EMBEDDING ENTITIES AND RELATIONS FOR LEARNING AND INFERENCE IN KNOWLEDGE BASES

Bishan Yang[1]*, **Wen-tau Yih**[2], **Xiaodong He**[2], **Jianfeng Gao**[2] **& Li Deng**[2]
[1]Department of Computer Science, Cornell University, Ithaca, NY, 14850, USA
bishan@cs.cornell.edu
[2]Microsoft Research, Redmond, WA 98052, USA
{scottyih,xiaohe,jfgao,deng}@microsoft.com

ABSTRACT

We consider learning representations of entities and relations in KBs using the neural-embedding approach. We show that most existing models, including NTN (Socher et al., 2013) and TransE (Bordes et al., 2013b), can be generalized under a unified learning framework, where entities are low-dimensional vectors learned from a neural network and relations are bilinear and/or linear mapping functions. Under this framework, we compare a variety of embedding models on the link prediction task. We show that a simple bilinear formulation achieves new state-of-the-art results for the task (achieving a top-10 accuracy of 73.2% vs. 54.7% by TransE on Freebase). Furthermore, we introduce a novel approach that utilizes the learned relation embeddings to mine logical rules such as $BornInCity(a, b) \wedge CityInCountry(b, c) \implies Nationality(a, c)$. We find that embeddings learned from the bilinear objective are particularly good at capturing relational semantics, and that the composition of relations is characterized by matrix multiplication. More interestingly, we demonstrate that our embedding-based rule extraction approach successfully outperforms a state-of-the-art confidence-based rule mining approach in mining Horn rules that involve compositional reasoning.

1 INTRODUCTION

Recent years have witnessed a rapid growth of knowledge bases (KBs) such as Freebase[1], DBPedia (Auer et al., 2007), and YAGO (Suchanek et al., 2007). These KBs store facts about real-world entities (e.g. people, places, and things) in the form of RDF triples[2] (i.e. (*subject*, *predicate*, *object*)). Today's KBs are large in size. For instance, Freebase contains millions of entities and billions of facts (triples) involving a large variety of predicates (relation types). Such large-scale multi-relational data provide an excellent potential for improving a wide range of tasks, from information retrieval, question answering to biological data mining.

Recently, much effort has been invested in relational learning methods that can scale to large knowledge bases. Tensor factorization (e.g. (Nickel et al., 2011; 2012)) and neural-embedding-based models (e.g. (Bordes et al., 2013a;b; Socher et al., 2013)) are two popular kinds of approaches that learn to encode relational information using low-dimensional representations of entities and relations. These representation learning methods have shown good scalability and reasoning ability in terms of validating unseen facts given the existing KB.

In this work, we focus on the study of neural-embedding models, where the representations are learned using neural networks with energy-based objectives. Recent embedding models TransE (Bordes et al., 2013b) and NTN (Socher et al., 2013) have shown state-of-the-art prediction performance compared to tensor factorization methods such as RESCAL (Nickel et al., 2012). They are similar in model forms with slight differences on the choices of entity and relation representations. Without careful comparison, it is not clear how different design choices affect the

*Work conducted while interning at Microsoft Research.

[1] http://freebase.com

[2] http://www.w3.org/TR/rdf11-concepts/

learning results. In addition, the performance of the embedding models are evaluated on the link prediction task (i.e. predicting the correctness of unseen triples). This only indirectly shows the meaningfulness of low-dimensional embeddings. It is hard to explain what relational properties are being captured and to what extent they are captured during the embedding process.

We make three main contributions in this paper. (1) We present a general framework for multi-relational learning that unifies most multi-relational embedding models developed in the past, including NTN (Socher et al., 2013) and TransE (Bordes et al., 2013b). (2) We empirically evaluate different choices of entity representations and relation representations under this framework on the canonical link prediction task and show that a simple bilinear formulation achieves new state-of-the-art results for the task (a top-10 accuracy of 73.2% vs. 54.7% by TransE when evaluated on Freebase). (3) We propose and evaluate a novel approach that utilizes the learned embeddings to mine logical rules such as $BornInCity(a,b) \wedge CityOfCountry(b,c) \implies Nationality(a,c)$. We show that such rules can be effectively extracted by modeling the composition of relation embeddings, and that the embeddings learned from the bilinear objective are particularly good at capturing the compositional semantics of relations via matrix multiplication. Furthermore, we demonstrate that our embedding-based approach outperforms a state-of-the-art rule mining system AMIE (Galárraga et al., 2013) on mining rules that involve compositional reasoning.

The rest of this paper is structured as follows. Section 2 discusses related work. Section 3 presents the general framework for learning multi-relational representations. Sections 4 and 5 present two inference tasks: a canonical link prediction task and a novel rule extraction task where the learned embeddings are empirically evaluated. Section 6 concludes the paper.

2 Related Work

Multi-relational learning has been an active research area for the past couple of years. Traditional statistical learning approaches (Getoor & Taskar, 2007) such as Markov-logic networks (Richardson & Domingos, 2006) usually suffer from scalability issues. More recently, various types of representation learning methods have been proposed to embed multi-relational knowledge into low-dimensional representations of entities and relations, including tensor/matrix factorization (Singh & Gordon, 2008; Nickel et al., 2011; 2012), Bayesian clustering framework (Kemp et al., 2006; Sutskever et al., 2009), and neural networks (Paccanaro & Hinton, 2001; Bordes et al., 2013a;b; Socher et al., 2013). Our work focuses on the study of neural-embedding models as they have shown good scalability and strong generalizability on large-scale KBs.

