Neighborhood Mixture Model for Knowledge Base Completion

Dat Quoc Nguyen[1], **Kairit Sirts**[1], **Lizhen Qu**[2] and **Mark Johnson**[1]

[1] Department of Computing, Macquarie University, Sydney, Australia
dat.nguyen@students.mq.edu.au, {kairit.sirts, mark.johnson}@mq.edu.au
[2] Data61 & Australian National University
lizhen.qu@data61.csiro.au

Abstract

Knowledge bases are useful resources for many natural language processing tasks, however, they are far from complete. In this paper, we define a novel entity representation as a mixture of its neighborhood in the knowledge base and apply this technique on TransE—a well-known embedding model for knowledge base completion. Experimental results show that the neighborhood information significantly helps to improve the results of the TransE, leading to better performance than obtained by other state-of-the-art embedding models on three benchmark datasets for triple classification, entity prediction and relation prediction tasks.

Keywords: Knowledge base completion, embedding model, mixture model, link prediction, triple classification, entity prediction, relation prediction.

1 Introduction

Knowledge bases (KBs), such as WordNet (Miller, 1995), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and DBpedia (Lehmann et al., 2015), represent relationships between entities as triples $(\text{head entity}, \text{relation}, \text{tail entity})$. Even very large knowledge bases are still far from complete (Socher et al., 2013; West et al., 2014). *Knowledge base completion* or *link prediction* systems (Nickel et al., 2015) predict which triples not in a knowledge base are likely to be true (Taskar et al., 2004; Bordes et al., 2011).

Embedding models for KB completion associate entities and/or relations with dense feature vectors or matrices. Such models obtain state-of-the-art performance (Bordes et al., 2012; Bordes et al., 2013; Socher et al., 2013; Wang et al., 2014; Guu et al., 2015; Nguyen et al., 2016) and generalize to large KBs (Krompa et al., 2015).

Most embedding models for KB completion learn only from triples and by doing so, ignore lots of information implicitly provided by the structure of the knowledge graph. Recently, several authors have addressed this issue by incorporating relation path information into model learning (García-Durán et al., 2015; Lin et al., 2015a; Guu et al., 2015; Toutanova et al., 2016) and have shown that the relation paths between entities in KBs provide useful information and improve knowledge base completion. For instance, a three-relation path

$$\begin{aligned}&(\text{head}, \text{born\_in\_hospital}/\text{r}_1, \text{e}_1)\\ \Rightarrow&(\text{e}_1, \text{hospital\_located\_in\_city}/\text{r}_2, \text{e}_2)\\ \Rightarrow&(\text{e}_2, \text{city\_in\_country}/\text{r}_3, \text{tail})\end{aligned}$$

is likely to indicate that the fact $(\text{head}, \text{nationality}, \text{tail})$ could be true, so the relation path here $\text{p} = \{\text{r}_1, \text{r}_2, \text{r}_3\}$ is useful for predicting the relationship "nationality" between the head and tail entities.

Besides the relation paths, there could be other useful information implicitly presented in the knowledge base that could be exploited for better KB completion. For instance, the whole neighborhood of entities could provide lots of useful information for predicting the relationship between two entities. Consider for example a KB fragment given in Figure 1. If we know that Ben Affleck has won an Oscar award and Ben Affleck lives in Los Angeles, then this can help us to predict that Ben Affleck is an actor or a film maker, rather than a lecturer or a doctor. If we additionally know that Ben Affleck's gender is male then there is a higher chance for him to be a film maker. This intuition can be formalized by representing an entity vector as a relation-specific mixture of its neighborhood as follows:

Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pages 40–50,
Berlin, Germany, August 7-12, 2016.

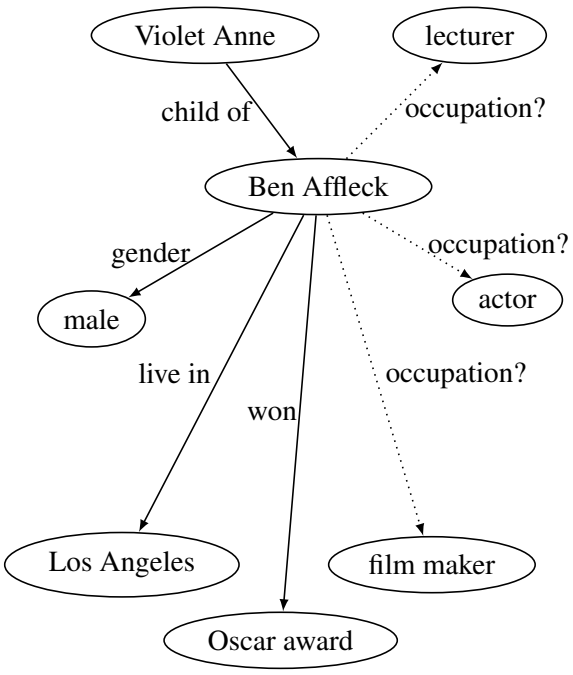

Figure 1: An example fragment of a KB.

$$\begin{aligned}\text{Ben\_Affleck} = &\ \omega_{r,1}(\text{Violet\_Anne}, \text{child\_of}) \\ &+ \omega_{r,2}(\text{male}, \text{gender}^{-1}) \\ &+ \omega_{r,3}(\text{Los\_Angeles}, \text{lives\_in}^{-1}) \\ &+ \omega_{r,4}(\text{Oscar\_award}, \text{won}^{-1}),\end{aligned}$$

where $\omega_{r,i}$ are the mixing weights that indicate how important each neighboring relation is for predicting the relation r. For example, for predicting the occupation relationship, the knowledge about the child_of relationship might not be that informative and thus the corresponding mixing coefficient can be close to zero, whereas it could be relevant for predicting some other relationship, such as parent or spouse, in which case the relation-specific mixing coefficient for the child_of relationship could be high.

The primary contribution of this paper is introducing and formalizing the neighborhood mixture model. We demonstrate its usefulness by applying it to the well-known TransE model (Bordes et al., 2013). However, it could be applied to other embedding models as well, such as Bilinear models (Bordes et al., 2012; Yang et al., 2015) and STransE (Nguyen et al., 2016). While relation path models exploit extra information using longer paths existing in the KB, the neighborhood mixture model effectively incorporates information about many paths simultaneously. Our extensive experiments on three benchmark datasets show that it achieves superior performance over competitive baselines in three KB completion tasks: triple classification, entity prediction and relation prediction.

2 Neighborhood mixture modeling

In this section, we start by explaining how to formally construct the neighbor-based entity representations in section 2.1, and then describe the **N**eighborhood **M**ixture **M**odel applied to the TransE model (Bordes et al., 2013) in section 2.2. Section 2.3 explains how we train our model.

2.1 Neighbor-based entity representation

Let $\mathcal{E}$ denote the set of entities and $\mathcal{R}$ the set of relation types. Denote by $\mathcal{R}^{-1}$ the set of inverse relations r^{-1}. Denote by $\mathcal{G}$ the knowledge graph consisting of a set of correct tiples (h, r, t), such that $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. Let $\mathcal{K}$ denote the symmetric closure of $\mathcal{G}$, i.e. if a triple $(h, r, t) \in \mathcal{G}$, then both (h, r, t) and $(t, r^{-1}, h) \in \mathcal{K}$.

Define:

$$\mathcal{N}_{e,r} = \{e' | (e', r, e) \in \mathcal{K}\}$$

as a set of neighboring entities connected to entity e with relation r. Then

$$\mathcal{N}_e = \{(e', r) | r \in \mathcal{R} \cup \mathcal{R}^{-1}, e' \in \mathcal{N}_{e,r}\}$$

is the set of all entity and relation pairs that are neighbors for entity e.

Each entity e is associated with a k-dimensional vector $\boldsymbol{v}_e \in \mathbb{R}^k$ and relation-dependent vectors $\boldsymbol{u}_{e,r} \in \mathbb{R}^k, r \in \mathcal{R} \cup \mathcal{R}^{-1}$. Now we can define the neighborhood-based entity representation $\boldsymbol{\vartheta}_{e,r}$ for an entity $e \in \mathcal{E}$ for predicting the relation $r \in \mathcal{R}$ as follows:

$$\boldsymbol{\vartheta}_{e,r} = a_e \boldsymbol{v}_e + \sum_{(e',r') \in \mathcal{N}_e} b_{r,r'} \boldsymbol{u}_{e',r'}, \qquad (1)$$

a_e and $b_{r,r'}$ are the mixture weights that are constrained to sum to 1 for each neighborhood:

$$a_e \propto \delta + \exp \alpha_e \qquad (2)$$

$$b_{r,r'} \propto \exp \beta_{r,r'} \qquad (3)$$

where $\delta \geqslant 0$ is a hyper-parameter that controls the contribution of the entity vector $\boldsymbol{v}_e$ to the neighbor-based mixture, α_e and $\beta_{r,r'}$ are the learnable exponential mixture parameters.

In real-world factual KBs, e.g. Freebase (Bollacker et al., 2008), some entities, such as "male", can have thousands or millions neighboring entities sharing the same relation "gender." For such entities, computing the neighbor-based vectors can be computationally very expensive. To overcome this problem, we introduce in our implementation a filtering threshold τ and consider in the neighbor-based entity representation construction only those relation-specific neighboring entity sets for which $|\mathcal{N}_{e,r}| \leq \tau$.

2.2 TransE-NMM: applying neighborhood mixtures to TransE

Embedding models define for each triple $(h, r, t) \in \mathcal{G}$, a *score function* $f(h, r, t)$ that measures its implausibility. The goal is to choose f such that the score $f(h, r, t)$ of a plausible triple (h, r, t) is smaller than the score $f(h', r', t')$ of an implausible triple (h', r', t').

TransE (Bordes et al., 2013) is a simple embedding model for knowledge base completion, which, despite of its simplicity, obtains very competitive results (García-Durán et al., 2016; Nickel et al., 2016). In TransE, both entities e and relations r are represented with k-dimensional vectors $\boldsymbol{v}_e \in \mathbb{R}^k$ and $\boldsymbol{v}_r \in \mathbb{R}^k$, respectively. These vectors are chosen such that for each triple $(h, r, t) \in \mathcal{G}$:

$$\boldsymbol{v}_h + \boldsymbol{v}_r \approx \boldsymbol{v}_t \tag{4}$$

The score function of the TransE model is the norm of this translation:

$$f(h, r, t)_{\text{TransE}} = \|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\|_{\ell_{1/2}} \tag{5}$$

