Article

Learning Translation-Based Knowledge Graph Embeddings by N -Pair Translation Loss

Hyun-Je Song ¹ , A-Yeong Kim ² and Seong-Bae Park ^{3,*}

¹ Department of Information Technology, Jeonbuk National University, Jeonju 54896, Korea; hyunje.song@jbnu.ac.kr

² School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea; aykim@sejong.knu.ac.kr

³ Department of Computer Science and Engineering, Kyung Hee University, Youngin 17104, Korea

* Correspondence: sbpark71@khu.ac.kr

Received: 8 May 2020; Accepted: 4 June 2020; Published: 7 June 2020

Abstract: Translation-based knowledge graph embeddings learn vector representations of entities and relations by treating relations as translation operators over the entities in an embedding space. Since the translation is represented through a score function, translation-based embeddings are trained in general by minimizing a margin-based ranking loss, which assigns a low score to positive triples and a high score to negative triples. However, this type of embedding suffers from slow convergence and poor local optima because the loss adopts only one pair of a positive and a negative triple at a single update of learning parameters. Therefore, this paper proposes the N -pair translation loss that considers multiple negative triples at one update. The N -pair translation loss employs multiple negative triples as well as one positive triple and allows the positive triple to be compared against the multiple negative triples at each parameter update. As a result, it becomes possible to obtain better vector representations rapidly. The experimental results on link prediction prove that the proposed loss helps to quickly converge toward good optima at the early stage of training.

Keywords: knowledge graph embeddings; translation-based knowledge graph embeddings; N -pair translation loss; negative sampling; multiple negative triples

1. Introduction

Knowledge graph embedding aims at learning the representation of a knowledge graph by embedding the knowledge graph into a low dimensional vector space [1]. For a given knowledge graph expressed as a set of knowledge triples where each triple is composed of a relation (r) and two entities (h and t), knowledge graph embedding finds vector representations of h , t , and r by considering the structure of the knowledge graph. Since a knowledge graph is regarded as a key resource, various embedding models have been proposed [2,3] and have been applied to a number of applications such as entity disambiguation [4], relation extraction [5,6], and question answering [7,8].

Translation-based knowledge graph embedding is one of the embedding models that finds vector representations of h , t , and r by compelling the vector of t to be close to the sum of the vectors of h and r [9]. Several variants have been proposed by modifying a score function to find better vector representations [10–14]. These translation-based embeddings are usually trained by minimizing the margin-based ranking loss over the knowledge graph. That is, the scores of positive triples are forced to be lower than those of negative triples. In order to adopt the margin-based ranking loss, negative triples against the knowledge graph are required, but knowledge graphs contain only positive triples. Thus, negative triples are prepared by replacing the head or the tail of positive triples randomly under the closed-world assumption [9].

Even if it looks simple to train the translation-based embeddings with the margin-based ranking loss, the embeddings suffer from slow convergence and poor local optima because margin-based ranking losses consider only one negative triple. Thus, most previous studies have focused mainly on the quality of negative triples to avoid the cases where negative triples contribute little to finding vector representations [15–18]. That is, they have developed novel negative sampling methods to generate hard negative triples under the adversarial learning framework [19] or using the caching framework. However, these studies require extra modules, and thus the number of parameters to be optimized increases. Furthermore, the cost of sampling is expensive and would increase if the size of the knowledge graph is huge.

If minibatch size is one while training translation-based knowledge graph embeddings with the margin-based ranking loss, the positive triple in the minibatch is compared with only one negative triple at a single update of learning parameters. On the other hand, if the positive triple is compared against multiple negative triples and is validated from all the negative triples at each update, the learning model will converge faster and find better local optima. Therefore, this paper proposes a simple but effective learning method based on a new loss function to incorporate multiple negative triples in training translation-based embeddings. The new loss function is motivated by the N -pair loss [20] that optimizes identifying a positive example from $N-1$ negative examples in metric learning. The proposed N -pair translation loss takes one positive triple as well as multiple negative triples and optimizes not only minimizing the score of the positive triple to satisfy constraints of translation-based knowledge graph embeddings but also maximizing the differences between the score of the positive triple and those of the negative triples. Since the proposed method interacts with multiple negative triples in each update, it results in better vector representations rapidly. The effectiveness of the proposed method is verified through extensive experiments. The experimental results prove empirically that the translation-based embeddings by the proposed loss converge fast and produce better vector representations than those by the margin-based ranking loss.

