Jointly Embedding Knowledge Graphs and Logical Rules

Shu Guo[†‡], **Quan Wang**[†‡*], **Lihong Wang**[§], **Bin Wang**[†‡], **Li Guo**[†‡]
[†]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[‡]University of Chinese Academy of Sciences, Beijing 100049, China
{guoshu,wangquan,wangbin,guoli}@iie.ac.cn
[§]National Computer Network Emergency Response Technical Team
Coordination Center of China, Beijing 100029, China
wlh@isc.org.cn

Abstract

Embedding knowledge graphs into continuous vector spaces has recently attracted increasing interest. Most existing methods perform the embedding task using only fact triples. Logical rules, although containing rich background information, have not been well studied in this task. This paper proposes a novel method of jointly embedding knowledge graphs and logical rules. The key idea is to represent and model triples and rules in a unified framework. Specifically, triples are represented as atomic formulae and modeled by the translation assumption, while rules represented as complex formulae and modeled by t-norm fuzzy logics. Embedding then amounts to minimizing a global loss over both atomic and complex formulae. In this manner, we learn embeddings compatible not only with triples but also with rules, which will certainly be more predictive for knowledge acquisition and inference. We evaluate our method with link prediction and triple classification tasks. Experimental results show that joint embedding brings significant and consistent improvements over state-of-the-art methods. Particularly, it enhances the prediction of new facts which cannot even be directly inferred by pure logical inference, demonstrating the capability of our method to learn more predictive embeddings.

1 Introduction

Knowledge graphs (KGs) provide rich structured information and have become extremely useful resources for many NLP related applications like word sense disambiguation (Wasserman-Pritsker et al., 2015) and information extraction (Hoffmann et al., 2011). A typical KG represents knowledge as multi-relational data, stored in triples of the form (*head entity*, *relation*, *tail entity*), e.g., (`Paris`, `Capital-Of`, `France`). Although powerful in representing structured data, the symbolic nature of such triples makes KGs, especially large-scale KGs, hard to manipulate.

Recently, a promising approach, namely knowledge graph embedding, has been proposed and successfully applied to various KGs (Nickel et al., 2012; Socher et al., 2013; Bordes et al., 2014). The key idea is to embed components of a KG including entities and relations into a continuous vector space, so as to simplify the manipulation while preserving the inherent structure of the KG. The embeddings contain rich semantic information about entities and relations, and can significantly enhance knowledge acquisition and inference (Weston et al., 2013).

Most existing methods perform the embedding task based solely on fact triples (Bordes et al., 2013; Wang et al., 2014; Nickel et al., 2016). The only requirement is that the learned embeddings should be compatible with those facts. While logical rules contain rich background information and are extremely useful for knowledge acquisition and inference (Jiang et al., 2012; Pujara et al., 2013), they have not been well studied in this task. Wang et al. (2015) and Wei et al. (2015) tried to leverage both embedding methods and logical rules for KG completion. In their work, however, rules are modeled separately from embedding methods, serving as post-processing steps, and thus will not help to obtain

*Corresponding author: Quan Wang.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 192–202,
Austin, Texas, November 1-5, 2016.

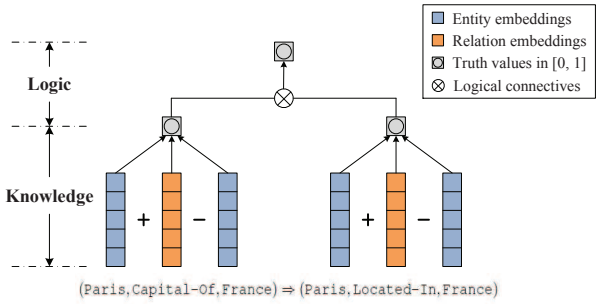

Figure 1: Simple illustration of KALE.

better embeddings. Rocktäschel et al. (2015) recently proposed a joint model which injects first-order logic into embeddings. But it focuses on the relation extraction task, and creates vector embeddings for entity pairs rather than individual entities. Since entities do not have their own embeddings, relations between unpaired entities cannot be effectively discovered (Chang et al., 2014).

In this paper we introduce KALE, a new approach that learns entity and relation Embeddings by jointly modeling Knowledge And Logic. Knowledge triples are taken as atoms and modeled by the translation assumption, i.e., relations act as translations between head and tail entities (Bordes et al., 2013). A triple (e_i, r_k, e_j) is scored by $\|\mathbf{e}_i + \mathbf{r}_k - \mathbf{e}_j\|_1$, where $\mathbf{e}_i$, $\mathbf{r}_k$, and $\mathbf{e}_j$ are the vector embeddings for entities and relations. The score is then mapped to the unit interval $[0, 1]$ to indicate the truth value of that triple. Logical rules are taken as complex formulae constructed by combining atoms with logical connectives (e.g., $\wedge$ and $\Rightarrow$), and modeled by t-norm fuzzy logics (Hájek, 1998). The truth value of a rule is a composition of the truth values of the constituent atoms, defined by specific logical connectives. In this way, KALE represents triples and rules in a unified framework, as atomic and complex formulae respectively. Figure 1 gives a simple illustration of the framework. After unifying triples and rules, KALE minimizes a global loss involving both of them to obtain entity and relation embeddings. The learned embeddings are therefore compatible not only with triples but also with rules, which will definitely be more predictive for knowledge acquisition and inference.

The main contributions of this paper are summarized as follows. (i) We devise a unified framework that jointly models triples and rules to obtain more predictive entity and relation embeddings. The new framework KALE is general enough to handle any type of rules that can be represented as first-order logic formulae. (ii) We evaluate KALE with link prediction and triple classification tasks on WordNet (Miller, 1995) and Freebase (Bollacker et al., 2008). Experimental results show significant and consistent improvements over state-of-the-art methods. Particularly, joint embedding enhances the prediction of new facts which cannot even be directly inferred by pure logical inference, demonstrating the capability of KALE to learn more predictive embeddings.

2 Related Work

Recent years have seen rapid growth in KG embedding methods. Given a KG, such methods aim to encode its entities and relations into a continuous vector space, by using neural network architectures (Socher et al., 2013; Bordes et al., 2013; Bordes et al., 2014), matrix/tensor factorization techniques (Nickel et al., 2011; Riedel et al., 2013; Chang et al., 2014), or Bayesian clustering strategies (Kemp et al., 2006; Xu et al., 2006; Sutskever et al., 2009). Among these methods, TransE (Bordes et al., 2013), which models relations as translating operations, achieves a good trade-off between prediction accuracy and computational efficiency. Various extensions like TransH (Wang et al., 2014) and TransR (Lin et al., 2015b) are later proposed to further enhance the prediction accuracy of TransE. Most existing methods perform the embedding task based solely on triples contained in a KG. Some recent work tries to further incorporate other types of information available, e.g., relation paths (Neelakantan et al., 2015; Lin et al., 2015a; Luo et al., 2015), relation type-constraints (Krompaßet al., 2015), entity types (Guo et al., 2015), and entity descriptions (Zhong et al., 2015), to learn better embeddings.

Logical rules have been widely studied in knowledge acquisition and inference, usually on the basis of Markov logic networks (Richardson and Domingos, 2006; Bröcheler et al., 2010; Pujara et al., 2013; Beltagy and Mooney, 2014). Recently, there has been growing interest in combining logical rules and embedding models. Wang et al. (2015) and Wei et al. (2015) tried to utilize rules to refine predictions

made by embedding models, via integer linear programming or Markov logic networks. In their work, however, rules are modeled separately from embedding models, and will not help obtain better embeddings. Rocktäschel et al. (2015) proposed a joint model that injects first-order logic into embeddings. But their work focuses on relation extraction, creating vector embeddings for entity pairs, and hence fails to discover relations between unpaired entities. This paper, in contrast, aims at learning more predictive embeddings by jointly modeling knowledge and logic. Since each entity has its own embedding, our approach can successfully make predictions between unpaired entities, providing greater flexibility for knowledge acquisition and inference.

3 Jointly Embedding Knowledge and Logic

We first describe the formulation of joint embedding. We are given a KG containing a set of triples $\mathcal{K} = \{(e_i, r_k, e_j)\}$, with each triple composed of two entities $e_i, e_j \in \mathcal{E}$ and their relation $r_k \in \mathcal{R}$. Here $\mathcal{E}$ is the entity vocabulary and $\mathcal{R}$ the relation set. Besides the triples, we are given a set of logical rules $\mathcal{L}$, either specified manually or extracted automatically. A logical rule is encoded, for example, in the form of $\forall x, y : (x, r_s, y) \Rightarrow (x, r_t, y)$, stating that any two entities linked by relation r_s should also be linked by relation r_t. Entities and relations are associated with vector embeddings, denoted by $\mathbf{e}, \mathbf{r} \in \mathbb{R}^d$, representing their latent semantics. The proposed method, KALE, aims to learn these embeddings by jointly modeling knowledge triples $\mathcal{K}$ and logical rules $\mathcal{L}$.

3.1 Overview

To enable joint embedding, a key ingredient of KALE is to unify triples and rules, in terms of first-order logic (Rocktäschel et al., 2014; Rocktäschel et al., 2015). A triple (e_i, r_k, e_j) is taken as a ground atom which applies a relation r_k to a pair of entities e_i and e_j. Given a logical rule, it is first instantiated with concrete entities in the vocabulary $\mathcal{E}$, resulting in a set of ground rules. For example, a universally quantified rule $\forall x, y : (x, \texttt{Capital-Of}, y) \Rightarrow (x, \texttt{Located-In}, y)$ might be instantiated with the concrete entities of `Paris` and `France`, giving the ground rule $(\texttt{Paris}, \texttt{Capital-Of}, \texttt{France}) \Rightarrow (\texttt{Paris}, \texttt{Located-In}, \texttt{France})$.[1] A ground rule can then be interpreted as a complex formula, constructed by combining ground atoms with logical connectives (e.g. $\wedge$ and $\Rightarrow$).

Let $\mathcal{F}$ denote the set of training formulae, both atomic (triples) and complex (ground rules). KALE further employs a truth function $I : \mathcal{F} \rightarrow [0, 1]$ to assign a soft truth value to each formula, indicating how likely a triple holds or to what degree a ground rule is satisfied. The truth value of a triple is determined by the corresponding entity and relation embeddings. The truth value of a ground rule is determined by the truth values of the constituent triples, via specific logical connectives. In this way, KALE models triples and rules in a unified framework. See Figure 1 for an overview. Finally, KALE minimizes a global loss over the training formulae $\mathcal{F}$ to learn entity and relation embeddings compatible with both triples and rules. In what follows, we describe the key components of KALE, including triple modeling, rule modeling, and joint learning.

3.2 Triple Modeling

To model triples we follow TransE (Bordes et al., 2013), as it is simple and efficient while achieving state-of-the-art predictive performance. Specifically, given a triple (e_i, r_k, e_j), we model the relation embedding $\mathbf{r}_k$ as a translation between the entity embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$, i.e., we want $\mathbf{e}_i + \mathbf{r}_k \approx \mathbf{e}_j$ when the triple holds. The intuition here originates from linguistic regularities such as $\texttt{France} - \texttt{Paris} = \texttt{Germany} - \texttt{Berlin}$ (Mikolov et al., 2013). In relational data, such analogy holds because of the certain relation `Capital-Of`, through which we will get $\texttt{Paris} + \texttt{Capital-Of} = \texttt{France}$ and $\texttt{Berlin} + \texttt{Capital-Of} = \texttt{Germany}$. Then, we score each triple on the basis of $\|\mathbf{e}_i + \mathbf{r}_k - \mathbf{e}_j\|_1$, and define its soft truth value as

$$I(e_i, r_k, e_j) = 1 - \frac{1}{3\sqrt{d}} \|\mathbf{e}_i + \mathbf{r}_k - \mathbf{e}_j\|_1, \quad (1)$$

where d is the dimension of the embedding space. It is easy to see that $I(e_i, r_k, e_j) \in [0, 1]$ with the constraints $\|\mathbf{e}_i\|_2 \leq 1$, $\|\mathbf{e}_j\|_2 \leq 1$, and $\|\mathbf{r}_k\|_2 \leq$

[1] Our approach actually takes as input rules represented in first-order logic, i.e., those with quantifiers such as $\forall$. But it could be hard to deal with quantifiers, so we use ground rules, i.e., propositional statements during learning.

1.[2] $I(e_i, r_k, e_j)$ is expected to be large if the triple holds, and small otherwise.

3.3 Rule Modeling

To model rules we use t-norm fuzzy logics (Hájek, 1998), which define the truth value of a complex formula as a composition of the truth values of its constituents, through specific t-norm based logical connectives. We follow Rocktäschel et al. (2015) and use the product t-norm. The compositions associated with logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) are defined as follow:

$$\begin{aligned} I(f_1 \wedge f_2) &= I(f_1) \cdot I(f_2), \\ I(f_1 \vee f_2) &= I(f_1) + I(f_2) - I(f_1) \cdot I(f_2), \\ I(\neg f_1) &= 1 - I(f_1), \end{aligned}$$

where f_1 and f_2 are two constituent formulae, either atomic or complex. Given these compositions, the truth value of any complex formula can be calculated recursively, e.g.,

$$\begin{aligned} I(\neg f_1 \wedge f_2) &= I(f_2) - I(f_1) \cdot I(f_2), \\ I(f_1 \Rightarrow f_2) &= I(f_1) \cdot I(f_2) - I(f_1) + 1. \end{aligned}$$

This paper considers two types of rules. The first type is $\forall x, y : (x, r_s, y) \Rightarrow (x, r_t, y)$. Given a ground rule $f \triangleq (e_m, r_s, e_n) \Rightarrow (e_m, r_t, e_n)$, the truth value is calculated as:

$$\begin{aligned} I(f) = & I(e_m, r_s, e_n) \cdot I(e_m, r_t, e_n) \\ & - I(e_m, r_s, e_n) + 1, \end{aligned} \tag{2}$$

where $I(\cdot,\cdot,\cdot)$ is the truth value of a constituent triple, defined by Eq. (1). The second type is $\forall x, y, z : (x, r_{s_1}, y) \wedge (y, r_{s_2}, z) \Rightarrow (x, r_t, z)$. Given a ground rule $f \triangleq (e_\ell, r_{s_1}, e_m) \wedge (e_m, r_{s_2}, e_n) \Rightarrow (e_\ell, r_t, e_n)$, the truth value is:

$$\begin{aligned} I(f) = & I(e_\ell, r_{s_1}, e_m) \cdot I(e_m, r_{s_2}, e_n) \cdot I(e_\ell, r_t, e_n) \\ & - I(e_\ell, r_{s_1}, e_m) \cdot I(e_m, r_{s_2}, e_n) + 1. \end{aligned} \tag{3}$$

The larger the truth values are, the better the ground rules are satisfied. It is easy to see that besides these two types of rules, the KALE framework is general enough to handle any rules that can be represented as first-order logic formulae. The investigation of other types of rules will be left for future work.

[2]Note that $0 \leq \|\mathbf{e}_i + \mathbf{r}_k - \mathbf{e}_j\|_1 \leq \|\mathbf{e}_i\|_1 + \|\mathbf{r}_k\|_1 + \|\mathbf{e}_j\|_1 \leq 3\sqrt{d}$, where the last inequality holds because $\|\mathbf{x}\|_1 = \sum_i |x_i| \leq \sqrt{d \sum_i x_i^2} = \sqrt{d}\,\|\mathbf{x}\|_2$ for any $\mathbf{x} \in \mathbb{R}^d$, according to the Cauchy-Schwarz inequality.

3.4 Joint Learning

After unifying triples and rules as atomic and complex formulae, we minimize a global loss over this general representation to learn entity and relation embeddings. We first construct a training set $\mathcal{F}$ containing all positive formulae, including (i) observed triples, and (ii) ground rules in which at least one constituent triple is observed. Then we minimize a margin-based ranking loss, enforcing positive formulae to have larger truth values than negative ones:

$$\begin{aligned} \min_{\{\mathbf{e}\},\{\mathbf{r}\}} & \sum_{f^+ \in \mathcal{F}} \sum_{f^- \in \mathcal{N}_{f^+}} \left[\gamma - I(f^+) + I(f^-)\right]_+, \\ \text{s.t.} \quad & \|\mathbf{e}\|_2 \leq 1, \forall e \in \mathcal{E}; \quad \|\mathbf{r}\|_2 \leq 1, \forall r \in \mathcal{R}. \end{aligned} \tag{4}$$