We define the score function of our new model TransE-NMM in terms of the neighbor-based entity vectors as follows:

$$f(h, r, t) = \|\boldsymbol{\vartheta}_{h,r} + \boldsymbol{v}_r - \boldsymbol{\vartheta}_{t,r^{-1}}\|_{\ell_{1/2}}, \tag{6}$$

using either the ℓ_1 or the ℓ_2-norm, and $\boldsymbol{\vartheta}_{h,r}$ and $\boldsymbol{\vartheta}_{t,r^{-1}}$ are defined following the Equation 1. The relation-specific entity vectors $\boldsymbol{u}_{e,r}$ used to construct the neighbor-based entity vectors $\boldsymbol{\vartheta}_{e,r}$ are defined based on the TransE translation operator:

$$\boldsymbol{u}_{e,r} = \boldsymbol{v}_e + \boldsymbol{v}_r \tag{7}$$

in which $\boldsymbol{v}_{r^{-1}} = -\boldsymbol{v}_r$. For each correct triple (h, r, t), the sets of neighboring entities $\mathcal{N}_{h,r}$ and $\mathcal{N}_{t,r^{-1}}$ exclude the entities t and h, respectively.

If we set the filtering threshold $\tau = 0$ then $\boldsymbol{\vartheta}_{h,r} = \boldsymbol{v}_h$ and $\boldsymbol{\vartheta}_{t,r^{-1}} = \boldsymbol{v}_t$ for all triples. In this case, TransE-NMM reduces to the plain TransE model. In all our experiments presented in section 4, the baseline TransE results are obtained with the TransE-NMM with $\tau = 0$.

2.3 Parameter optimization

The TransE-NMM model parameters include the vectors $\boldsymbol{v}_e, \boldsymbol{v}_r$ for entities and relation types, the entity-specific weights $\boldsymbol{\alpha} = \{\alpha_e | e \in \mathcal{E}\}$ and relation-specific weights $\boldsymbol{\beta} = \{\beta_{r,r'} | r, r' \in \mathcal{R} \cup \mathcal{R}^{-1}\}$. To learn these parameters, we minimize the L_2-regularized margin-based objective function:

$$\mathcal{L} = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'_{(h,r,t)}}} [\gamma + f(h, r, t) - f(h', r, t')]_+ + \frac{\lambda}{2}\Big(\|\boldsymbol{\alpha}\|_2^2 + \|\boldsymbol{\beta}\|_2^2\Big), \tag{8}$$

where $[x]_+ = \max(0, x)$, γ is the margin hyper-parameter, λ is the L_2 regularization parameter and

$$\mathcal{G}'_{(h,r,t)} = \{(h', r, t) \mid h' \in \mathcal{E}, (h', r, t) \notin \mathcal{G}\} \cup \{(h, r, t') \mid t' \in \mathcal{E}, (h, r, t') \notin \mathcal{G}\}$$

is the set of incorrect triples generated by corrupting the correct triple $(h, r, t) \in \mathcal{G}$. We applied the "*Bernoulli*" trick to choose whether to generate the head or tail entity when sampling an incorrect triple (Wang et al., 2014; Lin et al., 2015b; He et al., 2015; Ji et al., 2015; Ji et al., 2016).

We use Stochastic Gradient Descent (SGD) with RMSProp adaptive learning rate to minimize $\mathcal{L}$, and impose the following hard constraints during training: $\|\boldsymbol{v}_e\|_2 \leqslant 1$ and $\|\boldsymbol{v}_r\|_2 \leqslant 1$. We employ alternating optimization to minimize $\mathcal{L}$. We first initialize the entity and relation-specific mixing parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to zero and only learn the randomly initialized entity and relation vectors $\boldsymbol{v}_e$ and $\boldsymbol{v}_r$. Then we fix the learned vectors and only optimize the mixing parameters. In the final step, we fix again the mixing parameters and fine-tune the vectors. In all experiments presented in section 4, we train for 200 epochs during each three optimization step.

3 Related work

Table 1 summarizes related embedding models for link prediction and KB completion. The models

Model	Score function $f(h, r, t)$	**Opt.**
STransE	$\|\mathbf{W}_{r,1}\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_{r,2}\boldsymbol{v}_t\|_{\ell_{1/2}}$; $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$; $\boldsymbol{v}_r \in \mathbb{R}^k$	SGD
SE	$\|\mathbf{W}_{r,1}\boldsymbol{v}_h - \mathbf{W}_{r,2}\boldsymbol{v}_t\|_{\ell_{1/2}}$; $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$	SGD
Unstructured	$\|\boldsymbol{v}_h - \boldsymbol{v}_t\|_{\ell_{1/2}}$	SGD
TransE	$\|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\|_{\ell_{1/2}}$; $\boldsymbol{v}_r \in \mathbb{R}^k$	SGD
TransH	$\|(\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ $\boldsymbol{r}_p, \boldsymbol{v}_r \in \mathbb{R}^k$; $\mathbf{I}$: Identity matrix size $k \times k$	SGD
TransD	$\|(\mathbf{I} + \boldsymbol{r}_p\boldsymbol{h}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} + \boldsymbol{r}_p\boldsymbol{t}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ $\boldsymbol{r}_p, \boldsymbol{v}_r \in \mathbb{R}^n$; $\boldsymbol{h}_p, \boldsymbol{t}_p \in \mathbb{R}^k$; $\mathbf{I}$: Identity matrix size $n \times k$	AdaDelta
TransR	$\|\mathbf{W}_r\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_r\boldsymbol{v}_t\|_{\ell_{1/2}}$; $\mathbf{W}_r \in \mathbb{R}^{n\times k}$; $\boldsymbol{v}_r \in \mathbb{R}^n$	SGD
TranSparse	$\|\mathbf{W}_r^h(\theta_r^h)\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_r^t(\theta_r^t)\boldsymbol{v}_t\|_{\ell_{1/2}}$; $\mathbf{W}_r^h, \mathbf{W}_r^t \in \mathbb{R}^{n\times k}$; $\theta_r^h, \theta_r^t \in \mathbb{R}$; $\boldsymbol{v}_r \in \mathbb{R}^n$	SGD
SME	$(\mathbf{W}_{1,1}\boldsymbol{v}_h + \mathbf{W}_{1,2}\boldsymbol{v}_r + \mathbf{b}_1)^\top(\mathbf{W}_{2,1}\boldsymbol{v}_t + \mathbf{W}_{2,2}\boldsymbol{v}_r + \mathbf{b}_2)$ $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^n$; $\mathbf{W}_{1,1}, \mathbf{W}_{1,2}, \mathbf{W}_{2,1}, \mathbf{W}_{2,2} \in \mathbb{R}^{n\times k}$	SGD
DISTMULT	$\boldsymbol{v}_h^\top\mathbf{W}_r\boldsymbol{v}_t$; $\mathbf{W}_r$ is a diagonal matrix $\in \mathbb{R}^{k\times k}$	AdaGrad
NTN	$\boldsymbol{v}_r^\top tanh(\boldsymbol{v}_h^\top\mathbf{M}_r\boldsymbol{v}_t + \mathbf{W}_{r,1}\boldsymbol{v}_h + \mathbf{W}_{r,2}\boldsymbol{v}_t + \mathbf{b}_r)$ $\boldsymbol{v}_r, \mathbf{b}_r \in \mathbb{R}^n$; $\mathbf{M}_r \in \mathbb{R}^{k\times k\times n}$; $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{n\times k}$	L-BFGS
Bilinear-COMP	$\boldsymbol{v}_h^\top\mathbf{W}_{r_1}\mathbf{W}_{r_2}...\mathbf{W}_{r_m}\boldsymbol{v}_t$; $\mathbf{W}_{r_1}, \mathbf{W}_{r_2}, ..., \mathbf{W}_{r_m} \in \mathbb{R}^{k\times k}$	AdaGrad
TransE-COMP	$\|\boldsymbol{v}_h + \boldsymbol{v}_{r_1} + \boldsymbol{v}_{r_2} + ... + \boldsymbol{v}_{r_m} - \boldsymbol{v}_t\|_{\ell_{1/2}}$; $\boldsymbol{v}_{r_1}, \boldsymbol{v}_{r_2}, ..., \boldsymbol{v}_{r_m} \in \mathbb{R}^k$	AdaGrad

Table 1: The score functions $f(h, r, t)$ and the optimization methods (Opt.) of several prominent embedding models for KB completion. In all of these models, the entities h and t are represented by vectors $\boldsymbol{v}_h$ and $\boldsymbol{v}_t \in \mathbb{R}^k$ respectively.

differ in their score function $f(h, r, t)$ and the algorithm used to optimize their margin-based objective function, e.g., SGD, AdaGrad (Duchi et al., 2011), AdaDelta (Zeiler, 2012) or L-BFGS (Liu and Nocedal, 1989).

The *Unstructured* model (Bordes et al., 2012) assumes that the head and tail entity vectors are similar. As the Unstructured model does not take the relationship into account, it cannot distinguish different relation types. The *Structured Embedding* (SE) model (Bordes et al., 2011) extends the Unstructured model by assuming that the head and tail entities are similar only in a relation-dependent subspace, where each relation is represented by two different matrices. Futhermore, the SME model (Bordes et al., 2012) uses four different matrices to project entity and relation vectors into a subspace. The TransH model (Wang et al., 2014) associates each relation with a relation-specific hyperplane and uses a projection vector to project entity vectors onto that hyperplane. TransD (Ji et al., 2015) and TransR/CTransR (Lin et al., 2015b) extend the TransH model by using two projection vectors and a matrix to project entity vectors into a relation-specific space, respectively. STransE (Nguyen et al., 2016) and TranSparse (Ji et al., 2016) are extensions of the TransR model, where head and tail entities are associated with their own projection matrices.

The DISTMULT model (Yang et al., 2015) is based on the *Bilinear* model (Nickel et al., 2011; Bordes et al., 2012; Jenatton et al., 2012) where each relation is represented by a diagonal rather than a full matrix. The neural tensor network (NTN) model (Socher et al., 2013) uses a bilinear tensor operator to represent each relation. Similar quadratic forms are used to model entities and relations in KG2E (He et al., 2015) and TATEC (García-Durán et al., 2016).

Recently, Neelakantan et al. (2015), Gardner and Mitchell (2015), Luo et al. (2015), Lin et al. (2015a), García-Durán et al. (2015), Guu et al. (2015) and Toutanova et al. (2016) showed that relation paths between entities in KBs provide richer information and improve the relationship prediction. In fact, our new TransE-NMM model can be also viewed as a three-relation path model as it takes into account the neighborhood entity and relation information of both head and tail entities in each triple.

Luo et al. (2015) constructed relation paths between entities and viewing entities and relations in the path as pseudo-words applied Word2Vec algorithms (Mikolov et al., 2013) to produce pre-trained vectors for these pseudo-words. Luo et al. (2015) showed that using these pre-trained vectors for initialization helps to improve the performance of the TransE, SME and SE models. RTransE (García-Durán et al., 2015), PTransE (Lin et al., 2015a) and TransE-COMP (Guu et al., 2015) are extensions of the TransE model. These models similarly represent a relation path by a vector which is the sum of the vectors of all relations in the path, whereas in the Bilinear-COMP model (Guu et al., 2015), each relation is a matrix and so it represents the relation path by matrix multiplication. Our neighborhood mixture model can be adapted to both relation path models Bilinear-COMP and TransE-COMP, by replacing head and tail entity vectors by the neighbor-based vector representations, thus combining advantages of both path and neighborhood information. Nickel et al. (2015) reviews other approaches for learning from KBs and multi-relational data.

4 Experiments

To investigate the usefulness of the neighbor mixtures, we compare the performance of the TransE-NMM against the results of the baseline TransE and other state-of-the-art embedding models on the triple classification, entity prediction and relation prediction tasks.

4.1 Datasets

Dataset:	WN11	FB13	NELL186
#R	11	13	186
#E	38,696	75,043	14,463
#Train	112,581	316,232	31,134
#Valid	2,609	5,908	5,000
#Test	10,544	23,733	5,000

Table 2: Statistics of the experimental datasets used in this study (and *previous works*). #E is the number of entities, #R is the number of relation types, and #Train, #Valid and #Test are the numbers of correct triples in the training, validation and test sets, respectively. Each validation and test set also contains the same number of incorrect triples as the number of correct triples.

We conduct experiments using three publicly available datasets WN11, FB13 and NELL186. For all of them, the validation and test sets containing both correct and incorrect triples have already been constructed. Statistical information about these datasets is given in Table 2.

The two benchmark datasets[1], WN11 and FB13, were produced by Socher et al. (2013) for triple classification. WN11 is derived from the large lexical KB WordNet (Miller, 1995) involving 11 relation types. FB13 is derived from the large real-world fact KB FreeBase (Bollacker et al., 2008) covering 13 relation types. The NELL186 dataset[2] was introduced by Guo et al. (2015) for both triple classification and entity prediction tasks, containing 186 most frequent relations in the KB of the CMU Never Ending Language Learning project (Carlson et al., 2010).

4.2 Evaluation tasks

We evaluate our model on three commonly used benchmark tasks: triple classification, entity prediction and relation prediction. This subsection describes those tasks in detail.

Triple classification: The triple classification task was first introduced by Socher et al. (2013), and since then it has been used to evaluate various embedding models. The aim of the task is to predict whether a triple (h, r, t) is correct or not.

For classification, we set a relation-specific threshold θ_r for each relation type r. If the implausibility score of an unseen test triple (h, r, t) is smaller than θ_r then the triple will be classified as correct, otherwise incorrect. Following Socher et al. (2013), the relation-specific thresholds are determined by maximizing the micro-averaged accuracy, which is a per-triple average, on the validation set. We also report the macro-averaged accuracy, which is a per-relation average.

Entity prediction: The entity prediction task (Bordes et al., 2013) predicts the head or the tail entity given the relation type and the other entity, i.e. predicting h given $(?, r, t)$ or predicting t given $(h, r, ?)$ where ? denotes the missing element. The results are evaluated using a ranking induced by the function $f(h, r, t)$ on test triples. Note that the incorrect triples in the validation and test sets are not used for evaluating the entity prediction task nor the relation prediction task.

[1] http://cs.stanford.edu/people/danqi/data/nips13-dataset.tar.bz2

[2] http://aclweb.org/anthology/attachments/P/P15/P15-1009.Datasets.zip

Each correct test triple (h, r, t) is corrupted by replacing either its head or tail entity by each of the possible entities in turn, and then we rank these candidates in ascending order of their implausibility score. This is called as the "Raw" setting protocol. For the "Filtered" setting protocol described in Bordes et al. (2013), we also filter out before ranking any corrupted triples that appear in the KB. Ranking a corrupted triple appearing in the KB (i.e. a correct triple) higher than the original test triple is also correct, but is penalized by the "Raw" score, thus the "Filtered" setting provides a clearer view on the ranking performance.

In addition to the mean rank and the Hits@10 (i.e., the proportion of test triples for which the target entity was ranked in the top 10 predictions), which were originally used in the entity prediction task (Bordes et al., 2013), we also report the mean reciprocal rank (**MRR**), which is commonly used in information retrieval. In both "Raw" and "Filtered" settings, mean rank is always greater or equal to 1 and lower mean rank indicates better entity prediction performance. The MRR and Hits@10 scores always range from 0.0 to 1.0, and higher score reflects better prediction result.

Relation prediction: The relation prediction task (Lin et al., 2015a) predicts the relation type given the head and tail entities, i.e. predicting r given $(h, ?, t)$ where ? denotes the missing element. We corrupt each correct test triple (h, r, t) by replacing its relation r by each possible relation type in turn, and then rank these candidates in ascending order of their implausibility score. Just as in the entity prediction task, we use two setting protocols, "Raw" and "Filtered", and evaluate on mean rank, MRR and Hits@10.

4.3 Hyper-parameter tuning

For all evaluation tasks, results for TransE are obtained with TransE-NMM with the filtering threshold $\tau = 0$, while we set $\tau = 10$ for TransE-NMM.

