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

Dat Quoc Nguyen
School of Computing and Information Systems
The University of Melbourne, Australia
dqnguyen@unimelb.edu.au

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 article serves as a brief overview of embedding models of entities and relationships for knowledge base completion, summarizing up-to-date experimental results on standard benchmark datasets FB15k, WN18, FB15k-237, WN18RR, FB13 and WN11.

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

1 Introduction

Before introducing the KB completion task in details, 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: $v_{king} - v_{man} \approx v_{queen} - v_{woman}$, 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:

$v_{Tokyo} + v_{is\_capital\_of} - v_{Japan} \approx 0$

$v_{Berlin} + v_{is\_capital\_of} - v_{Germany} \approx 0$

$v_{Rome} + v_{is\_capital\_of} - v_{Italy} \approx 0$

$v_{Lisbon} + v_{is\_capital\_of} - v_{Portugal} \approx 0$

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 relationship (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 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 NLP 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 (Jane, born_in, ?) or predict whether the triple (Jane, born_in, 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 (Nickel et al., 2011; Jeannot et al., 2012; Bordes et al., 2013; Wang et al., 2014; Dong et al., 2014; Lin et al., 2015b; Guu et al., 2015; Krompaß et al., 2015; Toutanova and Chen, 2015; García-Durán et al., 2016; Trouillon et al., 2016; Toutanova et al., 2016; Nickel et al., 2016b). This article briefly overviews the embedding models for KB completion, and then summarizes up-to-date experimental results on two standard evaluation tasks: i) the entity prediction task—which is also referred to as the link prediction task (Bordes et al., 2013)—and ii) the triple classification task (Socher et al., 2013).

2 Embedding models for KB completion

2.1 A general approach

Let \( E \) denote the set of entities and \( R \) the set of relation types. Denote by \( G \) the knowledge base consisting of a set of correct triples \( (h, r, t) \), such that \( h, t \in E \) and \( r \in R \). For each triple \( (h, r, t) \), the embedding models define a score function \( f(h, r, t) \) of its implausibility. Their 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 \( (h', r', t') \).
| Model          | Score function $f(h, r, t)$                                      | Opt.   |
|---------------|-----------------------------------------------------------------|--------|
| Unstructured  | $\|v_h - v_t\|_{\ell_1/2}$                                      | SGD    |
| SE            | $\|W_{r,1}v_h - W_{r,2}v_t\|_{\ell_1/2} : W_{r,1}, W_{r,2} \in \mathbb{R}^{k \times k}$ | SGD    |
| SME           | $(W_{1,1}v_h + W_{1,2}v_r + b_1)^\top (W_{2,1}v_1 + W_{2,2}v_r + b_2)$ | SGD    |
| TransE        | $\|v_h + v_r - v_t\|_{\ell_1/2} : v_r \in \mathbb{R}^k$         | SGD    |
| TransH        | $\|(I - r_p^\top r_p) v_h + v_r - (I - r_p^\top r_p) v_t\|_{\ell_1/2}$ | SGD    |
| TransR        | $\|W_r v_h + v_r - W_r v_t\|_{\ell_1/2} : W_r \in \mathbb{R}^{n \times k} ; v_r \in \mathbb{R}^n$ | SGD    |
| TransD        | $\|(I + r_p h_p^\top) v_h + v_r - (I + r_p t_p^\top) v_t\|_{\ell_1/2}$ | AdaDelta |
| IppTransD     | $\|(I + r_p,1 h_p^\top) v_h + v_r - (I + r_p,2 t_p^\top) v_t\|_{\ell_1/2}$ | SGD    |
| STTransE      | $\|W_{r,1}v_h + v_r - W_{r,2}v_t\|_{\ell_1/2} : W_{r,1}, W_{r,2} \in \mathbb{R}^{k \times k} ; v_r \in \mathbb{R}^k$ | SGD    |
| TransSparse   | $\|W_p^h(\theta_p^h) v_h + v_r - W_p^t(\theta_p^t) v_t\|_{\ell_1/2}$ | SGD    |
| DISTMULT      | $\|W_r v_h ; W_r \text{ is a diagonal matrix } \in \mathbb{R}^{k \times k}$ | AdaGrad |
| NTN           | $\|v_r^t \text{tanh} (v_h^t M_r v_h + W_{r,1} v_h + W_{r,2} v_r + b_r)\|_{\ell_1/2}$ | L-BFGS |
| HolE          | $\text{sigmoid} (v_r^t (v_h \circ v_t)) : v_r \in \mathbb{R}^k , \circ \text{ denotes circular correlation}$ | AdaGrad |
| Bilinear-COMP | $\|v_h, W_r, W_{r,2} \ldots W_{r,m} v_t ; W_{r,1}, W_{r,2} \ldots , W_{r,m} \in \mathbb{R}^{k \times k}$ | AdaGrad |
| TransE-COMP   | $\|v_h + v_r + v_r + \ldots + v_r - v_t\|_{\ell_1/2} : v_r, v_r, \ldots , v_r \in \mathbb{R}^k$ | AdaGrad |
| ConvE         | $\| \text{vec} (g (\text{concat} (\nabla_h (\nabla_r * \Omega)) W)) ; g \text{ denotes a non-linear function}$ | Adam   |
| ConvKB        | $\| \text{concat} (g ((v_h, v_r, v_t \circ \Omega)) ; \circ \text{ denotes a convolution operator}$ | Adam   |

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 $v_h$ and $v_t \in \mathbb{R}^k$, respectively. In ConvE, $\nabla_h$ and $\nabla_r$ denote a 2D reshaping of $v_h$ and $v_r$, respectively. In both ConvE and ConvKB, $\Omega$ denotes a set of filters.

2.2 Specific 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. Furthermore, the SME model (Bordes et al., 2012) uses four different matrices to project entity and relation vectors into a subspace. The TransE model (Bordes et al., 2013) is inspired by models such as the Word2Vec Skip-gram model (Mikolov et al., 2013) where
relationships between words often correspond to translations in latent feature space. TorusE (Ebisu and Ichise, 2018) embeds entities and relations on a torus to handle TransE’s regularization problem.

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. 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., 2016b) and TranSparse (Ji et al., 2016) can be viewed as direct extensions of the TransR model, where head and tail entities are associated with their own projection matrices. Unlike STransE, the TranSparse model 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 sharing statistic regularities between relation projection matrices and alleviates data sparsity issue.

DISTMULT (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 while ER-MLP (Dong et al., 2014) and ProjE (Shi and Weninger, 2017) can be viewed as simplified versions of NTN. Such quadratic forms are also used to model entities and relations in KG2E (He et al., 2015), TransG (Xiao et al., 2016), ComplEx (Trouillon et al., 2016), TATEC (García-Durán et al., 2016), RSTE (Tay et al., 2017) and ANALOGY (Liu et al., 2017). In addition, the HoIE model (Nickel et al., 2016b) uses circular correlation—a compositional operator—which can be interpreted as a compression of the tensor product.

ConvE (Dettmers et al., 2017) and ConvKB (Nguyen et al., 2017) are based on convolutional neural networks. ConvE uses a 2D convolutional layer directly over head-entity and relation vector embeddings while ConvKB applies a convolutional layer over embedding triples. Unlike ConvE and ConvKB, the IRN model (Shen et al., 2017) uses a shared memory and recurrent neural network-based controller to implicitly model multi-step structured relationships.

Recent 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; Nguyen et al., 2016a; Durán and Niepert, 2017). 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 implausibility 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. In addition, rTransE (García-Durán et al., 2015), PTransE-ADD (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) and the PRUNED-PATHS model (Toutanova et al., 2016), each relation is a matrix and so it represents the relation path by matrix multiplication. The neighborhood mixture model TransE-NMM (Nguyen et al., 2016a) 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 the relational graph convolutional networks R-GCN (Schlichtkrull et al., 2017). Furthermore, Durán and Niepert (2017) proposed the KB_LRN framework to combine relational paths of length one and two with latent and numerical features.
2.3 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 recurrent neural network 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) and Mazumder and Liu (2017). Recently, Yang et al. (2017) have proposed a Neural Logic Programming (LP) framework to learning probabilistic first-order logical rules for KB reasoning, producing competitive link prediction performances. See other methods for learning from KBs and multi-relational data in Nickel et al. (2016a).

3 Evaluation tasks

Two standard tasks are proposed to evaluate embedding models for KB completion including: the entity prediction task, i.e. link prediction (Bordes et al., 2013), and the triple classification task (Socher et al., 2013).

Information about benchmark datasets for KB completion evaluation is given in Table 2. Commonly, datasets FB15k and WN18 (Bordes et al., 2013) are used for entity prediction evaluation, while datasets FB13 and WN11 (Socher et al., 2013) are used for triple classification evaluation. FB15k and FB13 are derived from the large real-world fact KB FreeBase (Bollacker et al., 2008). WN18 and WN11 are derived from the large lexical KB WordNet (Miller, 1995).

Toutanova and Chen (2015) noted that FB15k and WN18 are not challenging datasets because they contain many reversible triples. Dettmers et al. (2017) 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., 2017) 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. Note that when creating the FB13 and WN11 datasets, Socher et al. (2013) already filtered out triples from the test set if either or both of their head and tail entities also appear in the training set in a different relation type or order.

3.1 Entity prediction

3.1.1 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 ascending order of their implausibility score. This is called as the “Raw” setting protocol. Furthermore, 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, but is penalized by the “Raw” score, thus the “Filtered” setting provides a clearer view on the ranking performance.

| Dataset   | |E| |R| | #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           |
| FB13      | 75,043          | 13              | 316,232         | 5,908           | 23,733          |
| WN11      | 38,696          | 11              | 112,581         | 2,609           | 10,544          |
| 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 the experimental datasets. In both WN11 and FB13, each validation and test set also contains the same number of incorrect triples as the number of correct triples.
| Method | FB15k MR @10 | WN18 MR @10 | FB15k MR @10 | WN18 MR @10 |
|--------|--------------|-------------|--------------|-------------|
| SE (Bordes et al., 2011) | 162 39.8 - | 985 80.5 - | 273 28.8 - | 1011 68.5 - |
| Unstructured (Bordes et al., 2012) | 979 6.3 - | 304 38.2 - | 1074 4.5 - | 315 35.3 - |
| SME (Bordes et al., 2012) | 154 40.8 - | 533 74.1 - | 274 30.7 - | 545 65.1 - |
| TransH (Wang et al., 2014) | 87 64.4 - | 303 86.7 - | 212 45.7 - | 401 73.0 - |
| TransR (Lin et al., 2015b) | 77 68.7 - | 225 92.0 - | 198 48.2 - | 238 79.8 - |
| CTransR (Lin et al., 2015b) | 75 70.2 - | 218 92.3 - | 199 48.4 - | 231 79.4 - |
| KG2E (He et al., 2015) | 59 74.0 - | 331 92.8 - | 174 48.9 - | 342 80.2 - |
| TransD (Ji et al., 2015) | 91 77.3 - | 212 92.2 - | 194 53.4 - | 224 79.6 - |
| lppTransD (Yoon et al., 2016) | 78 78.7 - | 270 94.3 - | 195 53.0 - | 283 80.5 - |
| TransG (Xiao et al., 2016) | 98 79.8 - | 470 93.3 - | 203 52.8 - | 483 81.4 - |
| TransSparse (Ji et al., 2016) | 82 79.5 - | 211 93.2 - | 187 53.5 - | 223 80.1 - |
| TransSparse-DT (Chang et al., 2017) | 79 80.2 - | 221 94.3 - | 188 53.9 - | 234 81.4 - |
| TransF (Xie et al., 2017) | 65 81.0 - | 205 94.2 - | - - - | - - - |
| NTN (Socher et al., 2013) | - 41.4 0.25 - | 66.1 0.53 - | - - - | - - - |
| RESCAL (Nickel et al., 2011) [✓] | - 58.7 0.354 - | 92.8 0.890 - | - 0.189 - | - 0.603 - |
| TransE (Bordes et al., 2013) [✓] | - 74.9 0.463 - | 94.3 0.495 - | - 0.222 - | - 0.351 - |
| HolE (Nickel et al., 2016b) | - 73.9 0.524 - | 94.9 0.938 - | - 0.232 - | - 0.616 - |
| ComplEx (Trouillon et al., 2016) | - 84.0 0.692 - | 94.7 0.941 - | - 0.242 - | - 0.587 - |
| ANALOGY (Liu et al., 2017) | - 85.4 0.725 - | 94.7 0.942 - | - 0.253 - | - 0.657 - |
| ToruE (Ebisu and Ichise, 2018) | - 83.2 0.733 - | 95.4 0.947 - | - 0.256 - | - 0.619 - |
| STTransE (Nguyen et al., 2016b) | - 69 79.7 0.543 - | 206 93.4 0.657 - | 219 51.6 0.252 - | 217 80.9 0.469 - |
| ER-MLP (Dong et al., 2014) [♠] | - 81 80.1 0.570 - | 299 94.2 0.895 - | - - - | - - - |
| DISTMULT (Yang et al., 2015) [♠] | - 42 89.3 0.798 - | 655 94.6 0.797 - | - - - | - - - |
| ConvE (Dettmers et al., 2017) | - 64 87.3 0.745 - | 504 95.5 0.942 - | - - - | - - - |
| IRN (Shen et al., 2017) | - 38 92.7 - | 249 95.3 - | - - - | - - - |
| ProjE (Shi and Weninger, 2017) | - 34 88.4 - | - - - | 124 54.7 - | - - - |
| rTransE (García-Durán et al., 2015) | - 50 76.2 - | - - - | - - - | - - - |
| PTransE-ADD (Lin et al., 2015a) | - 58 84.6 - | - - - | 207 51.4 - | - - - |
| PTransE-RNN (Lin et al., 2015a) | - 92 82.2 - | - - - | 242 50.6 - | - - - |
| GAKE (Feng et al., 2016b) | - 119 64.8 - | - - - | 228 44.5 - | - - - |
| Gaffman (Niepert, 2016) | - 75 84.2 - | 352 93.9 - | - - - | - - - |
| Hiri (Liu et al., 2016) | - 70.3 0.603 - | 90.8 0.691 - | - - - | - - - |
| NeuralLP (Yang et al., 2017) | - 83.7 0.76 - | 94.5 0.94 - | - - - | - - - |
| R-GCN+ (Schlichtkrull et al., 2017) | - 84.2 0.696 - | 96.4 0.819 - | - 0.262 - | - 0.561 - |
| KB_LIN (Durán and Niepert, 2017) | - 44 87.5 0.794 - | - - - | - - - | - - - |
| NLFeat (Toutanova and Chen, 2015) | - 87.0 0.822 - | 94.3 0.940 - | - - - | - - - |
| TEKE_H (Wang and Li, 2016) | 108 73.0 - | 114 92.9 - | 212 51.2 - | 127 80.3 - |
| SSP (Xiao et al., 2017) | 82 79.0 - | 156 93.2 - | 163 57.2 - | 168 81.2 - |

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) since NTN was originally evaluated only for triple classification. [✓]: Results are taken from Nickel et al. (2016b). [♠]: Results are taken from Ravishankar et al. (2017). [♥]: Results are taken from Kadlec et al. (2017). In the first 26 rows, the best score is in **bold**, while the second and third best scores are in **underline**.

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), recent work also reports the mean reciprocal rank (MRR). In both “Raw” and “Filtered” settings, mean rank is always greater or equal to 1 and the lower mean rank indicates better entity prediction performance. MRR and Hits@10 scores always range
from 0.0 to 1.0, and higher score reflects better prediction result.

3.1.2 Main results

Table 3 lists entity prediction results of KB completion models on the FB15k and WN18 datasets. The first 26 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. The last 3 rows present results for models which make use of textual mentions derived from a large external corpus. The reasons why much work has been devoted towards developing triple-based models are mentioned by Nguyen et al. (2016b) as follows: (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.