The rest of this paper is organized as follows. Section 2 surveys previous work to solve the problems in training translation-based knowledge graph embeddings. Section 3 introduces translation-based knowledge graph embeddings, and Section 4 proposes a new loss function for considering multiple negative triples. Section 5 presents the experimental settings and results. Finally, Section 6 concludes this paper with future research directions.

2. Related Work

Knowledge graph embedding has been studied steadily for the past couple of years since well-developed knowledge graphs have become available [21–24]. Among various embedding models, translation-based embedding has become one of the widely-used approaches to knowledge graph embedding due to its simplicity and efficiency. Since TransE [9] embeds entities and relations in the same vector space, several variants such as TransH [13], TransR [12], TransD [10], StransE [25], TransG [14], TransE-RS [26], TransAt [27], KGLG [28], and NTransGH [29] have been developed by extending or modifying TransE. Detailed explanations for translation-based variants can be found in the studies of Nickel et al. [2] and Wang et al. [3].

Even if translation-based knowledge graph embeddings yield promising results, they suffer from slow convergence and poor performance compared to other knowledge graph embeddings [16]. These problems of translation-based embeddings arise partially from the fact that the margin-based ranking loss used for training translation-based embeddings adopts only one negative triple per each update of learning parameters and some of the negative triples in which quality is low contribute little to training the embeddings, and this is called the *zero loss* problem. As a result, some previous studies focused on alleviating the zero loss problem by improving the quality of negative triples at the moment when the negative triples are generated [15–17,30]. One direction to improve the quality is to adopt the adversarial training framework [19] for training the knowledge graph embeddings [15,16]. In this direction, a generator calculates a probability distribution over a set of candidate negative

triples and provides a high-quality negative triple to a discriminator. Then, the discriminator, which is usually trained with the margin-based ranking loss, receives the negative triple from the generator as well as the positive triple and calculates a score of each triple. While the discriminator is trained with a marginal loss between positive and negative triples, the generator is trained using rewards from the discriminator. Another direction is to track the losses of negative triples and generate a difficult negative triple according to a loss distribution [17,30]. Although the studies above can generate high-quality negative triples, they require extra modules such as a generator or cache, and thus the number of parameters to be trained increases.

3. Translation-Based Knowledge Graph Embeddings

Translation-based embeddings project entities or/and relations between entities onto an embedding space by treating relations as translation operators of entities on the space. Assume that a knowledge graph $\mathcal{G} = \{(h_i, r_i, t_i)\}_{i=1}^{|\mathcal{G}|}$, which is represented as a set of triples (h, r, t) , is given where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. Here, $|\mathcal{G}|$ is the number of triples, $\mathcal{E}$ is a set of entities, and $\mathcal{R}$ is that of relations between entities. Then, translation-based knowledge graph embeddings find vector representations $\mathbf{h}$, $\mathbf{t}$, and $\mathbf{r}$ for entities h , t and relation r by enforcing $\mathbf{t}$ to be a sum of $\mathbf{h}$ and $\mathbf{r}$.

Among various translation-based knowledge graph embeddings, TransE [9] is the most representative and simplest method, which projects both entities and relations in the same vector space. That is, the vector representations $\mathbf{h}$, $\mathbf{t}$, and $\mathbf{r}$ are represented in the vector space $\mathbb{R}^k$, where k is a dimension of vector representations. Then, for every triple (h, r, t) in a knowledge graph $\mathcal{G}$, TransE forces $\mathbf{h} + \mathbf{r}$ to be close to $\mathbf{t}$. This translation is represented by the following score function

$$s(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{\ell_1/\ell_2}, \quad (1)$$

where ℓ_1 and ℓ_2 indicate L1 and L2 norm, respectively. Note that the smaller the score $s(\cdot)$ is, the better the triple (h, r, t) is.

To learn these vector representations, the margin-based ranking loss between a positive triple (h_p, r_p, t_p) and a negative triple (h_n, r_n, t_n) is defined as

$$\begin{aligned} \mathcal{L}_{one}(\{(h_p, r_p, t_p), (h_n, r_n, t_n)\}) \\ = [(s(h_p, r_p, t_p) + \gamma - s(h_n, r_n, t_n))]_{+}, \end{aligned} \quad (2)$$

where $[x]_{+} = \max(0, x)$ and γ is a margin. Then, TransE minimizes the margin-based ranking risk over $\mathcal{G}$ given as

$$\mathcal{R}(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{(h,r,t) \in \mathcal{G}} \sum_{(h',r,t') \in \mathcal{G}'_{(h,r,t)}} \mathcal{L}_{one}(\{(h, r, t), (h', r, t')\}). \quad (3)$$

Here, $\mathcal{G}'$ is a set of negative or corrupted triples. This set is usually constructed artificially because a knowledge graph contains only correct triples. One simple way to make $\mathcal{G}'$ is to replace the entities in the positive or correct triples with another entities in $\mathcal{E}$ following the closed-world assumption. That is, $\mathcal{G}'_{(h,r,t)}$ can be prepared by

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

The learning of TransE is carried out by stochastic gradient descent (SGD) in minibatch mode over the possible triples in $\mathcal{G}$. While optimizing the vector representations using Equation (3), some additional constraints can be enforced to embeddings of entities in order to avoid the embeddings from diverging unlimitedly.

Several variants of TransE are obtained by changing the score function in Equation (1). For instance, TransR [12] maps the entity representations onto a different relation space for every relation. Thus, its score function is

$$s(h, r, t) = \|\mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t}\|_{\ell_1/\ell_2},$$

where $\mathbf{M}_r$ is a projection matrix that projects entities from an entity space to the space of relation r . Another is TransD [10], which maps entity representations onto different vectors in relation spaces according to entity and relation types. Thus, the score function of TransD is

$$s(h, r, t) = \|\mathbf{M}_{rh} \mathbf{h} + \mathbf{r} - \mathbf{M}_{rt} \mathbf{t}\|_{\ell_1/\ell_2},$$

where $\mathbf{M}_{rh}$ and $\mathbf{M}_{rt}$ are entity-relation specific mapping matrices. These two variants also can be trained using the margin-based ranking loss as in TransE.

4. Considering Multiple Negative Triples though N -Pair Translation Loss

Even if translation-based knowledge graph embeddings with the margin-based ranking loss in Equation (2) learn vector representations of entities and relations, they suffer from slow convergence and poor local optima of parameters. These problems arise mostly from the fact that a positive triple is compared with only one negative triple at a single update of the parameters. Figure 1 illustrates the progress of training TransE with a positive triple (h, r, t) and negative triples (h, r, t'_i) obtained by replacing the tail entity t . The bold vector in this figure represents a vector representation of the relation r in the positive triple, and it is trained to be distinguished from the relation vector representations of the negative triples represented as dotted vectors. When the margin-based ranking loss is used, as in Figure 1a, the only thing that is guaranteed is that the bold vector is better than one dotted vector because the positive triple is compared with only one negative triple. This implies that the trained vector representations using the margin-based ranking loss could be far from (local) optima at the early stage of training, even if the vector representations could reach (local) optima after iterating over a number of randomly-sampled negative triples. Furthermore, this type of learning could be unstable depending on the selected negative triple.

Figure 1. How to find vector representations of entities and relations in TransE. Margin-based ranking loss (a) adopts only one negative triple at a time, while N -pair translation loss (b) considers multiple negative triples at once.

On the other hand, if the positive triple is compared against multiple negative triples at once as in Figure 1b, the bold vector should be distinguished from all dotted vectors at the same time. In this figure, the vector representations of tail entities t_i in the negative triples are distributed over the embedding space, and thus the bold vector can be trained more accurately. In other words, learning with multiple negative triples is stable compared to that optimizing the margin-based ranking loss, and the convergence toward (local) optima is also achieved at the early stage of training.

In order to consider multiple negative triples, this paper proposes *N-pair translation loss*, which takes one positive triple and multiple negative triples for each parameter update. The proposed loss is calculated by comparing the positive triple with the multiple negative triples at the same time. Assume that there are $N+1$ triples of one positive triple (h_p, r_p, t_p) and N negative triples $\{(h_i, r_i, t_i)\}_{i=1}^N$. Then, the *N-pair translation loss* is defined as

$$\begin{aligned} \mathcal{L}_{n\text{pair}}(\{(h_p, r_p, t_p), \{(h_i, r_i, t_i)\}_{i=1}^N\}) \\ = \log \left(1 + \sum_{i=1}^N \exp(s(h_p, r_p, t_p) - s(h_i, r_i, t_i)) \right), \end{aligned}$$

where $s(\cdot)$ is a score function that is determined by the type of translation-based embedding. The proposed loss first tries to minimize the score of the positive triple to satisfy the constraints of translation-based knowledge graph embeddings. It also maximizes the differences between the score of the positive triple and those of the negative triples. That is, the *N-pair translation loss* satisfies these two constraints at the same time. In addition, since it considers all negative triples at once, the learning parameters move toward better optima. Note that the special case of this loss when $N = 1$ is the margin-based ranking loss [20].

Assume that there is NG , a negative triple generator that receives a set of all possible negative triples $\mathcal{G}'_{(h,r,t)}$ and the number of generated triples N . Then, NG generates $\mathcal{G}'^N_{(h,r,t)}$, a set of N negative triples. From $\mathcal{G}$ and $\mathcal{G}'^N_{(h,r,t)}$, the vector representations of entities and relations are found by minimizing the following *N-pair translation risk* over $\mathcal{G}$.

$$\mathcal{R}(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{(h,r,t) \in \mathcal{G}} \mathcal{L}_{n\text{pair}}(\{(h, r, t), \mathcal{G}'^N_{(h,r,t)}\}). \quad (4)$$

Minimizing the *N-pair translation risk* is also carried out by SGD in minibatch mode over the possible triples in $\mathcal{G}$. If the minibatch size of SGD is B , then $\mathcal{L}_{n\text{pair}}$, the *N-pair translation loss*, uses $B \times (N + 1)$ triples at one update. Since the generator NG has to generate N negative triples, it takes more time than the margin-based ranking loss. However, this cost can be alleviated by using offset-based negative sampling algorithm [31] with a parallel implementation [32]. Algorithm 1 summarizes our proposed method.

Algorithm 1: Learning Translation-based Knowledge Graph Embeddings by Considering Multiple Negative Triples

Input : Knowledge graph $\mathcal{G} = \{(h_i, r_i, t_i)\}_{i=1}^{|\mathcal{G}|}$, entity and relation sets $\mathcal{E}$ and $\mathcal{R}$

Output: Trained knowledge graph embeddings θ

repeat

$g_\theta \leftarrow 0$. // gradient of parameter θ

 Sample a mini-batch of data $\mathcal{G}_{\text{batch}}$ from $\mathcal{G}$.

for $(h, r, t) \in \mathcal{G}_{\text{batch}}$ **do**

$\mathcal{G}'_{(h,r,t)} \leftarrow NG((h, r, t))$. // generate N negative samples

$g_\theta \leftarrow g_\theta + \nabla_\theta(\mathcal{L}_{n\text{pair}}(\{(h, r, t), \mathcal{G}'_{(h,r,t)}\}))$. // accumulate gradients

end

$\theta \leftarrow \theta + \eta g_\theta$. // update parameters

until convergence;

5. Experiments

5.1. