eq. 9 is a potentially intractable combinatorial problem. Efficiently finding discrete adversarial examples is an open problem in adversarial training of neural architectures [10].

In this work, we attack this problem by instead finding a tuple of *adversarial entity embeddings* $\bar{\mathbf{E}} \triangleq \langle \bar{\mathbf{e}}_1, \ldots, \bar{\mathbf{e}}_n \rangle \in \mathbb{R}^{k \times n}$ maximizing the loss in Eq. 9 using gradient-based optimization methods [11]. We will refer to the loss $\mathcal{J}_G$ evaluated on the adversarial embeddings as $\mathcal{J}_G(\bar{\mathbf{E}}; \Theta)$. For the sake of clarity, in the following we denote the score of a $\langle s, p, o \rangle$ triple $\phi(\langle s, p, o \rangle; \Theta)$ as $\phi(\mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o)$.

The resulting learning procedure, which we refer to as *Adversarial Rule Injection* (ARI), can be described as a minimax game where: 1) One agent (the *adversary*) finds a set of adversarial counter-examples maximizing the loss $\mathcal{J}_G$ in Eq. 8, and 2) another agent (the *discriminator*) tunes the model parameters Θ so to simultaneously minimize the losses $\mathcal{J}$ in Eq. 2 and $\mathcal{J}_G$ in Eq. 8.

The training procedure in ARI alternates between the following two objectives:

Adversary:

$$\begin{aligned} \underset{\bar{\mathbf{E}}}{\text{maximize}} \quad & \mathcal{J}_A(\bar{\mathbf{E}}) \triangleq \mathcal{J}_G(\bar{\mathbf{E}}; \Theta) \\ \text{subject to} \quad & \forall \mathbf{e} \in \bar{\mathbf{E}} \;\; \|\mathbf{e}\| = 1, \end{aligned} \tag{10}$$

Discriminator:

$$\begin{aligned} \underset{\Theta}{\text{minimize}} \quad & \mathcal{J}_D(\Theta) \triangleq \mathcal{J}(\Theta) + \lambda \mathcal{J}_G(\bar{\mathbf{E}}; \Theta) \\ \text{subject to} \quad & \forall e \in \mathcal{E} : \;\; \|\mathbf{e}_e\| = 1, \end{aligned} \tag{11}$$

where $\lambda \geq 0$ is a user-specified hyper-parameter specifying the weight of the rule-based loss in the training process. The learning procedure in ARI can thus be formalized by the following minimax problem:

$$\min_{\Theta} \max_{\bar{\mathbf{E}}} \mathcal{J}_D(\Theta) + \mathcal{J}_A(\bar{\mathbf{E}}). \tag{12}$$

In Alg. 1 we provide an algorithm for solving the minimax problem in Eq. 12.

4. RELATED WORKS

Research on leveraging rules when learning Knowledge Graph embeddings has been deeply important for new developments in the field of Knowledge Base completion. In [24, 26] authors provide a framework for jointly maximizing the probability of observed facts and propositionalised First-Order logic rules. In [32] authors show how different types of rules can be included after training the model by using Integer Linear Programming. In [33] authors propose a method for embedding facts and rules using matrix factorization. However, all these approaches ground the rules in the training data, limiting their scalability towards large Knowledge Bases containing a large number of entities. As mentioned in [7], such problems provide an important motivation for *lifted* rule injections methods that do not rely on the grounding of logic formulas. In [35] authors try to work around this problem by reasoning on a filtered subset of grounded facts.

In [37], authors propose using the Path Ranking Algorithm [17] for capturing long-range interactions between entities and modelling these by defining an additional loss term. The model in this paper differs substantially: we can inject arbitrarily complex First-Order logic rules rather than just paths between two entities.

In [5, 16, 15], authors 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 [18], 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 consider type information about entities: in this work we propose a method for leveraging equivalence and inversion axioms, which can be used jointly with the aforementioned methods.

In [8, 19, 32], authors propose combining observable patterns in the form of rules and latent features for link prediction tasks. However, in such models, rules are not used *during* the parameters learning process, but rather *after*, in an ensemble fashion.

The work in this paper is related to the Model FSL proposed in [7]: they use simple rules in the form of Eq. 6 for defining a partial ordering in the relation embeddings for a variant of Model F [23]. Their approach is *lifted* since they do not rely on entity embeddings for satisfying the rules. ARI extends and improves over Model FSL – in particular: 1) It can be used for injecting arbitrarily complex First-Order logic rules as in Eq. 4, and 2) It can be used jointly with any Knowledge Graph embedding model that provides a fact scoring function $\phi(\,\cdot\,; \Theta)$. It also improves over the work in [26], since it does not need to generate all possible instantiations of each rule.

5. EXPERIMENTS

6. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. Dbpedia: A nucleus for a web of open data. In K. Aberer et al., editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *LNCS*, pages 722–735. Springer, 2007.