Existing neural embedding models (Bordes et al., 2013a;b; Socher et al., 2013) all represent entities as low-dimensional vectors and represent relations as operators that combine the representations of two entities. They differ in different parametrization of relation operators. For instance, given two entity vectors, the model of Neural Tensor Network (NTN) (Socher et al., 2013) represents each relation as a bilinear tensor operator followed by a linear matrix operator. The model of TransE (Bordes et al., 2013b), on the other hand, represents each relation as a single vector that linearly interacts with the entity vectors. Likewise, variations on entity representations also exist. Most methods represent each entity as a unit vector while NTN (Socher et al., 2013) represent entities as an average of word vectors and initializing word vectors with pre-trained vectors from external text corpora. There has not been work that closely examines the effectiveness of these different design choices.

Our work on embedding-based rule extraction presented in part of this paper is related to the earlier study on logical inference with learned continuous-space representations. Much existing work along this line focuses on learning logic-based representations for natural language sentences. For example, Socher et al. (2012) builds a neural network that recursively combines word representations based on parse tree structures and shows that such neural network can simulate the behavior of conjunction and negation. Bowman (2014) further demonstrates that recursive neural network can capture certain aspects of natural logical reasoning on examples involving quantifiers like *some* and *all*. Recently, Grefenstette (2013) shows that in theory most aspects of predicate logic can be simulated using tensor calculus. Rocktäschel et al. (2014) further implements the idea by introducing a supervised objective that trains embeddings to be consistent with given logical rules. The evaluation was conducted on toy data and uses limited logical forms. Different from these earlier studies, we propose a novel approach to utilizing embeddings learned without explicit logical constraints to directly mine logical rules from KBs. We demonstrate that the learned embeddings of relations can capture the compositional semantics of relations. Moreover, we systematically evaluate our approach and compare it favorably with a state-of-the-art rule mining approach on the rule extraction task on Freebase.

3 Multi-Relational Representation Learning

In this section, we present a general neural network framework for multi-relational representation learning. We discuss different design choices for the representations of entities and relations which will be empirically compared in Section 4.

Given a KB that is represented as a list of relation triplets (e_1, r, e_2) (denoting e_1 (the *subject*) and e_2 (the *object*) that are in a certain relationship r), we want to learn representations for entities and relations such that valid triplets receive high scores (or low energies). The embeddings can be learned via a neural network. The first layer projects a pair of input entities to low dimensional vectors, and the second layer combines these two vectors to a scalar for comparison via a scoring function with relation-specific parameters.

3.1 Entity Representations

Each input entity corresponds to a high-dimensional vector, either a "one-hot" index vector or a "n-hot" feature vector. Denote by $\mathbf{x}_{e_1}$ and $\mathbf{x}_{e_2}$ the input vectors for entity e_1 and e_2, respectively. Denote by W the first layer projection matrix. The learned entity representations, $\mathbf{y}_{e_1}$ and $\mathbf{y}_{e_2}$ can be written as

$$\mathbf{y}_{e_1} = f\big(\mathbf{W}\mathbf{x}_{e_1}\big), \;\; \mathbf{y}_{e_2} = f\big(\mathbf{W}\mathbf{x}_{e_2}\big)$$

where f can be a linear or non-linear function, and $\mathbf{W}$ is a parameter matrix, which can be randomly initialized or initialized using pre-trained vectors.

Most existing embedding models adopt the "one-hot" input vectors except for NTN (Socher et al., 2013) which represents each entity as an average of its word vectors. This can be viewed as adopting "bag-of-words" vectors as input and learning a projection matrix consisting of word vectors.

3.2 Relation Representations

The choice of relation representations reflects in the form of the scoring function. Most of the existing scoring functions in the literature can be unified based on a basic linear transformation g_r^a, a bilinear transformation g_r^b or their combination, where g_r^a and g_r^b are defined as

$$g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{A}_r^T \left(\begin{array}{c} \mathbf{y}_{e_1} \\ \mathbf{y}_{e_2} \end{array} \right) \quad \text{and} \quad g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{B}_r \mathbf{y}_{e_2}, \tag{1}$$

which $\mathbf{A}_r$ and $\mathbf{B}_r$ are relation-specific parameters.

Models	$\mathbf{B}_r$	$\mathbf{A}_r^T$	Scoring Function
Distance (Bordes et al., 2011)	-	$\big(\mathbf{Q}_{r_1}^T \quad -\mathbf{Q}_{r_2}^T\big)$	$-\lvert\lvert g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})\rvert\rvert_1$
Single Layer (Socher et al., 2013)	-	$\big(\mathbf{Q}_{r1}^T \quad \mathbf{Q}_{r2}^T\big)$	$\mathbf{u}_r^T \tanh(g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}))$
TransE (Bordes et al., 2013b)	$\mathbf{I}$	$\big(\mathbf{V}_r^T \quad -\mathbf{V}_r^T\big)$	$-(2g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) - 2g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) + \lvert\lvert\mathbf{V}_r\rvert\rvert_2^2)$
NTN (Socher et al., 2013)	$\mathbf{T}_r$	$\big(\mathbf{Q}_{r1}^T \quad \mathbf{Q}_{r2}^T\big)$	$\mathbf{u}_r^T \tanh\big(g_r^a(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) + g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2})\big)$

Table 1: Comparisons among several multi-relational models in their scoring functions.

In Table 1, we summarize several popular scoring functions in the literature for a relation triplet (e_1, r, e_2), reformulated in terms of the above two functions. Denote by $\mathbf{y}_{e_1}, \mathbf{y}_{e_2} \in R^n$ two entity vectors. Denote by $\mathbf{Q}_{r_1}, \mathbf{Q}_{r_2} \in R^{n \times m}$ and $\mathbf{V}_r \in R^n$ matrix or vector parameters for linear transformation g_r^a . Denote by $\mathbf{T}_r \in R^{n \times n \times m}$ tensor parameters for bilinear transformation g_r^b. $\mathbf{I} \in R^n$ is an identity matrix. $\mathbf{u}_r \in R^m$ is an additional parameter for relation r. The scoring function for TransE (L2 formulation) is derived from $||\mathbf{y}_{e_1} - \mathbf{y}_{e_2} + V_r||_2^2 = 2V_r^T(\mathbf{y}_{e_1} - \mathbf{y}_{e_2}) - 2\mathbf{y}_{e_1}^T \mathbf{y}_{e_2} + ||V_r||_2^2 + ||\mathbf{y}_{e_1}||_2^2 + ||\mathbf{y}_{e_2}||_2^2$, where $\mathbf{y}_{e_1}$ and $\mathbf{y}_{e_2}$ are unit vectors.