Here $f^+ \in \mathcal{F}$ is a positive formula, $f^- \in \mathcal{N}_{f^+}$ a negative one constructed for f^+, γ a margin separating positive and negative formulae, and $[x]_+ \triangleq \max\{0, x\}$. If $f^+ \triangleq (e_i, r_k, e_j)$ is a triple, we construct f^- by replacing either e_i or e_j with a random entity $e \in \mathcal{E}$, and calculate its truth value according to Eq. (1). For example, we might generate a negative instance $(\texttt{Paris}, \texttt{Capital-Of}, \texttt{Germany})$ for the triple $(\texttt{Paris}, \texttt{Capital-Of}, \texttt{France})$. If $f^+ \triangleq (e_m, r_s, e_n) \Rightarrow (e_m, r_t, e_n)$ or $(e_\ell, r_{s_1}, e_m) \wedge (e_m, r_{s_2}, e_n) \Rightarrow (e_\ell, r_t, e_n)$ is a ground rule, we construct f^- by replacing r_t in the consequent with a random relation $r \in \mathcal{R}$, and calculate its truth value according to Eq. (2) or Eq. (3). For example, given a ground rule $(\texttt{Paris}, \texttt{Capital-Of}, \texttt{France}) \Rightarrow (\texttt{Paris}, \texttt{Located-In}, \texttt{France})$, a possible negative instance $(\texttt{Paris}, \texttt{Capital-Of}, \texttt{France}) \Rightarrow (\texttt{Paris}, \texttt{Has-Spouse}, \texttt{France})$ could be generated. We believe that most instances (both triples and ground rules) generated in this way are truly negative. Stochastic gradient descent in mini-batch mode is used to carry out the minimization. To satisfy the ℓ_2-constraints, $\mathbf{e}$ and $\mathbf{r}$ are projected to the unit ℓ_2-ball before each mini-batch. Embeddings learned in this way are required to be compatible with not only triples but also rules.

3.5 Discussions

Complexity. We compare KALE with several state-of-the-art embedding methods in space complexity and time complexity (per iteration) during learning. Table 1 shows the results, where d is the dimension

Method	Complexity (Space/Time)	
SE (Bordes et al., 2011)	$n_e d+2n_r d^2$	$O(n_t d^2)$
LFM (Jenatton et al., 2012)	$n_e d+n_r d^2$	$O(n_t d^2)$
TransE (Bordes et al., 2013)	$n_e d+n_r d$	$O(n_t d)$
TransH (Wang et al., 2014)	$n_e d+2n_r d$	$O(n_t d)$
TransR (Lin et al., 2015b)	$n_e d+n_r(d^2+d)$	$O(n_t d^2)$
KALE (this paper)	$n_e d+n_r d$	$O(n_t d+n_g d)$

Table 1: Complexity of different embedding methods.

of the embedding space, and $n_e/n_r/n_t/n_g$ is the number of entities/relations/triples/ground rules. The results indicate that incorporating additional rules will not significantly increase the space or time complexity of KALE, keeping the model complexity almost the same as that of TransE (optimal among the methods listed in the table). But please note that KALE needs to ground universally quantified rules before learning, which further requires $O(n_u n_t/n_r)$ in time complexity. Here, n_u is the number of universally quantified rules, and n_t/n_r is the averaged number of observed triples per relation. During grounding, we select those ground rules with at least one triple observed. Grounding is required only once before learning, and is not included during the iterations.

Extensions. Actually, our approach is quite general. (i) Besides TransE, a variety of embedding methods, e.g., those listed in Table 1, can be used for triple modeling (Section 3.2), as long as we further define a mapping $f : \mathbb{R} \to [0, 1]$ to map original scores to soft truth values. (ii) Besides the two types of rules introduced in Section 3.3, other types of rules can also be handled as long as they can be represented as first-order logic formulae. (iii) Besides the product t-norm, other types of t-norm based fuzzy logics can be used for rule modeling (Section 3.3), e.g., the Łukasiewicz t-norm used in probabilistic soft logic (Bröcheler et al., 2010) and the minimum t-norm used in fuzzy description logic (Stoilos et al., 2007). (iv) Besides the pairwise ranking loss, other types of loss functions can be designed for joint learning (Section 3.4), e.g., the pointwise squared loss or the logarithmic loss (Rocktäschel et al., 2014; Rocktäschel et al., 2015).

4 Experiments

We empirically evaluate KALE with two tasks: (i) link prediction and (ii) triple classification.

Dataset	# Ent	# Rel	# Train/Valid/Test-I/Test-II				# Rule
FB122	9,738	122	91,638	9,595	5,057	6,186	78,488
WN18	40,943	18	141,442	5,000	1,394	3,606	119,222

Table 3: Statistics of datasets.

4.1 Experimental Setup

Datasets. We use two datasets: WN18 and FB122. WN18 is a subgraph of WordNet containing 18 relations. FB122 is composed of 122 Freebase relations regarding the topics of "people", "location", and "sports", extracted from FB15K. Both WN18 and FB15K are released by Bordes et al. (2013)[3]. Triples on each dataset are split into training/validation/test sets, used for model training, parameter tuning, and evaluation respectively. For WN18 we use the original data split, and for FB122 we extract triples associated with the 122 relations from the training, validation, and test sets of FB15K.

We further create logical rules for each dataset, in the form of $\forall x, y : (x, r_s, y) \Rightarrow (x, r_t, y)$ or $\forall x, y, z : (x, r_{s_1}, y) \wedge (y, r_{s_2}, z) \Rightarrow (x, r_t, z)$. To do so, we first run TransE to get entity and relation embeddings, and calculate the truth value for each of such rules according to Eq. (2) or Eq. (3). Then we rank all such rules by their truth values and manually filter those ranked at the top. We finally create 47 rules on FB122, and 14 on WN18 (see Table 2 for examples). The rules are then instantiated with concrete entities (grounding). Ground rules in which at least one constituent triple is observed in the *training* set are used in joint learning.