[2] K. D. Bollacker, R. P. Cook, and P. Tufts. Freebase: A shared database of structured general human knowledge. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1962–1963. AAAI Press, 2007.

[3] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges et al., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2787–2795, 2013.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In M. Fox et al., editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010.* AAAI Press, 2010.

[5] K. Chang, W. Yih, B. Yang, and C. Meek. Typed tensor decomposition of knowledge bases for relation extraction. In A. Moschitti et al., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1568–1579. ACL, 2014.

[6] R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In L. P. Kaelbling et al., editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 1319–1325. Professional Book Center, 2005.

[7] T. Demeester, T. Rocktäschel, and S. Riedel. Lifted rule injection for relation embeddings. In J. Su et al., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1389–1399. The Association for Computational Linguistics, 2016.

[8] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In S. A. Macskassy et al., editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 601–610. ACM, 2014.

[9] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning.* The MIT Press, 2007.

[10] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2016.

[11] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 201r.

[13] M. M. Gupta and J. Qi. Theory of t-norms and fuzzy inference methods. *Fuzzy Sets Syst.*, 40(3):431–450, Apr. 1991.

[14] P. Hayes and P. Patel-Schneider. RDF 1.1 semantics. W3C recommendation, W3C, Feb. 2014. http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/.

[15] D. Krompaß, S. Baier, and V. Tresp. Type-constrained representation learning in knowledge graphs. In M. Arenas et al., editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *LNCS*, pages 640–655. Springer, 2015.

[16] D. Krompass, M. Nickel, and V. Tresp. Large-scale factorization of type-constrained multi-relational data. In *International Conference on Data Science and Advanced Analytics, DSAA 2014, Shanghai, China, October 30 - November 1, 2014*, pages 18–24. IEEE, 2014.

[17] N. Lao, T. M. Mitchell, and W. W. Cohen. Random walk inference and learning in A large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 529–539. ACL, 2011.

[18] P. Minervini, C. d'Amato, N. Fanizzi, and F. Esposito. Leveraging the schema in latent factor models for knowledge graph completion. In S. Ossowski, editor, *Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016*, pages 327–332. ACM, 2016.

[19] M. Nickel, X. Jiang, and V. Tresp. Reducing the rank in relational factorization models by including observable patterns. In Z. Ghahramani et al., editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1179–1187, 2014.

[20] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[21] M. Nickel, V. Tresp, and H. Kriegel. Factorizing YAGO: scalable machine learning for linked data. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 271–280. ACM, 2012.

[22] D. Poole. First-order probabilistic inference. In G. Gottlob et al., editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 985–991. Morgan Kaufmann, 2003.

[23] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin. Relation extraction with matrix factorization and universal schemas. In L. Vanderwende et al., editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 74–84. The Association for Computational Linguistics, 2013.

[24] T. Rocktaeschel, M. Bosnjak, S. Singh, and S. Riedel. Low-dimensional embeddings of logic. In *ACL Workshop on Semantic Parsing (SP'14)*, 2014.

[25] T. Rocktäschel and S. Riedel. Learning knowledge base inference with neural theorem provers. In J. Pujara, T. Rocktäschel, D. Chen, and S. Singh, editors, *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT 2016, San Diego, CA, USA, June 17, 2016*, pages 45–50. The Association for Computer Linguistics, 2016.

[26] T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In R. Mihalcea et al., editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1119–1129. The Association for Computational Linguistics, 2015.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In C. L. Williamson et al., editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706. ACM, 2007.

[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[29] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In M. Balcan et al., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016.

[30] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *CoRR*, abs/1511.06361, 2015.

[31] P. Verga and A. McCallum. Row-less universal schema. *CoRR*, abs/1604.06361, 2016.

[32] Q. Wang, B. Wang, and L. Guo. Knowledge base completion using embeddings and rules. In Q. Yang et al., editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1859–1866. AAAI Press, 2015.

[33] W. Y. Wang and W. W. Cohen. Learning first-order logic embeddings via matrix factorization. In S. Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2132–2138. IJCAI/AAAI Press, 2016.

[34] W. Y. Wang, K. Mazaitis, and W. W. Cohen. Structure learning via parameter learning. In J. Li et al., editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1199–1208. ACM, 2014.

[35] Z. Wei, J. Zhao, K. Liu, Z. Qi, Z. Sun, and G. Tian. Large-scale knowledge base completion: Inferring via grounding network sampling over selected instances. In J. Bailey et al., editors, *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1331–1340. ACM, 2015.

[36] D. Wood, M. Lanthaler, and R. Cyganiak. RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C, Feb. 2014. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[37] F. Wu, J. Song, Y. Yang, X. Li, Z. M. Zhang, and Y. Zhuang. Structured embedding via pairwise relations and long-range interactions in knowledge base. In B. Bonet et al., editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1663–1670. AAAI Press, 2015.

[38] B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations (ICLR)*, 2015.