Note that NTN is the most expressive model as it contains both linear and bilinear relation operators as special cases. In terms of the number of parameters, TransE is the simplest model which only parametrizes the linear relation operators with one-dimensional vectors.

In this paper, we also consider the basic bilinear scoring function:

$$g_r^b(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}) = \mathbf{y}_{e_1}^T \mathbf{M}_r \mathbf{y}_{e_2} \tag{2}$$

which is a special case of NTN without the non-linear layer and the linear operator, and uses a 2-d matrix operator $\mathbf{M}_r \in R^{n \times n}$ instead of a tensor operator. Such bilinear formulation has been used in other matrix factorization models such as in (Nickel et al., 2011; Jenatton et al., 2012; García-Durán et al., 2014) with different forms of regularization. Here, we consider a simple way to reduce the number of relation parameters by restricting $\mathbf{M}_r$ to be a diagonal matrix. This results in the same number of relation parameters as TransE. Our experiments in Section 4 demonstrate that this simple formulation enjoys the same scalable property as TransE and it achieves superior performance over TransE and other more expressive models on the task of link prediction.

This general framework for relationship modeling also applies to the recent deep-structured semantic model (Huang et al., 2013; Shen et al., 2014a;b; Gao et al., 2014; Yih et al., 2014), which learns the relevance or a single relation between a pair of word sequences. The framework above applies when using multiple neural network layers to project entities and using a relation-independent scoring function $G_r\big(\mathbf{y}_{e_1}, \mathbf{y}_{e_2}\big) = \cos[\mathbf{y}_{e_1}(\mathbf{W}_r), \mathbf{y}_{e_2}(\mathbf{W}_r)]$. The cosine scoring function is a special case of g_r^b with normalized $\mathbf{y}_{e_1}, \mathbf{y}_{e_2}$ and with $\mathbf{B}_r = \mathbf{I}$.

3.3 Parameter Learning

The neural network parameters of all the models discussed above can be learned by minimizing a margin-based ranking objective , which encourages the scores of positive relationships (triplets) to be higher than the scores of any negative relationships (triplets). Usually only positive triplets are observed in the data. Given a set of positive triplets T, we can construct a set of "negative" triplets T' by corrupting either one of the relation arguments, $T' = \{(e_1', r, e_2) | e_1' \in E, (e_1', r, e_2) \notin T\} \cup \{(e_1, r, e_2') | e_2' \in E, (e_1, r, e_2') \notin T\}$. Denote the scoring function for triplet (e_1, r, e_2) as $S_{(e_1, r, e_2)}$. The training objective is to minimize the margin-based ranking loss

$$L(\Omega) = \sum_{(e_1, r, e_2) \in T} \sum_{(e_1', r, e_2') \in T'} \max\{S_{(e_1', r, e_2')} - S_{(e_1, r, e_2)} + 1, 0\} \tag{3}$$

4 Inference Task I: Link Prediction

We first conduct a comparison study of different embedding models on the canonical link prediction task, which is to predict the correctness of unseen triplets. As in (Bordes et al., 2013b), we formulate link prediction as an entity ranking task. For each triplet in the test data, we treat each entity as the target entity to be predicted in turn. Scores are computed for the correct entity and all the corrupted entities in the dictionary and are ranked in descending order. We consider *Mean Reciprocal Rank (MRR)* (an average of the reciprocal rank of an answered entity over all test triplets), *HITS@10* (top-10 accuracy), and *Mean Average Precision* (MAP) (as used in (Chang et al., 2014)) as the evaluation metrics.

We examine five embedding models in decreasing order of complexity: (1) NTN with 4 tensor slices as in (Socher et al., 2013); (2) Bilinear+Linear, NTN with 1 tensor slice and without the non-linear layer; (3) TransE, a special case of Bilinear+Linear (see Table 1); (4) Bilinear: using scoring function in Eq. (2); (5) Bilinear-diag: a special case of Bilinear where the relation matrix is a diagonal matrix.

Datasets We used the WordNet (WN) and Freebase (FB15k) datasets introduced in (Bordes et al., 2013b). WN contains $151,442$ triplets with $40,943$ entities and 18 relations, and FB15k consists of $592,213$ triplets with $14,951$ entities and 1345 relations. We use the same training/validation/test split as in (Bordes et al., 2013b). We also consider a subset of FB15k (FB15k-401) containing only frequent relations (relations with at least 100 training examples). This results in $560,209$ triplets with $14,541$ entities and 401 relations.

Implementation details All the models were implemented in C# and using GPU. Training was implemented using mini-batch stochastic gradient descent with AdaGrad (Duchi et al., 2011). At each gradient step, we sampled for each positive triplet two negative triplets, one with a corrupted subject entity and one with a corrupted object entity. The entity vectors are renormalized to have unit length after each gradient step (it is an effective technique that empirically improved all the models). For the relation parameters, we used standard L2 regularization. For all models, we set the number of mini-batches to 10, the dimensionality of the entity vector $d = 100$, the regularization parameter 0.0001, and the number of training epochs $T = 100$ on FB15k and FB15k-401 and $T = 300$ on WN (T was determined based on the learning curves where the performance of all models plateaued.) The learning rate was initially set to 0.1 and then adapted during training by AdaGrad.

4.1 Results

	FB15k		FB15k-401		WN	
	MRR	HITS@10	MRR	HITS@10	MRR	HITS@10
NTN	0.25	41.4	0.24	40.5	0.53	66.1
Blinear+Linear	0.30	49.0	0.30	49.4	0.87	91.6
TransE (DistAdd)	0.32	53.9	0.32	54.7	0.38	90.9
Bilinear	0.31	51.9	0.32	52.2	**0.89**	92.8
Bilinear-diag (DistMult)	**0.35**	**57.7**	**0.36**	**58.5**	0.83	**94.2**

Table 2: Performance comparisons among different embedding models

Table 2 shows the results of all compared methods on all the datasets. In general, we observe that the performance increases as the complexity of the model decreases on FB. NTN, the most complex model, provides the worst performance on both FB and WN, which suggests overfitting. Compared to the previously published results of TransE (Bordes et al., 2013b), our implementation achieves much better results (53.9% vs. 47.1% on FB15k and 90.9% vs. 89.2% on WN) using the same evaluation metric (HITS@10). We attribute such discrepancy mainly to the different choice of SGD optimization: AdaGrad vs. constant learning rate. We also found that Bilinear consistently provides comparable or better performance than TransE, especially on WN. Note that WN contains much more entities than FB, it may require the parametrization of relations to be more expressive to better handle the richness of entities. Interestingly, we found that a simple variant of Bilinear – Bilinear-diag, clearly outperforms all baselines on FB and achieves comparable performance to Bilinear on WN. Note that Bilinear-diag has the limitation of encoding the difference between a relation and its inverse. Still, as there is a large variety of relations in FB and the average number of training examples seen by each relation is relatively small (compared to WN), the simple form of Bilinear-diag is able to provide good prediction performance.

Multiplicative vs. Additive Interactions Note that Bilinear-diag and TransE have the same number of model parameters and their difference can be viewed as the operational choices of the composition of two entity vectors – Bilinear-diag uses weighted element-wise dot product (multiplicative operation) and TransE uses element-wise subtraction with a bias (additive operation). To highlight the difference, here we use DistMult and DistAdd to refer to Bilinear-diag and TransE, respectively. Comparisons between these two models can provide us more insights on the effect of two common choices of compositional operations – multiplication and addition for modeling entity relations. Overall, we observed superior performance of DistMult on all the datasets in Table 2. Table 3 shows the *HITS@10* score on four types of relation categories (as defined in (Bordes et al., 2013b)) on FB15k-401 when predicting the subject entity and the object entity respectively. We can see that DistMult significantly outperforms DistAdd in almost all the categories.

	Predicting subject entities				Predicting object entities			
	1-to-1	1-to-n	n-to-1	n-to-n	1-to-1	1-to-n	n-to-1	n-to-n
DistADD	70.0	76.7	21.1	53.9	68.7	17.4	**83.2**	57.5
DistMult	**75.5**	**85.1**	**42.9**	**55.2**	**73.7**	**46.7**	81.0	**58.8**

Table 3: Results by relation categories: one-to-one, one-to-many, many-to-one and many-to-many

Initialization of Entity Vectors In the following, we examine the learning of entity representations and introduce two further improvements: using non-linear projection and initializing entity vectors with pre-trained vectors. We focus on DistMult as our baseline and compare it with the two modifications DistMult-tanh (using $f = \tanh$ for entity projection) and DistMult-tanh-EV-init (initializing the entity parameters with the 1000-dimensional pre-trained entity vectors released by *word2vec* (Mikolov et al., 2013)) on FB15k-401. We also reimplemented the initialization technique introduced in (Socher et al., 2013) – each entity is represented as an average of its word vectors and the word vectors are initialized using the 300-dimensional pre-trained word vectors released by *word2vec*. We denote this method as DISTMULT-tanh-WV-init. Inspired by (Chang et al., 2014), we design a new evaluation setting where the predicted entities are automatically filtered according to "entity types" (entities that appear as the subjects/objects of a relation have the same type defined by that relation). This provides us with better understanding of the model performance when some entity type information is provided.

	MRR	HITS@10	MAP (w/ type checking)
DISTMULT	0.36	58.5	64.5
DISTMULT-tanh	0.39	63.3	76.0
DISTMULT-tanh-WV-init	0.28	52.5	65.5
DISTMULT-tanh-EV-init	**0.42**	**73.2**	**88.2**

Table 4: Evaluation with pre-trained vectors

In Table 4, we can see that DISTMULT-tanh-EV-init provides the best performance on all the metrics. Surprisingly, we observed performance drops by DISTMULT-tanh-WV-init. We suspect that this is because word vectors are not appropriate for modeling entities described by non-compositional phrases (more than 73% of the entities in FB15k-401 are person names, locations, organizations and films). The promising performance of DISTMULT-tanh-EV-init suggests that the embedding model can greatly benefit from pre-trained entity-level vectors using external textual resources.

5 INFERENCE TASK II: RULE EXTRACTION

In this section, we focus on a complementary inference task, where we utilize the learned embeddings to extract logical rules from the KB. For example, given the fact that a person was born in New York and New York is a city of the United States, then the person's nationality is the United States:

$$BornInCity(a,b) \wedge CityOfCountry(b,c) \implies Nationality(a,c)$$

Such logical rules can serve four important purposes. First, they can help deduce new facts and complete the existing KBs. Second, they can help optimize data storage by storing only rules instead of large amounts of extensional data, and generate facts only at inference time. Third, they can support complex reasoning. Lastly, they can provide explanations for inference results, e.g. we may infer that people's professions usually involve the specialization of the field they study, etc.

The key problem of extracting Horn rules like the aforementioned example is how to effectively explore the search space. Traditional rule mining approaches directly operate on the KB graph – they search for possible rules (i.e. closed-paths in the graph) by pruning rules with low statistical significance and relevance (Schoenmackers et al., 2010). These approaches often fail on large KB graphs due to scalability issues. In the following, we introduce a novel embedding-based rule mining approach whose efficiency is not affected by the size of the KB graph but rather by the number of distinct types of relations in the KB (which is usually relatively small). It can also mine better rules due to its strong generalizability.

