An overview of embedding models of entities and relationships for knowledge base completion

Dat Quoc Nguyen
VinAI Research, Vietnam
v.datnq9@vinai.io

Abstract

Knowledge bases (KBs) of real-world facts about entities and their relationships are useful resources for a variety of natural language processing tasks. However, because knowledge bases are typically incomplete, it is useful to be able to perform *knowledge base completion* or *link prediction*, i.e., predict whether a relationship not in the knowledge base is likely to be true. This paper serves as a comprehensive overview of embedding models of entities and relationships for knowledge base completion, summarizing up-to-date experimental results on standard benchmark datasets.

Keywords: Knowledge base completion, link prediction, embedding model, entity prediction.

1 Introduction

Before introducing the KB completion task in detail, let us return to the classic Word2Vec example of a "royal" relationship between "king" and "man", and between "queen" and "woman." As illustrated in this example: $\boldsymbol{v}_{king} - \boldsymbol{v}_{man} \approx \boldsymbol{v}_{queen} - \boldsymbol{v}_{woman}$, word vectors learned from a large corpus can model relational similarities or linguistic regularities between pairs of words as translations in the projected vector space (Mikolov et al., 2013; Pennington et al., 2014). Figure 1 shows another example of a relational similarity between word pairs of countries and capital cities:

$$\begin{aligned}\boldsymbol{v}_{Japan} - \boldsymbol{v}_{Tokyo} &\approx \boldsymbol{v}_{Germany} - \boldsymbol{v}_{Berlin}\\ \boldsymbol{v}_{Germany} - \boldsymbol{v}_{Berlin} &\approx \boldsymbol{v}_{Portugal} - \boldsymbol{v}_{Lisbon}\end{aligned}$$

Let us consider the country and capital pairs in Figure 1 to be pairs of entities rather than word types. That is, we now represent country and capital entities by low-dimensional and dense vectors. The relational similarity between word pairs is presumably to capture a "is_capital_of" relationship between country and capital entities. Also, we represent this relationship by a translation vector $\boldsymbol{v}_{is\_capital\_of}$ in the entity vector space. Thus, we expect:

$$\begin{aligned}\boldsymbol{v}_{Tokyo} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Japan} &\approx \mathbf{0}\\ \boldsymbol{v}_{Berlin} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Germany} &\approx \mathbf{0}\\ \boldsymbol{v}_{Lisbon} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Portugal} &\approx \mathbf{0}\end{aligned}$$

This intuition inspired the TransE model—a well-known embedding model for KB completion or link prediction in KBs (Bordes et al., 2013).

Knowledge bases are collections of real-world triples, where each triple or fact (h, r, t) in KBs represents some relation r between a head entity h and a tail entity t. KBs can thus be formalized as directed multi-relational graphs, where nodes correspond to entities and edges linking the nodes encode various kinds of relationships (García-Durán et al., 2016; Nickel et al., 2016a). Here entities are real-world things or objects such as persons, places, organizations, music tracks or movies. Each relation type defines a certain relationship between entities. For example, as illustrated in Figure 2, the relation type "child_of" relates person entities with each other, while the relation type "born_in" relates person entities

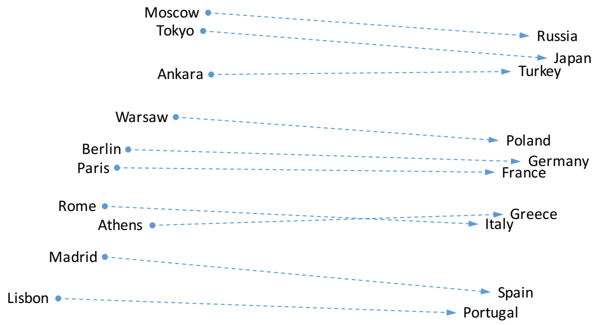

Figure 1: Two-dimensional projection of vectors of countries and their capitals. This figure is drawn based on Mikolov et al. (2013).

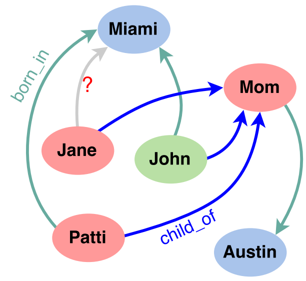

Figure 2: An illustration of (incomplete) knowledge base, with 4 person entities, 2 place entities, 2 relation types and total 6 triple facts. This figure is drawn based on Weston and Bordes (2014).

with place entities. Several KB examples include the domain-specific KB GeneOntology and popular generic KBs of WordNet (Fellbaum, 1998), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), NELL (Carlson et al., 2010) and DBpedia (Lehmann et al., 2015) as well as commercial KBs such as Google's Knowledge Graph, Microsoft's Satori and Facebook's Open Graph. Nowadays, KBs are used in a number of commercial applications including search engines such as Google, Microsoft's Bing and Facebook's Graph search. They also are useful resources for many natural language processing tasks such as question answering (Ferrucci, 2012; Fader et al., 2014), word sense disambiguation (Navigli and Velardi, 2005; Agirre et al., 2013), semantic parsing (Krishnamurthy and Mitchell, 2012; Berant et al., 2013) and co-reference resolution (Ponzetto and Strube, 2006; Dutta and Weikum, 2015).

A main issue is that even very large KBs, such as Freebase and DBpedia, which contain billions of fact triples about the world, are still far from complete. In particular, in English DBpedia 2014, 60% of person entities miss a place of birth and 58% of the scientists do not have a fact about what they are known for (Krompaß et al., 2015). In Freebase, 71% of 3 million person entities miss a place of birth, 75% do not have a nationality while 94% have no facts about their parents (West et al., 2014). So, in terms of a specific application, question answering systems based on incomplete KBs would not provide a correct answer given a correctly interpreted question. For example, given the incomplete KB in Figure 2, it would be impossible to answer the question "where was Jane born ?", although the question is completely matched with existing entity and relation type information (i.e., "Jane" and "born_in") in KB. Consequently, much work has been devoted towards knowledge base completion to perform link prediction in KBs, which attempts to predict whether a relationship/triple not in the KB is likely to be true, i.e., to add new triples by leveraging existing triples in the KB (Lao and Cohen, 2010; Bordes et al., 2012; Gardner et al., 2014; García-Durán et al., 2016). For example, we would like to predict the missing tail entity in the incomplete triple $(\mathsf{Jane}, \mathsf{born\_in}, ?)$ or predict whether the triple $(\mathsf{Jane}, \mathsf{born\_in}, \mathsf{Miami})$ is correct or not.

Embedding models for KB completion have been proven to give state-of-the-art link prediction performances, in which entities are represented by latent feature vectors while relation types are represented by latent feature vectors and/or matrices and/or third-order tensors (Bordes et al., 2013; Socher et al., 2013). *This paper overviews the embedding models for KB completion, and then summarizes up-to-date experimental results on the standard evaluation task of entity prediction*—which is also referred to as the link prediction task (Bordes et al., 2013).

2 A general approach of embedding models for KB completion

Let $\mathcal{E}$ denote the set of entities and $\mathcal{R}$ the set of relation types. Denote by $\mathcal{G}$ the knowledge base consisting of a set of correct triples (h, r, t), such that $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. For each triple (h, r, t), the embedding models define a *score function* $f(h, r, t)$ of its plausibility. Their goal here is to:

Choose f such that the score $f(h, r, t)$ of a correct triple (h, r, t) is higher than the score $f(h', r', t')$ of an incorrect triple (h', r', t').

For example, TransE defines a score function of $f_{\text{TransE}}(h, r, t) = -\|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\|$, where h, r and t are represented by low dimensional vectors $\boldsymbol{v}_h$, $\boldsymbol{v}_r$ and $\boldsymbol{v}_t$, respectively. As $(\mathsf{Tokyo}, \mathsf{is\_capital\_of}, \mathsf{Japan})$ is a correct triple, while $(\mathsf{Tokyo}, \mathsf{is\_capital\_of}, \mathsf{Portugal})$ and $(\mathsf{Lisbon}, \mathsf{is\_capital\_of}, \mathsf{Japan})$ are incorrect ones, we would have: $-\|\boldsymbol{v}_{Tokyo} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Japan}\| > -\|\boldsymbol{v}_{Tokyo} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Portugal}\|$, and $-\|\boldsymbol{v}_{Tokyo} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Japan}\| > -\|\boldsymbol{v}_{Lisbon} + \boldsymbol{v}_{is\_capital\_of} - \boldsymbol{v}_{Japan}\|$. Table 1 in Section 3 summarizes different score functions $f(h, r, t)$.

To learn model parameters (i.e., entity vectors, relation vectors or matrices), the embedding models minimize an objective loss $\mathcal{L}$. A conventional objective loss is the margin-based pairwise ranking loss (Bordes et al., 2013):

$$\mathcal{L}_{\text{Margin}} = \sum_{\substack{(h,r,t)\in\mathcal{G} \\ (h',r,t')\in\mathcal{G}'_{(h,r,t)}}} [\gamma - f(h, r, t) + f(h', r, t')]_+$$

where $[x]_+ = \max(0, x)$; γ is the margin hyper-parameter; and $\mathcal{G}'_{(h,r,t)}$ is the set of incorrect triples generated by corrupting the correct triple $(h, r, t) \in \mathcal{G}$.

Furthermore, the negative log-likelihood (NLL) of softmax regression (SoftReg) (Toutanova and Chen, 2015) and the NLL of logistic regression (LogReg) (Trouillon et al., 2016) are commonly used in recent work:[1]

$$\begin{aligned}
\mathcal{L}_{\text{SoftReg}} &= -\sum_{(h,r,t)\in\mathcal{G}} \left(\frac{\exp\big(f(h,r,t)\big)}{\sum_{t' \in \mathcal{E}\setminus\{t\}} \exp\big(f(h,r,t')\big)} \right. \\
&\qquad\qquad \left. + \frac{\exp\big(f(h,r,t)\big)}{\sum_{h' \in \mathcal{E}\setminus\{h\}} \exp\big(f(h',r,t)\big)} \right) \\
\mathcal{L}_{\text{LogReg}} &= \sum_{(h,r,t)\in\{\mathcal{G}\cup\mathcal{G}'\}} \log\Big(1 + \exp\big(-\mathrm{I}_{(h,r,t)} \cdot f(h,r,t)\big)\Big) \\
&\qquad \text{with: } \mathrm{I}_{(h,r,t)} = \begin{cases} 1 \text{ for } (h,r,t) \in \mathcal{G} \\ -1 \text{ for } (h,r,t) \in \mathcal{G}' \end{cases}
\end{aligned}$$

To corrupt the head or tail entities, a common strategy is to uniformly replace the entities when sampling incorrect triples (Bordes et al., 2013), however it results in many false negative labels (Wang et al., 2014). Domain sampling (Krompaß et al., 2015; Xie et al., 2017) generates corrupted triples by sampling entities from the same domain or from the set of relation-dependent entities. The "*Bernoulli*" trick (Wang et al., 2014) is widely used to set different probabilities for generating head or tail entities: For each relation type r, we calculate the averaged number $a_{r,1}$ of heads h for a pair (r, t) and the averaged number $a_{r,2}$ of tails t for a pair (h, r). We then define a Bernoulli distribution with success probability $\lambda_r = \frac{a_{r,1}}{a_{r,1} + a_{r,2}}$ for sampling: given a correct triple (h, r, t), we corrupt this triple by replacing head entity with probability λ_r while replacing the tail entity with probability $(1 - \lambda_r)$.

Recently, Cai and Wang (2018) and Sun et al. (2019) proposed adversarial learning-based strategies for sampling incorrect triples. However, they did not provide a comparison between the adversarial learning-based strategies and the "*Bernoulli*" trick.

3 Specific models

3.1 Triple-based embedding models

Translation-based models: 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) assumes that the head and tail entities are similar only in a relation-dependent subspace, where each relation is represented by two different matrices. TransE (Bordes et al., 2013) is inspired by models

[1] All the losses can also include an L2 regularization on the model parameters, which is not shown for simplification.

	Model	Score function $f(h,r,t)$
Translation	Unstructured	$-\|\boldsymbol{v}_h - \boldsymbol{v}_t\|_{\ell_{1/2}}$
	SE	$-\|\mathbf{W}_{r,1}\boldsymbol{v}_h - \mathbf{W}_{r,2}\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$
	TransE	$-\|\boldsymbol{v}_h + \boldsymbol{v}_r - \boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\boldsymbol{v}_r \in \mathbb{R}^k$
	TransH	$-\|(\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} - \boldsymbol{r}_p\boldsymbol{r}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\boldsymbol{r}_p, \boldsymbol{v}_r \in \mathbb{R}^k$, $\mathbf{I}$ denotes an identity matrix size $k \times k$
	TransR	$-\|\mathbf{W}_r\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_r\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\mathbf{W}_r \in \mathbb{R}^{n\times k}$, $\boldsymbol{v}_r \in \mathbb{R}^n$
	STransE	$-\|\mathbf{W}_{r,1}\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_{r,2}\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{k\times k}$, $\boldsymbol{v}_r \in \mathbb{R}^k$
	TranSparse	$-\|\mathbf{W}_{r,1}(\theta_{r,1})\boldsymbol{v}_h + \boldsymbol{v}_r - \mathbf{W}_{r,2}(\theta_{r,2})\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{n\times k}$; $\theta_{r,1}, \theta_{r,2} \in \mathbb{R}$; $\boldsymbol{v}_r \in \mathbb{R}^n$
	TransD	$-\|(\mathbf{I} + \boldsymbol{r}_p\boldsymbol{h}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} + \boldsymbol{r}_p\boldsymbol{t}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\boldsymbol{r}_p, \boldsymbol{v}_r, \boldsymbol{h}_p, \boldsymbol{t}_p \in \mathbb{R}^k$
	lppTransD	$-\|(\mathbf{I} + \boldsymbol{r}_{p,1}\boldsymbol{h}_p^\top)\boldsymbol{v}_h + \boldsymbol{v}_r - (\mathbf{I} + \boldsymbol{r}_{p,2}\boldsymbol{t}_p^\top)\boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\boldsymbol{r}_{p,1}, \boldsymbol{r}_{p,2}, \boldsymbol{v}_r, \boldsymbol{h}_p, \boldsymbol{t}_p \in \mathbb{R}^k$
Bilinear & Tensor	Bilinear	$\boldsymbol{v}_h^\top\mathbf{W}_r\boldsymbol{v}_t$ where $\mathbf{W}_r \in \mathbb{R}^{k\times k}$
	DISTMULT	$\boldsymbol{v}_h^\top\mathbf{W}_r\boldsymbol{v}_t$ where $\mathbf{W}_r$ is a diagonal matrix $\in \mathbb{R}^{k\times k}$
	SimplE	$\frac{1}{2}\left(\boldsymbol{v}_{h,1}^\top\mathbf{W}_r\boldsymbol{v}_{t,2} + \boldsymbol{v}_{t,1}^\top\mathbf{W}_{r^{-1}}\boldsymbol{v}_{h,2}\right)$ where $\boldsymbol{v}_{h,1}, \boldsymbol{v}_{h,2}, \boldsymbol{v}_{t,1}, \boldsymbol{v}_{t,2} \in \mathbb{R}^k$; $\mathbf{W}_r$ and $\mathbf{W}_{r^{-1}}$ are diagonal matrices $\in \mathbb{R}^{k\times k}$
	SME(bilinear)	$\boldsymbol{v}_h^\top(\mathbf{M}_1 \times_3 \boldsymbol{v}_r)^\top(\mathbf{M}_2 \times_3 \boldsymbol{v}_r)\boldsymbol{v}_t$ where $\boldsymbol{v}_r \in \mathbb{R}^k$; $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{n\times k\times k}$
	TuckER	$\mathbf{M} \times_1 \boldsymbol{v}_h \times_2 \boldsymbol{v}_r \times_3 \boldsymbol{v}_t$ where $\boldsymbol{v}_r \in \mathbb{R}^n$, $\mathbf{M} \in \mathbb{R}^{k\times n\times k}$; $\times_d$ denotes the tensor product along the d-th mode
	HolE	$\mathsf{sigmoid}(\boldsymbol{v}_t^\top(\boldsymbol{v}_h \star \boldsymbol{v}_r))$ where $\star$ denotes circular correlation
Neural network	NTN	$\boldsymbol{v}_r^\top\mathsf{tanh}(\boldsymbol{v}_h^\top\mathbf{M}_r\boldsymbol{v}_t + \mathbf{W}_{r,1}\boldsymbol{v}_h + \mathbf{W}_{r,2}\boldsymbol{v}_t + \mathbf{b}_r)$ where $\boldsymbol{v}_r, \mathbf{b}_r \in \mathbb{R}^n$; $\mathbf{M}_r \in \mathbb{R}^{k\times k\times n}$; $\mathbf{W}_{r,1}, \mathbf{W}_{r,2} \in \mathbb{R}^{n\times k}$
	ER-MLP	$\mathsf{sigmoid}(\mathbf{w}^\top\mathsf{tanh}(\boldsymbol{W}\mathsf{concat}(\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t)))$
	ConvE	$\boldsymbol{v}_t^\top\mathsf{sigmoid}\left(\boldsymbol{W}\mathsf{vec}\left(\mathsf{sigmoid}\left(\mathsf{concat}(\overline{\boldsymbol{v}}_h, \overline{\boldsymbol{v}}_r) * \boldsymbol{\Omega}\right)\right)\right)$
	ConvKB	$\mathbf{w}^\top\mathsf{concat}\left(\mathsf{sigmoid}\left([\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t] * \boldsymbol{\Omega}\right)\right)$
Complex vector	ComplEx	$\mathsf{Re}\left(\boldsymbol{c}_h^\top\mathbf{C}_r\hat{\boldsymbol{c}}_t\right)$ where $\mathsf{Re}(c)$ denotes the real part of the complex value $c \in \mathbb{C}$ $\boldsymbol{c}_h, \boldsymbol{c}_t \in \mathbb{C}^k$; $\mathbf{C}_r \in \mathbb{C}^{k\times k}$ is a diagonal matrix ; $\hat{\boldsymbol{c}}_t$ is the conjugate of $\boldsymbol{c}_t$
	RotatE	$-\|\boldsymbol{c}_h \circ \boldsymbol{c}_r - \boldsymbol{c}_t\|_{\ell_{1/2}}$; $\boldsymbol{c}_h, \boldsymbol{c}_r, \boldsymbol{c}_t \in \mathbb{C}^k$ where $\circ$ denotes the element-wise product
	QuatE	$\boldsymbol{q}_h \otimes \frac{\boldsymbol{q}_r}{\|\boldsymbol{q}_r\|} \bullet \boldsymbol{q}_t$; $\boldsymbol{q}_h, \boldsymbol{q}_r, \boldsymbol{q}_t \in \mathbb{H}^k$ where $\otimes$ and $\bullet$ denote Hamilton and quaternion inner products
Path	TransE-COMP	$-\|\boldsymbol{v}_h + \boldsymbol{v}_{r_1} + \boldsymbol{v}_{r_2} + ... + \boldsymbol{v}_{r_m} - \boldsymbol{v}_t\|_{\ell_{1/2}}$ where $\boldsymbol{v}_{r_1}, \boldsymbol{v}_{r_2}, ..., \boldsymbol{v}_{r_m} \in \mathbb{R}^k$
	Bilinear-COMP	$\boldsymbol{v}_h^\top\mathbf{W}_{r_1}\mathbf{W}_{r_2}...\mathbf{W}_{r_m}\boldsymbol{v}_t$ where $\mathbf{W}_{r_1}, \mathbf{W}_{r_2}, ..., \mathbf{W}_{r_m} \in \mathbb{R}^{k\times k}$

Table 1: The score functions $f(h,r,t)$ of several prominent embedding models for KB completion. In these models, the entities h and t are represented by vectors $\boldsymbol{v}_h$ and $\boldsymbol{v}_t \in \mathbb{R}^k$, respectively. $\ell_{1/2}$ denotes either the L_1-norm or the squared L_2-norm. In ConvE, $\overline{\boldsymbol{v}}_h$ and $\overline{\boldsymbol{v}}_r$ denote a 2D reshaping of $\boldsymbol{v}_h$ and $\boldsymbol{v}_r$, respectively. In both ConvE and ConvKB models, $*$ and $\boldsymbol{\Omega}$ denote a convolution operator and a set of filters, respectively.

such as the Word2Vec Skip-gram model (Mikolov et al., 2013) where relationships between words often correspond to translations in latent feature space. In particular, TransE learns low-dimensional and dense vectors for every entity and relation type, so that each relation type corresponds to a translation vector operating on the vectors representing the entities, i.e., $\boldsymbol{v}_h + \boldsymbol{v}_r \approx \boldsymbol{v}_t$ for each fact triple (h, r, t). TransE thus is suitable for 1-to-1 relationships, such as "is_capital_of", where a head entity is linked to at most one tail entity given a relation type. Because of using only one translation vector to represent each relation type, TransE is not well-suited for Many-to-1, 1-to-Many and Many-to-Many relationships,[2] such as for relation types "born_in", "place_of_birth" and "research_fields." For example in Figure 2, using one vector representing the relation type "born_in" cannot capture both the translating direction from "Patti" to "Miami" and its inverse direction from "Mom" to "Austin."

To overcome those issues of TransE, TransH (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 TransH by using two projection vectors and a matrix to project entity vectors into a relation-specific space, respectively. Similar to TransR, TransR-FT (Feng et al., 2016a) also uses a matrix to project head and tail entity vectors. TEKE_H (Wang and Li, 2016) extends TransH to incorporate rich context information in an external text corpus. lppTransD (Yoon et al., 2016) extends TransD to additionally use two projection vectors for representing each relation. STransE (Nguyen et al., 2016a) and TranSparse (Ji et al., 2016) can be

[2]A relation type r is classified Many-to-1 if multiple head entities can be connected by r to at most one tail entity. A relation type r is classified 1-to-Many if multiple tail entities can be linked by r from at most one head entity. A relation type r is classified Many-to-Many if multiple head entities can be connected by r to a tail entity and vice versa.

viewed as direct extensions of TransR, where head and tail entities are associated with their own projection matrices. Unlike STransE, TranSparse uses adaptive sparse matrices, whose sparse degrees are defined based on the number of entities linked by relations. TranSparse-DT (Chang et al., 2017) is an extension of TranSparse with a dynamic translation. ITransF (Xie et al., 2017) can be considered as a generalization of STransE, which allows the sharing of statistic regularities between relation projection matrices and alleviates data sparsity issue. Furthermore, TorusE (Ebisu and Ichise, 2018) embeds entities and relations on a torus to handle TransE's regularization problem which forces entity embeddings to be on a sphere in the embedding vector space.

Bilinear- & Tensor-based models: DISTMULT (Yang et al., 2015) is based on the Bilinear model (Nickel et al., 2011; Jenatton et al., 2012) where each relation is represented by a diagonal matrix rather than a full matrix. SimplE (Kazemi and Poole, 2018) extends DISTMULT to allow two embeddings of each entity to be learned dependently. Such quadratic forms are also used to model entities and relations in KG2E (He et al., 2015), TATEC (García-Durán et al., 2016), TransG (Xiao et al., 2016), RSTE (Tay et al., 2017), ANALOGY (Liu et al., 2017) and Dihedral (Xu and Li, 2019). SME(bilinear) (Bordes et al., 2012) is proposed to first separately combine entity-relation pairs (h, r) and (r, t) and then semantically match these combinations, using tensor product. HolE (Nickel et al., 2016b) uses circular correlation– a compositional operator–which can be interpreted as a compression of the tensor product. In addition, TuckER (Balazevic et al., 2019) is a linear model based on the Tucker tensor decomposition of the binary tensor representation of KB triples.

Neural network-based models: The neural tensor network (NTN) model (Socher et al., 2013) also uses a bilinear tensor operator to represent each relation while ProjE (Shi and Weninger, 2017) can be viewed as simplified versions of NTN. ER-MLP (Dong et al., 2014) represents each triple by a vector obtained from concatenating head, relation and tail embeddings, then feeds this vector into a single-layer MLP with one-node output layer. ConvE (Dettmers et al., 2018) and ConvKB (Nguyen et al., 2018) are based on convolutional neural networks. ConvE uses a convolution layer directly over 2D reshaping of head-entity and relation embeddings, while ConvKB applies a convolution layer over the embedding triples (here each triple (h, r, t) is represented as a 3-column matrix where each column vector represents a triple element). HypER (Balažević et al., 2019) simplifies ConvE by using a hypernetwork to produce 1D convolutional filters for each relation, then extracts relation-specific features from head entity embeddings. Conv-TransE (Shang et al., 2019) extends ConvE to keep the translational characteristic between entities and relations. InteractE (Vashishth et al., 2020) uses a circular convolution operator and a checkered reshaping function instead of the standard convolution operator and 2D stack reshaping function in ConvE. CapsE (Nguyen et al., 2019) extends ConvKB by stacking a capsule network layer (Sabour et al., 2017) on top of the convolution layer.

Complex vector-based models: Instead of embedding entities and relations in the real-valued vector space, ComplEx (Trouillon et al., 2016) is an extension of DISTMULT in the complex vector space. ComplEx-N3 (Lacroix et al., 2018) extends ComplEx with weighted nuclear 3-norm. Also in the complex vector space, RotatE (Sun et al., 2019) defines each relation as a rotation from the head entity to the tail entity. QuatE (Zhang et al., 2019) represents entities by quaternion embeddings (i.e. hypercomplex-valued embeddings) and models relations as rotations in the quaternion space by employing the Hamilton and quaternion-inner products.

3.2 Relation path-based embedding models

All embedding models mentioned above in Section 3.1 only take triples into account. Thus, these models ignore potentially useful information implicitly presented by the structure of the KB. For example, the relation path $h \xrightarrow{\mathsf{born\_in\_city}} e \xrightarrow{\mathsf{city\_in\_country}} t$ should indicate a relationship "nationality" between the h and t entities. Also, neighborhood information of entities could be useful for predicting the relationship between two entities as well. For example, in the KB NELL (Carlson et al., 2010), we have information such as if a person works for an organization and this person also leads that organization, then it is likely that this person is the CEO of that organization.

Many research has shown that relation paths between entities in KBs provide richer context information and improve the performance of embedding models for KB completion (Luo et al., 2015; Liang and Forbus, 2015; García-Durán et al., 2015; Guu et al., 2015; Toutanova et al., 2016; Durán and Niepert, 2018; Takahashi et al., 2018; Chen et al., 2018). In particular, Luo et al. (2015) constructed relation paths between entities and, viewing entities and relations in the path as pseudo-words, then 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 models TransE (Bordes et al., 2013), SME (Bordes et al., 2012) and SE (Bordes et al., 2011). Liang and Forbus (2015) used the plausibility score produced by SME to compute the weights of relation paths.

PTransE-RNN (Lin et al., 2015a) models relation paths by using a recurrent neural network (RNN). Das et al. (2017)'s model and ROPs (Yin et al., 2018) also apply RNN to model the path between an entity pair, however, in contrast to PTransE-RNN, they additionally take the intermediate entities present in the path into account. IRN (Shen et al., 2017) uses a shared memory and RNN-based controller to implicitly model multi-step structured relationships. RTransE (García-Durán et al., 2015), PTransE-ADD (Lin et al., 2015a) and TransE-COMP (Guu et al., 2015) extend TransE to represent a relation path by a vector which is the sum of the vectors of all relations in the path. In Bilinear-COMP (Guu et al., 2015) and PRUNED-PATHS (Toutanova et al., 2016), each relation is a matrix and so it represents the relation path by matrix multiplication. Durán and Niepert (2018) proposed the KB_{LRN} framework to combine relational paths with latent and numerical features.

The neighborhood mixture model TransE-NMM (Nguyen et al., 2016b) 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. Neighborhood information is also exploited in R-GCN (Schlichtkrull et al., 2018), SACN (Shang et al., 2019) and KBAT (Nathani et al., 2019), which generalize graph convolutional networks (Kipf and Welling, 2017) and graph attention networks (Veličković et al., 2018) for dealing with highly multi-relational data such as knowledge bases. For computing the final representation of an entity, they make use of layer-wise propagation to accumulate linearly-transformed embeddings of its neighboring entities through a normalized sum with different relational weights. For link prediction, R-GCN, SACN and KBAT apply DISTMULT, Conv-TransE and ConvKB to compute triple scores, respectively.

3.3 Other KB completion models

The Path Ranking Algorithm (PRA) (Lao and Cohen, 2010) is a random walk inference technique which was proposed to predict a new relationship between two entities in KBs. Lao et al. (2011) used PRA to estimate the probability of an unseen triple as a combination of weighted random walks that follow different paths linking the head entity and tail entity in the KB. Gardner et al. (2014) made use of an external text corpus to increase the connectivity of the KB used as the input to PRA. Gardner and Mitchell (2015) improved PRA by proposing a subgraph feature extraction technique to make the generation of random walks in KBs more efficient and expressive, while Wang et al. (2016) extended PRA to couple the path ranking of multiple relations. PRA can also be used in conjunction with first-order logic in the discriminative Gaifman model (Niepert, 2016). In addition, Neelakantan et al. (2015) used a RNN to learn vector representations of PRA-style relation paths between entities in the KB. Other random-walk based learning algorithms for KB completion can be also found in Feng et al. (2016b), Liu et al. (2016), Wei et al. (2016), Mazumder and Liu (2017) and Das et al. (2018).

Yang et al. (2017) proposed a Neural Logic Programming (LP) framework to learning probabilistic first-order logical rules for KB reasoning, producing competitive link prediction performances. Feldman et al. (2019) presented an approach to generate sentences from triples via hand-craft templates, and then use the likelihoods produced by the pre-trained BERT (Devlin et al., 2019) for these generated sentences to score the plausibility of the corresponding triples. See other methods for learning from KBs and multi-relational data in Nickel et al. (2016a) and Wang et al. (2017).

Dataset	$\mid \mathcal{E} \mid$	$\mid \mathcal{R} \mid$	#Triples in train/valid/test		
FB15k	14,951	1,345	483,142	50,000	59,071
WN18	40,943	18	141,442	5,000	5,000
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134

Table 2: Statistics of benchmark experimental datasets.

4 Evaluation task

The standard evaluation task of entity prediction, i.e. link prediction (Bordes et al., 2013), is proposed to evaluate embedding models for KB completion.[3]

Datasets: Information about benchmark datasets for KB completion evaluation is given in Table 2. FB15k and WN18 are derived from the large real-world KB FreeBase (Bollacker et al., 2008) and the large lexical KB WordNet (Miller, 1995), respectively. Toutanova and Chen (2015) noted that FB15k and WN18 are not challenging datasets because they contain many reversible triples. Dettmers et al. (2018) showed a concrete example: A test triple (feline, hyponym, cat) can be mapped to a training triple (cat, hypernym, feline), thus knowing that "hyponym" and "hypernym" are reversible allows us to easily predict the majority of test triples. So, datasets FB15k-237 (Toutanova and Chen, 2015) and WN18RR (Dettmers et al., 2018) are created to serve as realistic KB completion datasets which represent a more challenging learning setting. FB15k-237 and WN18RR are subsets of FB15k and WN18, respectively.

4.1 Task description

The entity prediction task, i.e. link prediction (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.

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 these candidates are ranked in descending order of their plausibility score. The "Filtered" setting protocol, described in Bordes et al. (2013), filters 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, thus this "Filtered" setting provides a clear 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 is ranked in the top 10 predictions), which were originally used in the entity prediction task (Bordes et al., 2013), recent work also reports the mean reciprocal rank (**MRR**).[4] Mean rank is always greater or equal to 1 and the lower mean rank indicates better entity prediction performance, while MRR and Hits@10 scores always range from 0.0 to 1.0, and higher score reflects better prediction result.