Note that some of the test triples can be inferred by directly applying these rules on the training set (pure logical inference). On each dataset, we further split the test set into two parts, test-I and test-II. The former contains triples that *cannot* be directly inferred by pure logical inference, and the latter the remaining test triples. Table 3 gives some statistics of the datasets, including the number of entities, relations, triples in training/validation/test-I/test-II set, and ground rules.

Comparison settings. As baselines we take the embedding techniques of TransE, TransH, and TransR. TransE models relation embeddings as translation operations between entity embeddings. TransH

[3] https://everest.hds.utc.fr/doku.php?id=en:smemlj12

$\forall x, y$: /sports/athlete/team$(x, y) \Rightarrow$ /sports/sports_team/player(y, x)
$\forall x, y$: /location/country/capital$(x, y) \Rightarrow$ /location/location/contains(x, y)
$\forall x, y, z$: /people/person/nationality$(x, y) \wedge$ /location/country/official_language$(y, z) \Rightarrow$ /people/person/languages(x, z)
$\forall x, y, z$: /country/administrative_divisions$(x, y) \wedge$ /administrative_division/capital$(y, z) \Rightarrow$ /country/second_level_divisions(x, z)
$\forall x, y$: _hypernym$(x, y) \Rightarrow$ _hyponym(y, x)
$\forall x, y$: _instance_hypernym$(x, y) \Rightarrow$ _instance_hyponym(y, x)
$\forall x, y$: _synset_domain_topic_of$(x, y) \Rightarrow$ _member_of_domain_topic(y, x)

Table 2: Examples of rules created.

and TransR are extensions of TransE. They further allow entities to have distinct embeddings when involved in different relations, by introducing relation-specific hyperplanes and projection matrices respectively. All the three methods have been demonstrated to perform well on WordNet and Freebase data.

We further test our approach in three different scenarios. (i) KALE-Trip uses triples alone to perform the embedding task, i.e., only the training triples are included in the optimization Eq. (4). It is a linearly transformed version of TransE. The only difference is that relation embeddings are normalized in KALE-Trip, but not in TransE. (ii) KALE-Pre first repeats pure logical inference on the training set and adds inferred triples as additional training data, until no further triples can be inferred. Both original and inferred triples are then included in the optimization. For example, given a logical rule $\forall x, y : (x, r_s, y) \Rightarrow (x, r_t, y)$, a new triple (e_i, r_t, e_j) can be inferred if (e_i, r_s, e_j) is observed in the training set, and both triples will be used as training instances for embedding. (iii) KALE-Joint is the joint learning scenario, which considers both training triples and ground rules in the optimization. In the aforementioned example, training triple (e_i, r_s, e_j) and ground rule $(e_i, r_s, e_j) \Rightarrow (e_i, r_t, e_j)$ will be used in the training process of KALE-Joint, *without* explicitly incorporating triple (e_i, r_t, e_j). Among the methods, TransE/TransH/TransR and KALE-Trip use only triples, while KALE-Pre/KALE-Joint further incorporates rules, before or during embedding.

Implementation details. We use the code provided by Bordes et al. (2013) for TransE[4], and the code provided by Lin et al. (2015b) for TransH and TransR[5]. KALE is implemented in Java. Note that Lin et al. (2015b) initialized TransR with the results of TransE. However, to ensure fair comparison, we randomly initialize all the methods in our experiments. For all the methods, we create 100 mini-batches on each dataset, and tune the embedding dimension d in $\{20, 50, 100\}$. For TransE, TransH, and TransR which score a triple by a distance in $\mathbb{R}^+$, we tune the learning rate η in $\{0.001, 0.01, 0.1\}$, and the margin γ in $\{1, 2, 3, 4\}$. For KALE which scores a triple (as well as a ground rule) by a soft truth value in the unit interval $[0, 1]$, we set the learning rate η in $\{0.01, 0.02, 0.05, 0.1\}$, and the margin γ in $\{0.1, 0.12, 0.15, 0.2\}$. KALE allows triples and rules to have different weights, with the former fixed to 1, and the latter (denoted by λ) selected in $\{0.001, 0.01, 0.1, 1\}$.

4.2 Link Prediction

This task is to complete a triple (e_i, r_k, e_j) with e_i or e_j missing, i.e., predict e_i given (r_k, e_j) or predict e_j given (e_i, r_k).

Evaluation protocol. We follow the same evaluation protocol used in TransE (Bordes et al., 2013). For each test triple (e_i, r_k, e_j), we replace the head entity e_i by every entity e'_i in the dictionary, and calculate the truth value (or distance) for the corrupted triple (e'_i, r_k, e_j). Ranking the truth values in descending order (or the distances in ascending order), we get the rank of the correct entity e_i. Similarly, we can get another rank by corrupting the tail entity e_j. Aggregated over all the test triples, we report three metrics: (i) the mean reciprocal rank (MRR), (ii) the median of the ranks (MED), and (iii) the proportion of ranks no larger than n (HITS@N). We do not report the averaged rank (i.e., the "Mean Rank" metric used by Bordes et al. (2013)), since it is usually sensitive to outliers (Nickel et al., 2016).

Note that a corrupted triple may exist in KGs, which should also be taken as a valid triple. Consider a test triple (`Paris`, `Located-In`, `France`)

[4]https://github.com/glorotxa/SME

[5]https://github.com/mrlyk423/relation_extraction

		Test-I					Test-II					Test-ALL				
		MRR	MED	HITS@N (%)			MRR	MED	HITS@N (%)			MRR	MED	HITS@N (%)		
				3	5	10			3	5	10			3	5	10
FB122	TransE	0.220	29.0	25.7	32.4	40.6	0.296	5.0	40.0	50.8	57.8	0.262	10.0	33.6	42.5	50.0
	TransH	0.218	29.0	25.0	31.3	39.2	0.297	6.0	37.5	48.5	56.3	0.249	12.0	31.9	40.7	48.6
	TransR	0.219	31.0	24.7	30.8	38.9	0.273	9.0	32.4	42.8	51.6	0.261	15.0	28.9	37.4	45.9
	KALE-Trip	0.201	25.0	23.9	31.6	40.1	0.309	5.0	40.9	51.3	58.0	0.261	11.0	33.3	42.4	50.0
	KALE-Pre	0.203	25.0	24.1	31.7	40.2	**0.368**	**4.0**	**47.3**	**55.4**	**61.4**	**0.294**	**9.0**	**36.9**	**44.8**	**51.9**
	KALE-Joint	**0.229**	**21.0**	**26.3**	**33.8**	**42.2**	0.357	**4.0**	44.0	53.0	59.3	0.299	**9.0**	36.1	44.3	51.6
WN18	TransE	0.248	**4.0**	40.9	60.6	77.0	0.363	3.0	59.4	70.8	81.4	0.331	3.0	54.3	67.9	80.2
	TransH	0.242	**4.0**	39.2	60.1	75.9	0.482	**2.0**	63.5	70.8	79.3	0.415	3.0	56.7	67.8	78.3
	TransR	0.240	**4.0**	40.1	57.7	71.6	0.449	3.0	55.7	64.5	74.3	0.391	3.0	51.3	62.6	73.5
	KALE-Trip	0.250	**4.0**	40.6	62.3	78.1	0.393	**2.0**	61.9	71.2	80.6	0.353	3.0	56.0	68.7	79.9
	KALE-Pre	0.248	**4.0**	40.4	61.5	78.2	0.451	3.0	**69.6**	**77.5**	**85.3**	0.395	3.0	**61.4**	**73.0**	**83.3**
	KALE-Joint	**0.260**	**4.0**	**43.6**	**64.1**	**79.2**	**0.563**	**2.0**	67.6	73.8	81.0	**0.478**	**2.0**	60.9	71.1	80.5

Table 4: Link prediction results on the test-I, test-II, and test-all sets of FB122 and WN18 (raw setting).

		Test-I					Test-II					Test-ALL				
		MRR	MED	HITS@N (%)			MRR	MED	HITS@N (%)			MRR	MED	HITS@N (%)		
				3	5	10			3	5	10			3	5	10
FB122	TransE	0.296	13.0	36.0	41.5	48.1	0.630	2.0	77.5	82.8	88.4	0.480	**2.0**	58.9	64.2	70.2
	TransH	0.280	15.0	33.6	39.1	46.4	0.606	2.0	70.1	75.4	82.0	0.460	3.0	53.7	59.1	66.0
	TransR	0.283	16.0	33.4	39.2	46.0	0.499	2.0	57.0	63.2	70.1	0.401	5.0	46.4	52.4	59.3
	KALE-Trip	0.299	10.0	36.6	42.9	50.2	0.650	2.0	79.0	83.4	88.7	0.492	**2.0**	59.9	65.2	71.4
	KALE-Pre	0.291	11.0	35.8	41.9	49.8	**0.713**	**1.0**	**82.9**	**86.1**	**89.9**	**0.523**	**2.0**	**61.7**	66.2	71.8
	KALE-Joint	**0.325**	**9.0**	**38.4**	**44.7**	**52.2**	0.684	**1.0**	79.7	84.1	89.6	**0.523**	**2.0**	61.2	**66.4**	**72.8**
WN18	TransE	0.306	**3.0**	57.4	72.3	80.1	0.511	2.0	87.5	95.6	98.7	0.453	**2.0**	79.1	89.1	93.6
	TransH	0.318	**3.0**	61.7	72.4	78.2	0.653	2.0	87.1	91.4	94.6	0.560	**2.0**	80.0	86.1	90.0
	TransR	0.299	**3.0**	56.1	66.7	74.5	0.597	2.0	75.0	81.7	88.0	0.514	**2.0**	69.7	77.5	84.3
	KALE-Trip	0.322	**3.0**	61.0	73.9	80.8	0.555	2.0	90.6	96.3	98.8	0.490	**2.0**	82.3	90.1	93.8
	KALE-Pre	0.322	**3.0**	60.6	74.5	81.1	0.612	2.0	**96.4**	**98.6**	**99.6**	0.532	**2.0**	**86.4**	**91.9**	**94.4**
	KALE-Joint	**0.338**	**3.0**	**65.5**	**76.3**	**82.1**	**0.787**	**1.0**	93.3	95.4	97.2	**0.662**	**2.0**	85.5	90.1	93.0

Table 5: Link prediction results on the test-I, test-II, and test-all sets of FB122 and WN18 (filtered setting).

and a possible corruption (`Lyon`, `Located-In`, `France`). Both triples are valid. In this case, ranking `Lyon` before the correct answer `Paris` should not be counted as an error. To avoid such phenomena, we follow Bordes et al. (2013) and remove those corrupted triples which exist in either the training, validation, or test set before getting the ranks. That means, we remove `Lyon` from the candidate list before getting the rank of `Paris` in the aforementioned example. We call the original setting "raw" and the new setting "filtered".

Optimal configurations. For each of the methods to be compared, we tune its hyperparameters in the ranges specified in Section 4.1, and select a best model that leads to the highest filtered MRR score on the validation set (with a total of 500 epochs over the training data). The optimal configurations for KALE are: $d = 100$, $\eta = 0.05$, $\gamma = 0.12$, and $\lambda = 1$ on FB122; $d = 50$, $\eta = 0.05$, $\gamma = 0.2$, and $\lambda = 0.1$ on WN18. To better see and understand the effects of rules, we use the same configuration for KALE-Trip, KALE-Pre, and KALE-Joint on each dataset.

Results. Table 4 and Table 5 show the results in the raw setting and filtered setting respectively. On each dataset we report the metrics on three sets: test-I, test-II, and the whole test set (denoted by test-all). Test-I contains test triples that cannot be directly inferred by performing pure logical inference on the training set, and hence might be intrinsically more difficult for the rules. The remaining test triples (i.e., the directly inferable ones) are included in Test-II. These triples have either been used directly as train-

		Raw						Filtered					
		Test-Incl			Test-Excl			Test-Incl			Test-Excl		
		MEAN	MED	HITS@10	MEAN	MED	HITS@10	MEAN	MED	HITS@10	MEAN	MED	HITS@10
FB122	KALE-Trip	0.150	49.0	34.2	0.235	17.0	44.1	0.267	14.0	46.2	0.321	8.0	52.9
	KALE-Joint	**0.175**	**36.0**	**36.6**	**0.265**	**15.0**	**45.9**	**0.290**	**11.0**	**49.3**	**0.349**	**7.0**	**54.2**
WN18	KALE-Trip	0.062	239.0	15.1	0.285	**4.0**	90.0	0.072	186.0	17.3	0.369	**2.0**	92.9
	KALE-Joint	**0.093**	**186.0**	**19.6**	**0.291**	**4.0**	**90.5**	**0.113**	**136.0**	**24.0**	**0.381**	**2.0**	**93.2**

Table 6: Comparison between KALE-Trip and KALE-Joint on Test-Incl and Test-Excl of FB122 and WN18.

ing instances in KALE-Pre, or encoded explicitly in training ground rules in KALE-Joint, making this set trivial for the rules to some extent. From the results, we can see that in both settings: (i) KALE-Pre and KALE-Joint outperform (or at least perform as well as) the other methods which use triples alone on almost all the test sets, demonstrating the superiority of incorporating logical rules. (ii) On the test-I sets which contain triples beyond the scope of pure logical inference, KALE-Joint performs significantly better than KALE-Pre. On these sets KALE-Joint can still beat all the baselines by a significant margin in most cases, while KALE-Pre can hardly outperform KALE-Trip. It demonstrates the capability of the joint embedding scenario to learn more predictive embeddings, through which we can make better predictions even beyond the scope of pure logical inference. (iii) On the test-II sets which contain directly inferable triples, KALE-Pre can easily beat all the baselines (even KALE-Joint). That means, for triples covered by pure logical inference, it is trivial to improve the performance by directly incorporating them as training instances.

To better understand how the joint embedding scenario can learn more predictive embeddings, on each dataset we further split the test-I set into two parts. Given a triple (e_i, r_k, e_j) in the test-I set, we assign it to the first part if relation r_k is covered by the rules, and the second part otherwise. We call the two parts Test-Incl and Test-Excl respectively. Table 6 compares the performance of KALE-Trip and KALE-Joint on the two parts. The results show that KALE-Joint outperforms KALE-Trip on both parts, but the improvements on Test-Incl are much more significant than those on Test-Excl. Take the filtered setting on WN18 as an example. On Test-Incl, KALE-Joint increases the metric MRR by 55.7%, decreases the metric MED by 26.9%, and increases the metric HITS@10 by 38.2%. On Test-Excl, however, MRR rises by 3.1%, MED remains the same, and HITS@10 rises by only 0.3%. This observation indicates that jointly embedding triples and rules helps to learn more predictive embeddings, especially for those relations that are used to construct the rules. This might be the main reason that KALE-Joint can make better predictions even beyond the scope of pure logical inference.

4.3 Triple Classification

This task is to verify whether an unobserved triple (e_i, r_k, e_j) is correct or not.

Evaluation protocol. We take the following evaluation protocol similar to that used in TransH (Wang et al., 2014). We first create labeled data for evaluation. For each triple in the test or validation set (i.e., a positive triple), we construct 10 negative triples for it by randomly corrupting the entities, 5 at the head position and the other 5 at the tail position.[6] To make the negative triples as difficult as possible, we corrupt a position using only entities that have appeared in that position, and further ensure that the corrupted triples do not exist in either the training, validation, or test set. We simply use the truth values (or distances) to classify triples. Triples with large truth values (or small distances) tend to be predicted as positive. To evaluate, we first rank the triples associated with each specific relation (in descending order according to their truth values, or in ascending order according to the distances), and calculate the average precision for that relation. We then report on the test sets the mean average precision (MAP)

[6] Previous work typically constructs only a single negative case for each positive one. We empirically found such a balanced classification task too simple for our datasets. So we consider a highly unbalanced setting, with a positive-to-negative ratio of 1:10, for which the previously used metric accuracy is no longer suitable.

	FB122			WN18		
	MAP (Test-I/II/ALL)			MAP (Test-I/II/ALL)		
TransE	0.552	0.852	0.634	0.592	0.993	0.958
TransH	0.576	0.758	0.641	0.604	0.978	0.947
TransR	0.572	0.699	0.619	0.412	0.854	0.836
KALE-Trip	0.578	0.829	0.652	0.618	0.995	0.953
KALE-Pre	0.575	**0.916**	0.668	0.620	**0.997**	**0.964**
KALE-Joint	**0.599**	0.870	**0.677**	**0.627**	**0.997**	0.961

Table 7: Triple classification results on the test-I, test-II, and test-all sets of FB122 and WN18.

aggregated over different relations.