5.1 BACKGROUND AND NOTATIONS

For a better illustration, we adopt the graph view of KB. Each binary relation $r(a,b)$ is a directed edge from node a to node b and with link type r. We are interested in extracting Horn rules that consist of a **head** relation H and a sequence of **body** relations $B_1, ..., B_n$:

$$B_1(a_1,a_2) \wedge B_2(a_2,a_3) \wedge ... \wedge B_n(a_n,a_{n+1}) \implies H(a_1,a_{n+1}) \quad (4)$$

where a_i are variables that can be substituted by entities. We constrain the body relations $B_1, ..., B_n$ to form a directed *path* in the graph and the head relation H to from a directed edge that *close* the path (from the start of the path to the end of the path). We denote such property as the **closed-path** property. For consecutive relations that share one variable but do not form a path, e,g, $B_{i-1}(a, b) \wedge B_i(a, c)$, we can replace one of the relations with its inverse relation, so that the relations are connected by an object and an subject, e.g. $B_{i-1}^{-1}(b, a) \wedge B_i(a, c)$. We are interested in mining rules that reflect relationships among different relation types, therefore we also constrain $B_1, ..., B_n, H$ to have distinct relation types. A rule is **instantiated** when all variables are substituted by entities. We denote the **length** of the rule as the number of body relations. In general longer rules are harder to extract due to the exponential search space. In our experiments, we focus on extracting rules of length 2 and 3.

In KBs, entities usually have types and relations often can only take arguments of certain types. For example, *BornInCity* relation can only take a *person* as the subject and a *location* as the object. For each relation r, we denote the domain of its subject argument (the set of entities that can appear in the subject position) as $\mathcal{X}_r$ and similarly the domain of its object argument as $\mathcal{Y}_r$. Such domain information can be extremely useful in restricting the search space of logical rules.

5.2 EMBEDDING-BASED RULE EXTRACTION

For simplicity, we consider Horn rules of length 2 (longer rules can be easily derived from this case):

$$B_1(a, b) \wedge B_2(b, c) \implies H(a, c) \tag{5}$$

Note that the body of the rule can be viewed as the composition of relations B_1 and B_2, which is a new relation that has the property that entities a and c are in a relation if and only if there is an entity b which simultaneously satisfies two relations $B_1(a, b)$ and $B_2(b, c)$.

We model relation composition as multiplication or addition of two relation embeddings. Here we focus on relation embeddings that are in the form of vectors (as in TRANSE) and matrices (as in BILINEAR and its variants). The composition results in a new embedding that lies in the same relation space. Specifically, we use addition for embeddings learned from TRANSE and multiplication for embeddings learned from BILINEAR, inspired by the two different model forms: TRANSE optimizes $\mathbf{y}_a + \mathbf{V}_1 \sim \mathbf{y}_b$ and $\mathbf{y}_b + \mathbf{V}_2 \sim \mathbf{y}_c$, which can indicate that $\mathbf{y}_a + (\mathbf{V}_1 + \mathbf{V}_2) \sim \mathbf{y}_c$; BILINEAR optimizes $\mathbf{y}_a^T \mathbf{M}_1 \mathbf{y}_b$ and $\mathbf{y}_b^T \mathbf{M}_2 \mathbf{y}_c$ to have high scores, which can imply that $\mathbf{y}_a^T(\mathbf{M}_1\mathbf{M}_2)\mathbf{y}_c$ has a high score. This is because for any unit vectors $\mathbf{y}_a, \mathbf{y}_b, \mathbf{y}_c$, $(\mathbf{y}_a^T \mathbf{M}_1 \mathbf{y}_b)(\mathbf{y}_b^T \mathbf{M}_2 \mathbf{y}_c) \leqslant \mathbf{y}_a^T(\mathbf{M}_1\mathbf{M}_2)\mathbf{y}_c$ (see Appendix A for derivations).

To simulate the implication in 5, we want the composition result of relation B_1 and B_2 to demonstrate similar behavior to the embedding of relation H. We assume the similarity between relation embeddings is related to the Euclidean distance if the embeddings are vectors and to the Frobenius norm if the embeddings are matrices. This distance metric allows us to rank possible pairs of relations with respect to the relevance of their composition to the target relation.

Note that we do not need to enumerate all possible pairs of relations in the KB. For example, if the relation in the head is r, then we are only interested in relation pairs (p, q) that satisfy the type constraints, namely: (1) $\mathcal{Y}_p \cap \mathcal{X}_q \neq \varnothing$; (2) $\mathcal{X}_p \cap \mathcal{X}_r \neq \varnothing$; (3) $\mathcal{Y}_q \cap \mathcal{Y}_r \neq \varnothing$. As mentioned before, the arguments (entities) of relations are usually strongly typed in KBs. Applying such domain constraints can effectively reduce the search space.

Algorithm 1 EMBEDRULE

```
1: Input: $KB = \{(e_1, r, e_2)\}$, relation set $R$
2: Output: Candidate rules $Q$
3: for each $r$ in $R$ do
4:     Select the set of start relations $S = \{s : \mathcal{X}_s \cap \mathcal{X}_r \neq \varnothing\}$
5:     Select the set of end relations $T = \{t : \mathcal{Y}_t \cap \mathcal{Y}_r \neq \varnothing\}$
6:     Find all possible relation sequences
7:     Select the $K$-NN sequences $P' \subseteq P$ for $r$ based on $dist(\mathbf{M}_r, \mathbf{M}_{p_1} \circ \cdots \circ \mathbf{M}_{p_n})$
8:     Form candidate rules using $P'$ where $r$ is the head relation and $p \in P'$ is the body in a rule
9:     Add the candidate rules into $Q$
10: end for
```

In Algorithm 1, we describe our rule extraction algorithm for general closed-path Horn rules as in Eq. (4). In Step 7, $\circ$ denotes vector addition or matrix multiplication. We apply a global threshold value δ in our experiments to filter candidate sequences for each relation r, and then automatically select the top remaining sequences by applying a heuristic thresholding strategy based on the difference of the distance scores: sort the sequences by increasing distance $d_1, ..., d_T$ and set the cut-off point to be the j-th sequence where $j = \arg\max_i (d_{i+1} - d_i)$.

5.3 EXPERIMENTS

We evaluate our rule extraction method (denoted as EMBEDRULE) on the FB15k-401 dataset. In our experiments, we remove the equivalence relations and relations whose domains have cardinality 1 since rules involving these relations are not interesting. This results in training data that contains 485,741 facts, 14,417 entities, and 373 relations. Our EMBEDRULE algorithm identifies 60,020 possible length-2 relation sequences and 2,156,391 possible length-3 relation sequences. We then apply the thresholding method described in Section 5.2 to further select top ~3.9K length-2 rules and ~2K length-3 rules [3]. By default all the extracted rules are ranked by decreasing confidence, which is computed as the ratio of the correct predictions to the total number of predictions, where predictions are triplets that are derived from the instantiated rules where the body relations are observed.

We implemented four versions of EMBEDRULE using embeddings trained from TRANSE (DISTADD), BILINEAR, BILINEAR-DIAG (DISTMULT) and DISTMULT-tanh-EV-init with corresponding composition functions. We also compare our approaches to AMIE (Galárraga et al., 2013), a state-of-the-art rule mining system that can efficiently search for Horn rules in large-scale KBs by using novel measurements of support and confidence. The system extracts *close* rules – a superset of the rules we consider in this paper: every relation in the body is connected to the following relation by sharing an entity variable, and every entity variable in the rule appears *at least* twice. We run AMIE with the default setting on the same training set. It extracts 2,201 possible length-1 rules and 46,975 possible length-2 rules, among which 3,952 rules have the *close-path* property. We compare these length-2 rules with the similar number of length-2 rules extracted by EMBEDRULE. By default AMIE ranks rules by PCA confidence (a normalized confidence that takes into account the incompleteness of KBs). However we found that ranking by the standard confidence gives better performance than the PCA confidence on the Freebase dataset we use.

For computational cost, $EmbedRule$ mines length-2 rules in 2 minutes and mines length-3 rules in 20 minutes (the computational time is similar when using different types of embeddings). AMIE mines rules of length $\leqslant 2$ in 9 minutes. All methods are evaluated on a machine with a 64-bit processor, 2 CPUs and 8GB memory.

We consider precision as the evaluation metric, which is the ratio of predictions that are in the test (unseen) data to all the generated unseen predictions. Note that this is an estimation, since a prediction is not necessarily "incorrect" if it is not seen. Galárraga et al. (2013) suggested to identify incorrect predictions based on the functional property of relations. However, we find that most relations in our datasets are not functional. For a better estimation, we manually labeled the top 30 unseen facts predicted by each method by checking Wikipedia. We also remove rules where the head relations are hard to justified due to dynamic factors (i.e. involving the word "current").

5.4 RESULTS

Figure 1 compares the predictions generated by the length-2 rules extracted by different methods. We plot the aggregated precision of the top rules that produce up to $10K$ predictions in total. From left to right, the n-th data point represents the total number of predictions of the top n rules and the estimated precision of these predictions. We can see that EMBEDRULE that uses embeddings trained from the bilinear objective (BILINEAR, DISTMULT and DISTMULT-TANH-EV-INIT) consistently outperforms AMIE. This suggests that the bilinear embeddings contain good amount of information about relations which allows for effective rule selection without looking at the entities. For example, AMIE fails to extract $TVProgramCountryofOrigin(a, b) \wedge CountryOfficialLanguage(b, c) \implies TVProgramLanguage(a, c)$ by relying on the instan-

[3] We consider K=100 nearest-neighbor sequences for each method, and set δ to 9.2, 36.3, 1.9 and 3.4 for DISTMULT-TANH-EV-INIT, DISTMULT, BILINEAR and DISTADD respectively for length-2 rules, and set it to 9.1, 48.8, 2.9, and 1.1 for lengh-3 rules.

tiations of the rule occurred in the observed KB while all the bilinear variants of EMBEDRULE successfully extract the rule purely based on the embeddings of the three involved relations.

We can also see that in general, using multiplicative composition of matrix embeddings (from DISTMULT and BILINEAR) results in better performance compared to using additive composition of vector embeddings (from DISTADD). We found many examples where DISTADD fails to retrieve rules because it assigns large distance between the composition of the body relations and the head relation in the embedding space while its multiplicative counterpart DISTMULT ranks the composition result much closer to the head relation. For example, DISTADD prunes the possible composition $FilmDistributorInRegion \wedge RegionGDPCurrency$ for relation $FilmBudgetCurrency$ while DISTMULT ranks the composition as the nearest neighbor of $FilmBudgetCurrency$.

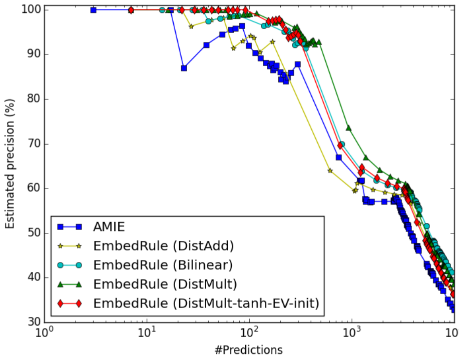

Figure 1: Aggregated precision of top length-2 rules extracted by different methods

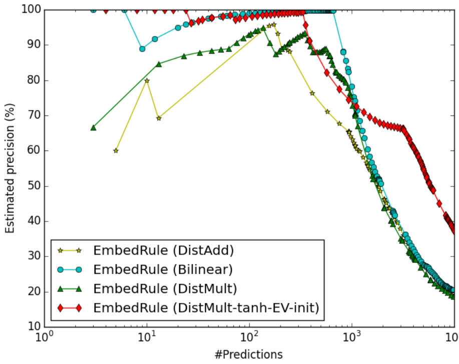

Figure 2: Aggregated precision of top length-3 rules extracted by different methods

We also look at the results for length-3 rules generated by different implementations of EMBEDRULE in Figure 2. We can see that the initial length-3 rules extracted by EMBEDRULE can provide very good precision in general. We can also see that BILINEAR consistently outperforms DISTMULT and DISTADD on the top 1K predictions and DISTMULT-TANH-EV-INIT tends to outperform the other methods as more predictions are generated. We think that the fact that BILINEAR starts to show more advantage over DISTMULT in extracting longer rules confirm the limitation of representing relations by diagonal matrices, as longer rules requires the modeling of more complex relation semantics.