4.2 Main results

Tables 3 and 4 list recent entity prediction results of KB completion models on FB15k and WN18 and on FB15k-237 and WN18RR, respectively. In Table 3, the first 27 rows report the performance of triple-based models that directly optimize a score function for the triples in a KB, i.e. they do not exploit information about alternative paths between head and tail entities. The next 9 rows report results of models that exploit information about relation paths or neighborhood information. The last 2 rows present results for models which make use of textual mentions derived from a large external corpus. In Table 4, the last 5 rows report results of models that exploit the path or neighborhood information.

In general, Tables 3 and 4 show that the models using external corpus information or employing path information achieve better scores than the triple-based models that do not use such information. In terms

[3]Another evaluation task for KB completion is triple classification (Socher et al., 2013), however, it is not as widely used as the link prediction task.

[4]See Baeza-Yates and Ribeiro-Neto (2011) for definitions of the mean rank, Hits@10 and MRR. Some recent work additionally reported Hits@1. However, formulas of MRR and Hits@1 show a strong correlation between these two scores. So using Hits@1 might not really reveal any additional insight.

Method	Filtered					
	FB15k			WN18		
	MR	@10	MRR	MR	@10	MRR
TransR (Lin et al., 2015b)	77	68.7	-	225	92.0	-
CTransR (Lin et al., 2015b)	75	70.2	-	218	92.3	-
KG2E (He et al., 2015)	59	74.0	-	331	92.8	-
TransD (Ji et al., 2015)	91	77.3	-	212	92.2	-
lppTransD (Yoon et al., 2016)	78	78.7	-	270	94.3	-
TransG (Xiao et al., 2016)	98	79.8	-	470	93.3	-
TranSparse (Ji et al., 2016)	82	79.5	-	211	93.2	-
TranSparse-DT (Chang et al., 2017)	79	80.2	-	221	94.3	-
ITransF (Xie et al., 2017)	65	81.0	-	<u>205</u>	94.2	-
NTN (Socher et al., 2013)	-	41.4	0.25	-	66.1	0.53
TransE (Bordes et al., 2013) [♡]	-	74.9	0.463	-	94.3	0.495
HolE (Nickel et al., 2016b)	-	73.9	0.524	-	94.9	0.938
ComplEx (Trouillon et al., 2016)	-	84.0	0.692	-	94.7	0.941
ANALOGY (Liu et al., 2017)	-	85.4	0.725	-	94.7	0.942
SimplE (Kazemi and Poole, 2018)	-	83.8	0.727	-	94.7	0.942
TorusE (Ebisu and Ichise, 2018)	-	83.2	0.733	-	95.4	0.947
STransE (Nguyen et al., 2016a)	69	79.7	0.543	206	93.4	0.657
ER-MLP (Dong et al., 2014) [♠]	81	80.1	0.570	299	94.2	0.895
DISTMULT (Yang et al., 2015) [♣]	42	89.3	<u>0.798</u>	655	94.6	0.797
ConvE (Dettmers et al., 2018)	64	87.3	0.745	504	95.5	0.942
HypER (Balažević et al., 2019)	44	88.5	0.790	431	95.8	<u>0.951</u>
RotatE (Sun et al., 2019)	40	88.4	0.797	309	<u>95.9</u>	0.949
QuatE (Zhang et al., 2019)	**17**	90.0	0.782	**162**	<u>95.9</u>	0.950
ComplEx-N3 (Lacroix et al., 2018)	-	<u>91</u>	**0.86**	-	**96**	0.95
TuckER (Balazevic et al., 2019)	-	89.2	0.795	-	95.8	**0.953**
IRN (Shen et al., 2017)	38	**92.7**	-	249	95.3	-
ProjE (Shi and Weninger, 2017)	<u>34</u>	88.4	-	-	-	-
rTransE (García-Durán et al., 2015)	50	76.2	-	-	-	-
PTransE-ADD (Lin et al., 2015a)	58	84.6	-	-	-	-
PTransE-RNN (Lin et al., 2015a)	92	82.2	-	-	-	-
GAKE (Feng et al., 2016b)	119	64.8	-	-	-	-
Gaifman (Niepert, 2016)	75	84.2	-	352	93.9	-
Hiri (Liu et al., 2016)	-	70.3	0.603	-	90.8	0.691
Neural LP (Yang et al., 2017)	-	83.7	0.76	-	94.5	**0.94**
R-GCN+ (Schlichtkrull et al., 2018)	-	84.2	0.696	-	**96.4**	0.819
KB_{LRN} (Durán and Niepert, 2018)	**44**	**87.5**	**0.794**	-	-	-
TEKE_H (Wang and Li, 2016)	108	73.0	-	**114**	92.9	-
SSP (Xiao et al., 2017)	82	79.0	-	156	93.2	-

Table 3: Entity prediction results on WN18 and FB15k. **MR** and **@10** denote evaluation metrics of mean rank and Hits@10 (in %), respectively. TransG's results are taken from its latest ArXiv version (https://arxiv.org/abs/1509.05488v7). NTN's results are taken from Yang *et al.* (2015). [♡], **[♠]** and **[♣]** denote results taking from Nickel *et al.* (2016b), Ravishankar *et al.* (2017) and Kadlec *et al.* (2017), respectively.

of models not exploiting path or external information, the complex vector-based models (e.g. QuatE, CompleEx-N3 and RotatE) produce the strongest evaluation scores, followed by the neural network-based models (e.g. CapsE, InteractE and HypER).[5] Tables 3 and 4 also show that TransE and DIST-MULT, despite of theirs simplicity, can produce very competitive results (by performing a careful grid search of hyper-parameters).

[5]CapsE uses the pre-trained word embeddings for entity vector initialization on WN18RR. It is not surprising that CapsE produces the best MR on WN18RR as many entity names in WordNet are lexically meaningful. It is possible for all other embedding models to utilize the pre-trained word vectors as well. However, averaging the pre-trained word embeddings 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 (Guu et al., 2015).