Table 3 shows that the models using external corpus information or employing path information generally achieve better scores than the triple-based models that do not use such information. In terms of models not exploiting path or external information, on FB15k the IRN model (Shen et al., 2017) obtains highest scores, followed by DISTMULT (Yang et al., 2015), ProjE (Shi and Weninger, 2017) and ConvE (Dettmers et al., 2017). On WN18 top-4 triple-based models are ConvE, IRN, TorusE (Ebisu and Ichise, 2018) and ANALOGY (Liu et al., 2017).

Table 4 lists recent results on datasets FB15k-237 and WN18RR. On FB15k-237, by exploiting external textual mentions of entities, the Conv-E + Conv-DISTMULT model (Toutanova et al., 2015) produces the highest Hits@10 and MRR. In terms of models not exploiting external textual information, on FB15k-237, ER-MLP (Dong et al., 2014) can be considered as the best model to date, followed by ConvKB (Nguyen et al., 2017) and KBLRN (Durán and Niepert, 2017). On WN18RR, ConvKB can be considered as the best one, followed by ComplEx (Toutillon et al., 2016) and TransE (Bordes et al., 2013). Clearly, tables 3 and 4 show that TransE, despite of its simplicity, can produce very competitive results (by performing a careful grid search of hyper-parameters).

3.2 Triple classification

3.2.1 Task description

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 this task is to predict whether a
Table 5: Accuracy results (in %) for triple classification on WN11 (labeled as W11) and FB13 (labeled as F13) test sets. “Avg.” denotes the averaged accuracy. [*]: TransE results are taken from Nguyen et al. (2016a).

| Method                  | W11 | F13 | Avg. |
|-------------------------|-----|-----|------|
| CTransR (Lin et al., 2015b) | 85.7 | -   | -    |
| TransR (Lin et al., 2015b) | 85.9 | 82.5 | 84.2 |
| TransD (Ji et al., 2015)  | 86.4 | 89.1 | 87.8 |
| TEKE_H (Wang and Li, 2016) | 84.8 | 84.2 | 84.5 |
| TranSparse-S (Ji et al., 2016) | 86.4 | 88.2 | 87.3 |
| TranSparse-US (Ji et al., 2016) | 86.8 | 87.5 | 87.2 |
| NTN (Socher et al., 2013) | 70.6 | 87.2 | 78.9 |
| TransH (Wang et al., 2014) | 78.8 | 83.3 | 81.1 |
| SLogAn (Liang and Forbas, 2015) | 75.3 | 85.3 | 80.3 |
| KG2E (He et al., 2015) | 85.4 | 85.3 | 85.4 |
| Bilinear-COMP (Guu et al., 2015) | 77.6 | 86.1 | 81.9 |
| TransE-COMP (Guu et al., 2015) | 80.3 | 87.6 | 84.0 |
| TransR-FT (Feng et al., 2016a) | 86.6 | 82.9 | 84.8 |
| TransG (Xiao et al., 2016) | 87.4 | 87.3 | 87.4 |
| lppTransD (Yoon et al., 2016) | 86.2 | 88.6 | 87.4 |
| TransE (Bordes et al., 2013) [*] | 85.2 | 87.6 | 86.4 |
| TransE-NMM (Nguyen et al., 2016a) | 86.8 | 88.6 | 87.7 |
| TranSparse-DT (Chang et al., 2017) | 87.1 | 87.9 | 87.5 |

Table 5: Accuracy results (in %) for triple classification on WN11 (labeled as W11) and FB13 (labeled as F13) test sets. “Avg.” denotes the averaged accuracy. [*]: TransE results are taken from Nguyen et al. (2016a).

### 3.2.2 Main results

Table 5 presents the triple classification results of KB completion models on the WN11 and FB13 datasets. The first 6 rows report the performance of models that use TransE to initialize the entity and relation vectors. The last 12 rows present the accuracy of models with randomly initialized parameters. Note that there are higher results reported for NTN, Bilinear-COMP 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.

### 4 Conclusions and further discussion

This article presented a brief overview of embedding models of entity and relationships for KB completion. The article also provided update-to-date experimental results of the embedding models on the entity prediction and triple classification tasks on benchmark datasets FB15k, WN18, FB15k-237, WN18RR, FB13 and WN11.