6 CONCLUSION

In this paper, we present a general framework for learning representations of entities and relations in KBs. Under the framework, we empirically evaluate different embedding models on knowledge inference tasks. We show that a simple formulation of bilinear model can outperform the state-of-the-art embedding models for link prediction on Freebase. Furthermore, we examine the learned embeddings by utilizing them to extract logical rules from KBs. We show that embeddings learned from the bilinear objective can capture compositional semantics of relations and be successfully used to extract Horn rules that involve compositional reasoning. For future work, we aim to exploit deep structure in the neural network framework. As learning representations using deep networks has shown great success in various applications (Hinton et al., 2012; Vinyals et al., 2012; Deng et al., 2013), it may also help capturing hierarchical structure hidden in the multi-relational data. Further, tensor constructs have been usefully applied to some deep learning architectures (Yu et al., 2013; Hutchinson et al., 2013). Related constructs and architectures may help improve multi-relational learning and inference.

APPENDIX

A DETAILED DERIVATIONS

Proposition 1. *For any unit vectors $a, b, c \in R^n$ and square matrices $M_1, M_2 \in R^{n\times n}$, $a^T M_1 bb^T M_2 c \leqslant a^T M_1 M_2 c$.*

Proof. Let $A = bb^T$. A is a normal matrix, because $A^T A = AA^T$. Therefore, there exists a diagonal matrix Λ and a unitary matrix U such that $A = U\Lambda U^T$, where the diagonal entries of Λ are the eigenvalues of A and the columns of U are the eigenvectors of A. Let the eigenvalues of A be $\lambda_1, ..., \lambda_n$. We can show that $\forall i, 0 \leqslant \lambda_i \leqslant 1$. This is because A is positive semi-definite and $||A||_2 = \sqrt{\sum_{i,j}(b_i b_j)^2} = \sqrt{\sum_i \lambda_i^2} = 1$.

Therefore, we have $a^T M_1 bb^T M_2 c = a^T M_1 U\Lambda U^T M_2 c \leqslant a^T M_1 UU^T M_2 c = a^T M_1 M_2 c$. □

B EXAMPLES OF THE EXTRACTED HORN RULES

Examples of length-2 rules extracted by EMBEDRULE with embeddings learned from DISTMULT-tanh-EV-init:

$$AwardInCeremany(a,b) \wedge CeremanyEventType(b,c) \implies AwardInEventType(a,c)$$

$$AtheletePlayInTeam(a,b) \wedge TeamPlaySport(b,c) \implies AtheletePlaySport(a,c)$$

$$TVProgramInTVNetwork(a,b) \wedge TVNetworkServiceLanguage(b,c) \implies TVProgramLanguage(a,c)$$

$$LocationInState(a,b) \wedge StateInCountry(b,c) \implies LocationInCountry(a,c)$$

$$BornInLocation(a,b) \wedge LocationInCountry(b,c) \implies Nationality(a,c)$$

Examples of length-3 rules extracted by EMBEDRULE with embeddings learned from DISTMULT-tanh-EV-init:

$$SportPlayByTeam(a,b) \wedge TeamInClub(b,c) \wedge ClubHasPlayer(c,d) \implies SportPlayByAthelete(a,d)$$

$$MusicTrackPerformer(a,b) \wedge PeerInfluence(b,c) \wedge PerformRole(c,d) \implies MusicTrackRole(a,d)$$

$$FilmHasActor(a,b) \wedge CelebrityFriendship(b,c) \wedge PersonLanguage(c,d) \implies FilmLanguage(a,d)$$

C VISUALIZATION OF RELATION EMBEDDINGS

Visualization of the relation embeddings learned by DISTMULT and DISTADD using t-SNE (see figure 3 and 4). We selected 189 relations in the FB15k-401 dataset. The embeddings learned by DISTMULT nicely reflect the clustering structures among these relations (e.g. /film/release_region is closed to /film/country); whereas the embeddings learned by DISTADD present structure that is harder to interpret.

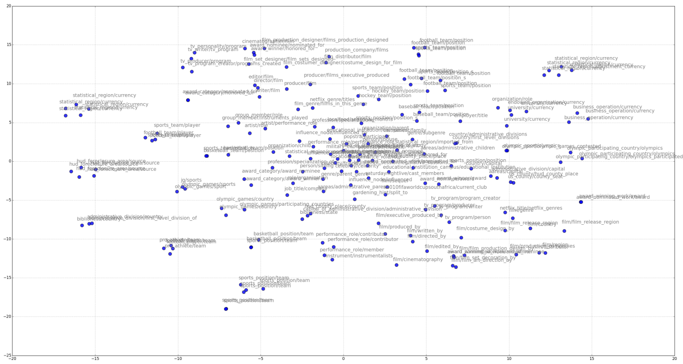

Figure 3: Relation embeddings (DISTADD)

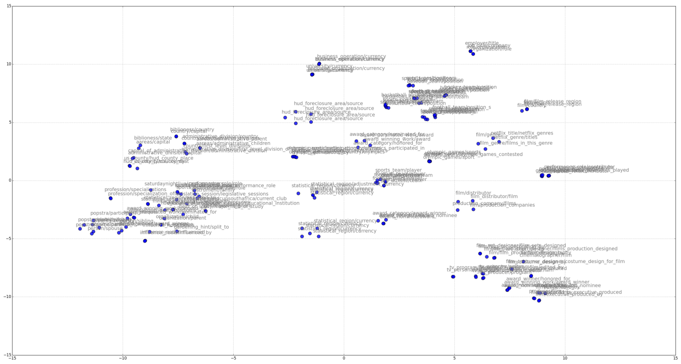

Figure 4: Relation embeddings (DISTMULT)

REFERENCES

Auer, Sören, Bizer, Christian, Kobilarov, Georgi, Lehmann, Jens, Cyganiak, Richard, and Ives, Zachary. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pp. 722–735. Springer, 2007.