Method	Filtered					
	FB15k-237			WN18RR		
	MR	@10	MRR	MR	@10	MRR
IRN (Shen et al., 2017)	211	46.4	-	-	-	-
KBGAN (Cai and Wang, 2018)	-	45.8	0.278	-	48.1	0.213
DISTMULT (Yang et al., 2015) [♢]	254	41.9	0.241	5110	49	0.43
ComplEx (Trouillon et al., 2016) [♢]	339	42.8	0.247	5261	51	0.44
ConvE (Dettmers et al., 2018)	246	49.1	0.316	5277	48	0.46
ER-MLP (Dong et al., 2014) [♠]	219	54.0	0.342	4798	41.9	0.366
HypER (Balažević et al., 2019)	250	52.0	0.341	5798	52.2	0.465
TransE (Bordes et al., 2013) [♡]	347	46.5	0.294	743	56.0	0.245
ConvKB (Nguyen et al., 2018) [♡]	254	53.2	0.418	763	56.7	0.253
CapsE (Nguyen et al., 2019)	303	**59.3**	**0.523**	**719**	56.0	0.415
InteractE (Vashishth et al., 2020)	172	53.5	0.354	5202	52.8	0.463
RotatE (Sun et al., 2019)	177	53.3	0.338	3340	57.1	0.476
QuatE (Zhang et al., 2019)	**87**	55.0	0.348	2314	**58.2**	**0.488**
ComplEx-N3 (Lacroix et al., 2018)	-	56	0.37	-	57	0.48
Conv-TransE (Shang et al., 2019)	-	51	0.33	-	52	0.46
TuckER (Balazevic et al., 2019)	-	54.4	0.358	-	52.6	0.470
Neural LP (Yang et al., 2017)	-	36.2	0.24	-	-	-
R-GCN+ (Schlichtkrull et al., 2018)	-	41.7	0.249	-	-	-
KB_{LRN} (Durán and Niepert, 2018)	**209**	49.3	0.309	-	-	-
KBAT (Nathani et al., 2019)	210	**62.6**	**0.518**	**1940**	**58.1**	0.440
SACN (Shang et al., 2019)	-	54	0.35	-	54	**0.47**

Table 4: Entity prediction results on WN18RR and FB15k-237. [♢], [♠] and [♡] denote results taking from Dettmers *et al.* (2018), Ravishankar *et al.* (2017) and Nguyen *et al.* (2019), respectively.

5 Discussion and Conclusion

The reasons why much work has been devoted towards developing triple-based models are: (1) additional information sources might not be available, e.g., for KBs for specialized domains, (2) models that do not exploit path information or external resources are simpler and thus typically much faster to train than the more complex models using path or external information, and (3) the more complex models that exploit path or external information are typically extensions of these simpler models, and are often initialized with parameters estimated by such simpler models, so improvements to the simpler models should yield corresponding improvements to the more complex models as well (Nguyen et al., 2016a).

It is worth to further explore those KB completion embedding models for a new application where we could formulate its corresponding data into triples. For example, in Web search engines, we observe *user-oriented* relationships between submitted *queries* and *documents* returned by the search engines. That is, we have triple representations (query, user, document) in which for each user-oriented relationship, we would have many queries and documents, resulting in a lot of Many-to-Many relationships. Inspired by this observation, Vu et al. (2017) applied STransE (Nguyen et al., 2016a) for a *search personalization* task to re-rank the search documents returned by a search engine for users' submitted queries. Other application examples can be also found for recommender systems (Zhang et al., 2016; He et al., 2017; Cao et al., 2019), social relation extraction (Tu et al., 2017) and visual relation detection (Zhang et al., 2017). Future directions might also include: (i) Combining logical rules which contain rich background information and KB triples in a unified KB completion framework, e.g. jointly embedding KBs and logical rules (Guo et al., 2016; Yang et al., 2017). (ii) Recent embedding models for KB completion hold a closed-world assumption where the KBs are fixed (i.e. new entities might not be added easily), thus it would be worth exploring open-world KB completion models to connect unseen entities to the KBs (Shi and Weninger, 2018). (iii) Investigate efficient methods which can be applied to large-scale KBs of millions of entities and relations (Zhang et al., 2020).

In this paper, we have presented a comprehensive overview of embedding models of entity and relationships for knowledge base completion. The paper also provides update-to-date experimental results of the embedding models on the entity prediction (i.e. link prediction) task on benchmark datasets FB15k, WN18, FB15k-237 and WN18RR.

References

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2013. Random Walks for Knowledge-Based Word Sense Disambiguation. *Computational Linguistics*, 40(1):57–84.

Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. 2011. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Hypernetwork knowledge graph embeddings. In *ICANN*, pages 553–565.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *EMNLP-IJCNLP*, pages 5185–5194.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *EMNLP*, pages 1533–1544.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*, pages 1247–1250.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *AAAI*, 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 *NIPS*, pages 2787–2795.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial Learning for Knowledge Graph Embeddings. In *NAACL-HLT*, pages 1470–1480.

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *WWW*, page 151–161.

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 *AAAI*, pages 1306–1313.

L. Chang, M. Zhu, T. Gu, C. Bin, J. Qian, and J. Zhang. 2017. Knowledge Graph Embedding by Dynamic Translation. *IEEE Access*, 5:20898–20907.

Wenhu Chen, Wenhan Xiong, Xifeng Yan, and William Yang Wang. 2018. Variational knowledge graph reasoning. In *NAACL-HLT*, pages 1823–1832.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017. Chains of Reasoning over Entities, Relations, and Text using Recurrent Neural Networks. In *EACL*, pages 132–141.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *ICLR*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*, pages 1811–1818.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610.

Alberto García Durán and Mathias Niepert. 2018. KBLRN: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In *UAI*.

Sourav Dutta and Gerhard Weikum. 2015. Cross-Document Co-Reference Resolution using Sample-Based Clustering with Knowledge Enrichment. *Transactions of ACL*, 3:15–28.

Takuma Ebisu and Ryutaro Ichise. 2018. TorusE: Knowledge Graph Embedding on a Lie Group. In *AAAI*, pages 1819–1826.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open Question Answering over Curated and Extracted Knowledge Bases. In *KDD*, pages 1156–1165.

Joshua Feldman, Joe Davison, and Alexander M. Rush. 2019. Commonsense Knowledge Mining from Pretrained Models. In *EMNLP-IJCNLP*, pages 1173–1178.

Christiane D. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Jun Feng, Minlie Huang, Mingdong Wang, Mantong Zhou, Yu Hao, and Xiaoyan Zhu. 2016a. Knowledge graph embedding by flexible translation. In *KR*, pages 557–560.

Jun Feng, Minlie Huang, Yang Yang, and xiaoyan zhu. 2016b. GAKE: Graph Aware Knowledge Embedding. In *COLING*, pages 641–651.

David Angelo Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3):235–249.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. 2015. Composing Relationships with Translations. In *EMNLP*, 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 *EMNLP*, pages 1488–1498.

Matt Gardner, Partha P. Talukdar, Jayant Krishnamurthy, and Tom M. Mitchell. 2014. Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases. In *EMNLP*, pages 397–406.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *EMNLP*, pages 192–202.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing Knowledge Graphs in Vector Space. In *EMNLP*, pages 318–327.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to Represent Knowledge Graphs with Gaussian Embedding. In *CIKM*, pages 623–632.

Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *RecSys*, page 161–169.

Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R Obozinski. 2012. A latent factor model for highly multi-relational data. In *NIPS*, pages 3167–3175.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *ACL-IJCNLP*, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In *AAAI*, pages 985–991.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *NIPS*, pages 4284–4295.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Jayant Krishnamurthy and Tom Mitchell. 2012. Weakly Supervised Training of Semantic Parsers. In *EMNLP-CoNLL*, pages 754–765.

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-Constrained Representation Learning in Knowledge Graphs. In *ISWC*, pages 640–655.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical Tensor Decomposition for Knowledge Base Completion. In *ICML*, pages 2869–2878.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random Walk Inference and Learning in a Large Scale Knowledge Base. In *EMNLP*, pages 529–539.

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 *AAAI*, 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 *EMNLP*, 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 *AAAI*, pages 2181–2187.

Qiao Liu, Liuyi Jiang, Minghao Han, Yao Liu, and Zhiguang Qin. 2016. Hierarchical Random Walk Inference in Knowledge Graphs. In *SIGIR*, pages 445–454.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical Inference for Multi-relational Embeddings. In *ICML*, pages 2168–2178.

Yuanfei Luo, Quan Wang, Bin Wang, and Li Guo. 2015. Context-Dependent Knowledge Graph Embedding. In *EMNLP*, pages 1656–1661.

Sahisnu Mazumder and Bing Liu. 2017. Context-aware Path Ranking for Knowledge Base Completion. In *IJCAI*, pages 1195–1201.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, pages 3111–3119.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *ACL*, pages 4710–4723.

Roberto Navigli and Paola Velardi. 2005. Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional Vector Space Models for Knowledge Base Completion. In *ACL-IJCNLP*, pages 156–166.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016a. STransE: a novel embedding model of entities and relationships in knowledge bases. In *NAACL-HLT*, pages 460–466.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016b. Neighborhood Mixture Model for Knowledge Base Completion. In *CoNLL*, pages 40–50.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *NAACL-HLT*, pages 327–333.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2019. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. In *NAACL-HLT*, pages pages 2180–2189.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*, pages 809–816.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016a. A Review of Relational Machine Learning for Knowledge Graphs. *the IEEE*, 104(1):11–33.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016b. Holographic embeddings of knowledge graphs. In *AAAI*, pages 1955–1961.

Mathias Niepert. 2016. Discriminative Gaifman Models. In *NIPS*, pages 3405–3413.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*, pages 1532–1543.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *NAACL*, pages 192–199.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *NIPS*, pages 3856–3866.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*, pages 593–607.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end Structure-Aware Convolutional Networks for Knowledge Base Completion. In *AAAI*, pages 3060–3067.

Yelong Shen, Po-Sen Huang, Ming-Wei Chang, and Jianfeng Gao. 2017. Modeling Large-Scale Structured Relationships with Shared Memory for Knowledge Base Completion. In *Rep4NLP*, pages 57–68.

Baoxu Shi and Tim Weninger. 2017. ProjE: Embedding Projection for Knowledge Graph Completion. In *AAAI*, pages 1236–1242.

Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. In *AAAI*, pages 1957–1964.

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

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge. In *WWW*, pages 697–706.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*.

Ryo Takahashi, Ran Tian, and Kentaro Inui. 2018. Interpretable and Compositional Relation Learning by Joint Training with an Autoencoder. In *ACL*, pages 2148–2159.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Falk Brauer. 2017. Random Semantic Tensor Ensemble for Scalable Knowledge Graph Link Prediction. In *WSDM*, pages 751–760.

Kristina Toutanova and Danqi Chen. 2015. Observed Versus Latent Features for Knowledge Base and Text Inference. In *CVSC*, pages 57–66.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional Learning of Embeddings for Relation Paths in Knowledge Base and Text. In *ACL*, pages 1434–1444.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML*, pages 2071–2080.

Cunchao Tu, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2017. TransNet: Translation-Based Network Representation Learning for Social Relation Extraction. In *IJCAI*, pages 2864–2870.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. InteractE: Improving Convolution-based Knowledge Graph Embeddings by Increasing Feature Interactions. In *AAAI*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search Personalization with Embeddings. In *ECIR*, pages 598–604.

Zhigang Wang and Juan-Zi Li. 2016. Text-Enhanced Representation Learning for Knowledge Graph. In *IJCAI*, pages 1293–1299.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*, pages 1112–1119.

Quan Wang, Jing Liu, Yuanfei Luo, Bin Wang, and Chin-Yew Lin. 2016. Knowledge Base Completion via Coupled Path Ranking. In *ACL*, pages 1308–1318.

Q. Wang, Z. Mao, B. Wang, and L. Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhuoyu Wei, Jun Zhao, and Kang Liu. 2016. Mining Inference Formulas by Goal-Directed Random Walks. In *EMNLP*, pages 1379–1388.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge Base Completion via Search-based Question Answering. In *WWW*, pages 515–526.

Jason Weston and Antoine Bordes. 2014. Embedding Methods for NLP. In *EMNLP 2014 tutorial*.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. TransG : A Generative Model for Knowledge Graph Embedding. In *ACL*, pages 2316–2325.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2017. SSP: semantic space projection for knowledge graph embedding with text descriptions. In *AAAI*, pages 3104–3110.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An Interpretable Knowledge Transfer Model for Knowledge Base Completion. In *ACL*, pages 950–962.

Canran Xu and Ruijiang Li. 2019. Relation Embedding with Dihedral Group in Knowledge Graph. In *ACL*, pages 263–272.

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 *ICLR*.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *NIPS*, pages 2316–2325.

Wenpeng Yin, Yadollah Yaghoobzadeh, and Hinrich Schütze. 2018. Recurrent One-Hop Predictions for Reasoning over Knowledge Graphs. In *COLING*, pages 2369–2378.

Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2016. A Translation-Based Knowledge Graph Embedding Preserving Logical Property of Relations. In *NAACL-HLT*, pages 907–916.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *KDD*, page 353–362.

Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual translation embedding network for visual relation detection. In *CVPR*, pages 5532–5540.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems 32*, pages 2735–2745.

Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. 2020. Efficient probabilistic logic reasoning with graph neural networks. In *ICLR*.