For triple classification, we first performed a grid search to choose the optimal hyper-parameters for TransE by monitoring the micro-averaged triple classification accuracy after each training epoch on the validation set. For all datasets, we chose either the ℓ_1 or ℓ_2 norm in the score function f and the initial RMSProp learning rate $\eta \in \{0.001, 0.01\}$. Following the previous work (Wang et al., 2014; Lin et al., 2015b; Ji et al., 2015; He et al., 2015; Ji et al., 2016), we selected the margin hyper-parameter $\gamma \in \{1, 2, 4\}$ and the number of vector dimensions $k \in \{20, 50, 100\}$ on WN11 and FB13. On NELL186, we set $\gamma = 1$ and $k = 50$ (Guo et al., 2015; Luo et al., 2015). The highest accuracy on the validation set was obtained when using $\eta = 0.01$ for all three datasets, and when using ℓ_2 norm for NELL186, $\gamma = 4$, $k = 20$ and ℓ_1 norm for WN11, and $\gamma = 1$, $k = 100$ and ℓ_2 norm for FB13.

We set the hyper-parameters η, γ, k, and the ℓ_1 or the ℓ_2-norm in our TransE-NMM model to the same optimal hyper-parameters searched for TransE. We then used a grid search to select the hyper-parameter $\delta \in \{0, 1, 5, 10\}$ and L_2 regularizer $\lambda \in \{0.005, 0.01, 0.05\}$ for TransE-NMM. By monitoring the micro-averaged accuracy after each training epoch, we obtained the highest accuracy on validation set when using $\delta = 1$ and $\lambda = 0.05$ for both WN11 and FB13, and $\delta = 0$ and $\lambda = 0.01$ for NELL186.

For both entity prediction and relation prediction tasks, we set the hyper-parameters η, γ, k, and the ℓ_1 or the ℓ_2-norm for both TransE and TransE-NMM to be the same as the optimal parameters found for the triple classification task. We then monitored on TransE the filtered MRR on validation set after each training epoch. We chose the model with highest validation MRR, which was then used to evaluate the test set. For TransE-NMM, we searched the hyper-parameter $\delta \in \{0, 1, 5, 10\}$ and L_2 regularizer $\lambda \in \{0.005, 0.01, 0.05\}$. By monitoring the filtered MRR after each training epoch, we selected the best model with the highest filtered MRR on the validation set. Specifically, for the entity prediction task, we selected $\delta = 10$ and $\lambda = 0.005$ for WN11, $\delta = 5$ and $\lambda = 0.01$ for FB13, and $\delta = 5$ and $\lambda = 0.005$ for NELL186. For the relation prediction task, we selected $\delta = 10$ and $\lambda = 0.005$ for WN11, $\delta = 10$ and $\lambda = 0.05$ for FB13, and $\delta = 1$ and $\lambda = 0.05$ for NELL186.

5 Results

5.1 Quantitative results

Table 3 presents the results of TransE and TransE-NMM on triple classification, entity prediction and relation prediction tasks on all experimental datasets. The results show that TransE-NMM generally performs better than TransE in all three evaluation tasks.

Specifically, TransE-NMM obtains higher triple

Data	Method		Triple class.		Entity prediction			Relation prediction		
			Mic.	Mac.	MR	MRR	H@10	MR	MRR	H@10
WN11	R	TransE	85.21	82.53	4324	**0.102**	**19.21**	2.37	0.679	**99.93**
		TransE-NMM	**86.82**	**84.37**	**3687**	0.094	17.98	**2.14**	**0.687**	99.92
	F	TransE			4304	**0.122**	**21.86**	2.37	0.679	**99.93**
		TransE-NMM			**3668**	0.109	20.12	**2.14**	**0.687**	99.92
FB13	R	TransE	87.57	86.66	9037	0.204	35.39	1.01	0.996	99.99
		TransE-NMM	**88.58**	**87.99**	**8289**	**0.258**	**35.53**	1.01	0.996	**100.0**
	F	TransE			5600	0.213	36.28	1.01	0.996	99.99
		TransE-NMM			**5018**	**0.267**	**36.36**	1.01	0.996	**100.0**
NELL186	R	TransE	92.13	88.96	309	0.192	36.55	8.43	0.580	77.18
		TransE-NMM	**94.57**	**90.95**	**238**	**0.221**	**37.55**	**6.15**	**0.677**	**82.16**
	F	TransE			279	0.268	47.13	8.32	0.602	77.26
		TransE-NMM			**214**	**0.292**	**47.82**	**6.08**	**0.690**	**82.20**

Table 3: Experimental results of TransE (i.e. TransE-NMM with $\tau = 0$) and TransE-NMM with $\tau = 10$. Micro-averaged (labeled as **Mic.**) and Macro-averaged (labeled as **Mac.**) accuracy results are for the triple classification task. MR, MRR and H@10 abbreviate the mean rank, the mean reciprocal rank and Hits@10 (in %), respectively. "R" and "F" denote the "Raw" and "Filtered" settings used in the entity prediction and relation prediction tasks, respectively.

Method	W11	F13
TransR (Lin et al., 2015b)	85.9	82.5
CTransR (Lin et al., 2015b)	85.7	-
TransD (Ji et al., 2015)	86.4	**89.1**
TranSparse-S (Ji et al., 2016)	86.4	88.2
TranSparse-US (Ji et al., 2016)	**86.8**	87.5
NTN (Socher et al., 2013)	70.6	87.2
TransH (Wang et al., 2014)	78.8	83.3
SLogAn (Liang and Forbus, 2015)	75.3	85.3
KG2E (He et al., 2015)	85.4	85.3
Bilinear-COMP (Guu et al., 2015)	77.6	86.1
TransE-COMP (Guu et al., 2015)	80.3	87.6
TransE	85.2	87.6
TransE-NMM	**86.8**	88.6

Table 4: Micro-averaged accuracy results (in %) for triple classification on WN11 (labeled as **W11**) and FB13 (labeled as **F13**) test sets. Scores in **bold** and underline are the best and second best scores, respectively.

Method	Triple class.		Entity pred.	
	Mic.	Mac.	MR	H@10
TransE-LLE	90.08	84.50	535	20.02
SME-LLE	93.64	89.39	253	37.14
SE-LLE	93.95	88.54	447	31.55
TransE-SkipG	85.33	80.06	385	30.52
SME-SkipG	92.86	89.65	293	**39.70**
SE-SkipG	93.07	87.98	412	31.12
TransE	92.13	88.96	309	36.55
TransE-NMM	**94.57**	**90.95**	**238**	37.55

Table 5: Results on on the NELL186 test set. Results for the entity prediction task are in the "Raw" setting. "-SkipG" abbreviates "-Skip-gram".

classification results than TransE in all three experimental datasets, for example, with 2.44% absolute improvement in the micro-averaged accuracy on the NELL186 dataset (i.e. 31% reduction in error). In terms of entity prediction, TransE-NMM obtains better mean rank, MRR and Hits@10 scores than TransE on both FB13 and NELL186 datasets. Specifically, on NELL186 TransE-NMM gains a significant improvement of $279 - 214 = 65$ in the filtered mean rank (which is about 23% relative improvement), while on the FB13 dataset, TransE-NMM improves with $0.267 - 0.213 = 0.054$ in the filtered MRR (which is about 25% relative improvement). On the WN11 dataset, TransE-NMM only achieves better mean rank for entity prediction. The relation prediction results of TransE-NMM and TransE are relatively similar on both WN11 and FB13 be-

cause the number of relation types is small in these two datasets. On NELL186, however, TransE-NMM does significantly better than TransE.

In Table 4, we compare the micro-averaged triple classification accuracy of our TransE-NMM model with the previously reported results on the WN11 and FB13 datasets. The first five rows report the performance of models that use TransE to initialize the entity and relation vectors. The last eight rows present the accuracy of models with randomly initialized parameters.

Table 4 shows that our TransE-NMM model obtains the highest accuracy on WN11 and achieves the second highest result on FB13. Note that there are higher results reported for NTN (Socher et al., 2013), Bilinear-COMP (Guu et al., 2015) and TransE-COMP when entity vectors are initialized by averaging the pre-trained word vectors (Mikolov et al., 2013; Pennington et al., 2014). It is not surprising as many entity names in WordNet and FreeBase are lexically meaningful. It is possible for all other embedding models to utilize the pre-trained word vectors as well. However, as pointed out by Wang et al. (2014) and Guu et al. (2015), averaging the pre-trained word vectors for initializing entity vectors is an open problem and it is not always useful since entity names in many domain-specific KBs are not lexically meaningful.

Table 5 compares the accuracy for triple classification, the raw mean rank and raw Hits@10 scores for entity prediction on the NELL186 dataset. The first three rows present the best results reported in Guo et al. (2015), while the next three rows present the best results reported in Luo et al. (2015). TransE-NMM obtains the highest triple classification accuracy, the best raw mean rank and the second highest raw Hits@10 on the entity prediction task in this comparison.

5.2 Qualitative results

Table 6 presents some examples to illustrate the useful information modeled by the neighbors. We took the relation-specific mixture weights from the learned TransE-NMM model optimized on the entity prediction task, and then extracted three neighbor relations with the largest mixture weights given a relation.

Table 6 shows that those relations are semantically coherent. For example, if we know the place of birth and/or the place of death of a person and/or the location where the person is living, it is likely that we can predict the person's nationality. On the other hand, if we know that a person works for an organization and that this person is also the top member of that organization, then it is possible that this person is the CEO of that organization.

Relation	**Top 3-neighbor relations**
has_instance (WN11)	type_of subordinate_instance_of domain_topic
synset_domain_topic (WN11)	domain_region member_holonym member_meronym
nationality (FB13)	place_of_birth place_of_death location
spouse (FB13)	children, spouse, parents
CEOof (NELL186)	WorksFor TopMemberOfOrganization PersonLeadsOrganization
AnimalDevelopDisease (NELL186)	AnimalSuchAsInsect AnimalThatFeedOnInsect AnimalDevelopDisease

Table 6: Qualitative examples.

5.3 Discussion

Despite of the lower triple classification scores of TransE reported in Wang et al. (2014), Table 4 shows that TransE in fact obtains a very competitive accuracy. Particularly, compared to the relation path model TransE-COMP (Guu et al., 2015), when model parameters were randomly initialized, TransE obtains $85.2 - 80.3 = 4.9\%$ absolute accuracy improvement on the WN11 dataset while achieving similar score on the FB13 dataset. Our high results of the TransE model are probably due to a careful grid search and using the "Bernoulli" trick. Note that Lin et al. (2015b), Ji et al. (2015) and Ji et al. (2016) did not report the TransE results used for initializing TransR, TransD and TranSparse, respectively. They directly copied the TransE results previously reported in Wang et al. (2014). So it is difficult to determine exactly how much TransR, TransD and TranSparse gain over TransE. These models might obtain better results than previously reported when the TransE used for initalization performs as well as reported in this paper. Furthermore, García-Durán et al. (2015), Lin et al. (2015a), García-Durán et al. (2016) and Nickel et al. (2016) also showed that for entity prediction TransE obtains very competitive results which are much higher than the TransE results

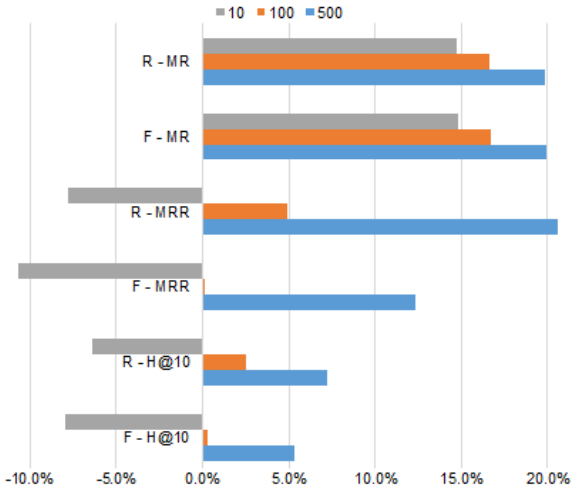

Figure 2: Relative improvement of TransE-NMM against TransE for entity prediction task in WN11 when the filtering threshold $\tau = \{10, 100, 500\}$ (with other hyper-parameters being the same as selected in Section 4.3). Prefixes "R-" and "F-" denote the "Raw" and "Filtered" settings, respectively. Suffixes "-MR", "-MRR" and "-H@10" abbreviate the mean rank, the mean reciprocal rank and Hits@10, respectively.

originally published in Bordes et al. (2013).[3]

As presented in Table 3, for entity prediction using WN11, TransE-NMM with the filtering threshold $\tau = 10$ only obtains better mean rank than TransE (about 15% relative improvement) but lower Hits@10 and mean reciprocal rank. The reason might be that in semantic lexical KBs such as WordNet where relationships between words or word groups are manually constructed, whole neighborhood information might be useful. So when using a small filtering threshold, the model ignores a lot of potential information that could help predicting relationships. Figure 2 presents relative improvements in entity prediction of TransE-NMM over TransE on WN11 when varying the filtering threshold τ. Figure 2 shows that TransE-NMM gains better scores with higher τ value. Specifically, when $\tau = 500$ TransE-NMM does significantly better than TransE in all entity prediction metrics.

6 Conclusion and future work

We introduced a neighborhood mixture model for knowledge base completion by constructing neighbor-based vector representations for entities. We demonstrated its effect by extending TransE (Bordes et al., 2013) with our neighborhood mixture model. On three different datasets, experimental results show that our model significantly improves TransE and obtains better results than the other state-of-the-art embedding models on triple classification, entity prediction and relation prediction tasks. In future work, we plan to apply the neighborhood mixture model to other embedding models, especially to relation path models such as TransE-COMP, to combine the useful information from both relation paths and entity neighborhoods.

[3]They did not report the results on WN11 and FB13 datasets, which are used in this paper, though.

Acknowledgments

This research was supported by a Google award through the Natural Language Understanding Focused Program, and under the Australian Research Council's *Discovery Projects* funding scheme (project number DP160102156). This research was also supported by NICTA, funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. The first author was supported by an International Postgraduate Research Scholarship and a NICTA NRPA Top-Up Scholarship.

References

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 Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 301–306.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. A Semantic Matching Energy Function for Learning with Multi-relational Data. *Machine Learning*, 94(2):233–259.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, Jr., and Tom M.

Mitchell. 2010. Toward an Architecture for Never-ending Language Learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1306–1313.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. 2015. Composing Relationships with Translations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 286–290.

Alberto García-Durán, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. 2016. Combining Two and Three-Way Embedding Models for Link Prediction in Knowledge Bases. *Journal of Artificial Intelligence Research*, 55:715–742.

Matt Gardner and Tom Mitchell. 2015. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498.

Shu Guo, Quan Wang, Bin Wang, Lihong 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 (Volume 1: Long Papers)*, pages 84–94.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing Knowledge Graphs in Vector Space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to Represent Knowledge Graphs with Gaussian Embedding. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 623–632.

Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R Obozinski. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25*, pages 3167–3175.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 985–991.

Denis Krompa, 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.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6(2):167–195.

Chen Liang and Kenneth D. Forbus. 2015. Learning Plausible Inferences from Semantic Web Knowledge by Combining Analogical Generalization with Structured Logistic Regression. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 551–557.

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 Twenty-Ninth AAAI Conference on Artificial Intelligence Learning*, pages 2181–2187.

D. C. Liu and J. Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, 45(3):503–528.

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 (Volume 1: Long Papers)*, pages 156–166.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. STransE: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 460–466.

Maximilian Nickel, Volker Tresp, and Hans-Peter 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, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE, to appear*.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1955–1961.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*, pages 926–934.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706.

Ben Taskar, Ming fai Wong, Pieter Abbeel, and Daphne Koller. 2004. Link Prediction in Relational Data. In *Advances in Neural Information Processing Systems 16*, pages 659–666.

Kristina Toutanova, Xi Victoria Lin, Wen tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional Learning of Embeddings for Relation Paths in Knowledge Bases and Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, June.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge Base Completion via Search-based Question Answering. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 515–526.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations*.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.