Bordes, Antoine, Weston, Jason, Collobert, Ronan, and Bengio, Yoshua. Learning structured embeddings of knowledge bases. In *AAAI*, 2011.

Bordes, Antoine, Glorot, Xavier, Weston, Jason, and Bengio, Yoshua. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, pp. 1–27, 2013a.

Bordes, Antoine, Usunier, Nicolas, Garcia-Duran, Alberto, Weston, Jason, and Yakhnenko, Oksana. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013b.

Bowman, Samuel R. Can recursive neural tensor networks learn logical reasoning? In *ICLR*, 2014.

Chang, Kai-Wei, Yih, Wen-tau, Yang, Bishan, and Meek, Chris. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*, 2014.

Deng, Li, Hinton, G., and Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: An overview. In *in ICASSP*, 2013.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Galárraga, Luis Antonio, Teflioudi, Christina, Hose, Katja, and Suchanek, Fabian. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, 2013.

Gao, Jianfeng, Pantel, Patrick, Gamon, Michael, He, Xiaodong, Deng, Li, and Shen, Yelong. Modeling interestingness with deep neural networks. In *EMNLP*, 2014.

García-Durán, Alberto, Bordes, Antoine, and Usunier, Nicolas. Effective blending of two and three-way interactions for modeling multi-relational data. In *Machine Learning and Knowledge Discovery in Databases*, pp. 434–449. Springer, 2014.

Getoor, Lise and Taskar, Ben (eds.). *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

Grefenstette, Edward. Towards a formal distributional semantics: Simulating logical calculi with tensors. In **SEM*, 2013.

Hinton, Geoff, Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition. *IEEE Sig. Proc. Mag.*, 29:82–97, 2012.

Huang, Po-Sen, He, Xiaodong, Gao, Jianfeng, Deng, Li, Acero, Alex, and Heck, Larry. Learning deep structured semantic models for Web search using clickthrough data. In *CIKM*, 2013.

Hutchinson, B, Deng, L., and Yu, D. Tensor deep stacking networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1944–1957, 2013.

Jenatton, Rodolphe, Le Roux, Nicolas, Bordes, Antoine, and Obozinski, Guillaume. A latent factor model for highly multi-relational data. In *NIPS*, 2012.

Kemp, Charles, Tenenbaum, Joshua B, Griffiths, Thomas L, Yamada, Takeshi, and Ueda, Naonori. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, pp. 5, 2006.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In *NIPS*, pp. 3111–3119, 2013.

Nickel, Maximilian, Tresp, Volker, and Kriegel, Hans-Peter. A three-way model for collective learning on multi-relational data. In *ICML*, pp. 809–816, 2011.

Nickel, Maximilian, Tresp, Volker, and Kriegel, Hans-Peter. Factorizing YAGO: scalable machine learning for linked data. In *WWW*, pp. 271–280, 2012.

Paccanaro, Alberto and Hinton, Geoffrey E. Learning distributed representations of concepts using linear relational embedding. *IEEE Transactions on Knowledge and Data Engineering*, 13(2): 232–244, 2001.

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

Rocktäschel, Tim, Bošnjak, Matko, Singh, Sameer, and Riedel, Sebastian. Low-dimensional embeddings of logic. In *ACL Workshop on Semantic Parsing*, 2014.

Schoenmackers, Stefan, Etzioni, Oren, Weld, Daniel S, and Davis, Jesse. Learning first-order horn clauses from web text. In *EMNLP*, 2010.

Shen, Yelong, He, Xiaodong, Gao, Jianfeng, Deng, Li, and Mesnil, Gregoire. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*, 2014a.

Shen, Yelong, He, Xiaodong, Gao, Jianfeng, Deng, Li, and Mesnil, Grégoire. Learning semantic representations using convolutional neural networks for Web search. In *WWW*, pp. 373–374, 2014b.

Singh, Ajit P and Gordon, Geoffrey J. Relational learning via collective matrix factorization. In *KDD*, pp. 650–658. ACM, 2008.

Socher, Richard, Huval, Brody, Manning, Christopher D., and Ng, Andrew Y. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP-CoNLL*, 2012.

Socher, Richard, Chen, Danqi, Manning, Christopher D., and Ng, Andrew Y. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, 2013.

Suchanek, Fabian M, Kasneci, Gjergji, and Weikum, Gerhard. Yago: a core of semantic knowledge. In *WWW*, 2007.

Sutskever, Ilya, Tenenbaum, Joshua B, and Salakhutdinov, Ruslan. Modelling relational data using Bayesian clustered tensor factorization. In *NIPS*, pp. 1821–1828, 2009.

Vinyals, O., Jia, Y., Deng, L., and Darrell, T. Learning with recursive perceptual representations. In *NIPS*, 2012.

Yih, Wen-tau, He, Xiaodong, and Meek, Christopher. Semantic parsing for single-relation question answering. In *ACL*, 2014.

Yu, D., Deng, L., and Seide, F. The deep tensor neural network with applications to large vocabulary speech recognition. *IEEE Trans. Audio, Speech and Language Proc.*, 21(2):388 –396, 2013.