Optimal configurations. The hyperparameters of each method are again tuned in the ranges specified in Section 4.1, and the best models are selected by maximizing MAP on the validation set. The optimal configurations for KALE are: $d = 100$, $\eta = 0.1$, $\gamma = 0.2$, and $\lambda = 0.1$ on FB122; $d = 100$, $\eta = 0.1$, $\gamma = 0.2$, and $\lambda = 0.001$ on WN18. Again, we use the same configuration for KALE-Trip, KALE-Pre, and KALE-Joint on each dataset.

Results. Table 7 shows the results on the test-I, test-II, and test-all sets of our datasets. From the results, we can see that: (i) KALE-Pre and KALE-Joint outperform the other methods which use triples alone on almost all the test sets, demonstrating the superiority of incorporating logical rules. (ii) KALE-Joint performs better than KALE-Pre on the test-I sets, i.e., triples that cannot be directly inferred by performing pure logical inference on the training set. This observation is similar to that observed in the link prediction task, demonstrating that the joint embedding scenario can learn more predictive embeddings and make predictions beyond the capability of pure logical inference.

5 Conclusion and Future Work

In this paper, we propose a new method for jointly embedding knowledge graphs and logical rules, referred to as KALE. The key idea is to represent and model triples and rules in a unified framework. Specifically, triples are represented as atomic formulae and modeled by the translation assumption, while rules as complex formulae and by the t-norm fuzzy logics. A global loss on both atomic and complex formulae is then minimized to perform the embedding task. Embeddings learned in this way are compatible not only with triples but also with rules, which are certainly more useful for knowledge acquisition and inference. We evaluate KALE with the link prediction and triple classification tasks on WordNet and Freebase data. Experimental results show that joint embedding brings significant and consistent improvements over state-of-the-art methods. More importantly, it can obtain more predictive embeddings and make better predictions even beyond the scope of pure logical inference.

For future work, we would like to (i) Investigate the efficacy of incorporating other types of logical rules such as $\forall x, y, z : (x, \texttt{Capital-Of}, y) \Rightarrow \neg(x, \texttt{Capital-Of}, z)$. (ii) Investigate the possibility of modeling logical rules using only relation embeddings as suggested by Demeester et al. (2016), e.g., modeling the above rule using only the embedding associated with `Capital-Of`. This avoids grounding, which might be time and space inefficient especially for complicated rules. (iii) Investigate the use of automatically extracted rules which are no longer hard rules and tolerant of uncertainty.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This research is supported by the National Natural Science Foundation of China (grant No. 61402465) and the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDA06030200).

References

Islam Beltagy and Raymond J. Mooney. 2014. Efficient markov logic inference for natural language semantics. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence - Workshop on Statistical Relational Artificial Intelligence*, pages 9–14.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 301–306.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 2787–2795.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

Matthias Bröcheler, Lilyana Mihalkova, and Lise Getoor. 2010. Probabilistic similarity logic. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 73–82.

Kai-wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1579.

Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Regularizing relation representations by first-order implications. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Workshop on Automated Knowledge Base Construction*, pages 75–80.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 84–94.

Petr Hájek. 1998. *The metamathematics of fuzzy logic*. Kluwer.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550.

Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R. Obozinski. 2012. A latent factor model for highly multi-relational data. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pages 3167–3175.

Shangpu Jiang, Daniel Lowd, and Dejing Dou. 2012. Learning to refine an automatically extracted knowledge base using markov logic. In *Proceedings of 12th IEEE International Conference on Data Mining*, pages 912–917.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pages 381–388.

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *Proceedings of the 14th International Semantic Web Conference*, pages 640–655.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015a. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 705–714.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187.

Yuanfei Luo, Quan Wang, Bin Wang, and Li Guo. 2015. Context-dependent knowledge graph embedding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1656–1661.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 156–166.

Maximilian Nickel, Volker Tresp, and Hans P. Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816.

Maximilian Nickel, Volker Tresp, and Hans P. Kriegel. 2012. Factorizing yago: Scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 271–280.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1955–1961.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *Proceed-*

ings of the 12th International Semantic Web Conference, pages 542–557.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference on North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Tim Rocktäschel, Matko Bošnjak, Sameer Singh, and Sebastian Riedel. 2014. Low-dimensional embeddings of logic. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics - Workshop on Semantic Parsing*, pages 45–49.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 926–934.

Giorgos Stoilos, Giorgos B. Stamou, Jeff Z. Pan, Vassilis Tzouvaras, and Ian Horrocks. 2007. Reasoning with very expressive fuzzy description logics. *Journal of Artificial Intelligence Research*, 30:273–320.

Ilya Sutskever, Joshua B. Tenenbaum, and Ruslan R. Salakhutdinov. 2009. Modelling relational data using bayesian clustered tensor factorization. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 1821–1828.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1859–1865.

Evgenia Wasserman-Pritsker, William W. Cohen, and Einat Minkov. 2015. Learning to identify the best contexts for knowledge-based wsd. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1662–1667.

Zhuoyu Wei, Jun Zhao, Kang Liu, Zhenyu Qi, Zhengya Sun, and Guanhua Tian. 2015. Large-scale knowledge base completion: inferring via grounding network sampling over selected instances. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1331–1340.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371.

Zhao Xu, Volker Tresp, Kai Yu, and Hanspeter Kriegel. 2006. Infinite hidden relational models. In *Proceedings of Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 544–551.

Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 267–272.