Dozens of embedding models have been proposed for KB completion, so it is worth to further explore these models for a new application where we could formulate its corresponding data into triples. For example of an interesting application, Vu et al. (2017) extended the STransE model (Nguyen et al., 2016b) for a search personalization task in information retrieval, to model user-oriented relationships between submitted queries and documents returned by search engines.

### 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.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 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 *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. pages 1247–1250.
- 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.
- 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.
- Andrew Carlson, Justin Betteridge, Bryan Kissiel, Burr Settles, Jr. Estevam R. Hruschka, 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.

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.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2017. Convolutional 2D Knowledge Graph Embeddings. arXiv preprint abs/1707.01476.

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 Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pages 601–610.

Alberto García-Durán and Mathias Niepert. 2017. KBLRN: End-to-end learning of knowledge base representations with latent, relational, and numerical features. arXiv preprint abs/1709.04676.

Sourav Dutta and Gerhard Weikum. 2015. Cross-Document Co-Reference Resolution using Sample-Based Clustering with Knowledge Enrichment. Transactions of the Association for Computational Linguistics 3:15–28.

Takuma Ebisu and RyutaroIchise. 2018. TorusE: Knowledge Graph Embedding on a Lie Group. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open Question Answering over Curated and Extracted Knowledge Bases. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pages 1156–1165.

Christian 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 Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference. pages 557–560.

Jun Feng, Minlie Huang, Yang Yang, and xiaoyan zhu. 2016b. GAKE: Graph Aware Knowledge Embedding. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 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 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.

Matt Gardner, Partha P. Talukdar, Jayant Krishnamurthy, and Tom M. Mitchell. 2014. Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pages 397–406.

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.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge Base Completion: Baselines Strike Back. In Proceedings of the 2nd Workshop on Representation Learning for NLP. pages 69–74.

Jayant Krishnamurthy and Tom Mitchell. 2012. Weakly Supervised Training of Semantic Parsers. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and
Mathias Niepert. 2016. Discriminative Gaifman Models. In *Advances in Neural Information Processing Systems* 29, pages 3405–3413.

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.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 192–199.

Srinivas Ravishankar, Chandrasah, and Partha Pratim Talukdar. 2017. Revisiting Simple Neural Networks for Learning Representations of Knowledge Graphs. In *Proceedings of the 6th Workshop on Automated Knowledge Base Construction*.

Michael Sejr Schlükturrill, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling Relational Data with Graph Convolutional Networks. *arXiv preprint abs/1703.06103*.

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 *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 57–68.

Baoxu Shi and Tim Weninger. 2017. ProjE: Embedding Projection for Knowledge Graph Completion. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

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.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Falk Brauer. 2017. Random Semantic Tensor Ensemble for Scalable Knowledge Graph Link Prediction. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 751–760.

Kristina Toutanova and Danqi Chen. 2015. Observed Versus Latent Features for Knowledge Base and Text Inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoi-fung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing Text for Joint Embedding of Text and Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoi-fung Poon, and Chris Quirk. 2016. Compositional Learning of Embeddings for Relation Paths in Knowledge Base and Text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1434–1444.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 39th European Conference on Information Retrieval*, pages 598–604.

Quan Wang, Jing Liu, Yuanfei Luo, Bin Wang, and Chin-Yew Lin. 2016. Knowledge Base Completion via Coupled Path Ranking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1308–1318.

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.

Zhigang Wang and Juan-Zi Li. 2016. Text-Enhanced Representation Learning for Knowledge Graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1293–1299.

Zhuooyu Wei, Jun Zhao, and Kang Liu. 2016. Mining Inference Formulas by Goal-Directed Random Walks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 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 *Proceedings of the 23rd International Conference on World Wide Web*, 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 *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2316–2325.
Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2017. SSP: semantic space projection for knowledge graph embedding with text descriptions. In Proceedings of the 31st AAAI Conference on Artificial Intelligence.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An Interpretable Knowledge Transfer Model for Knowledge Base Completion. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pages 950–962.

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.

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In Advances in Neural Information Processing Systems 30, pages 2316–2325.

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 Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pages 907–916.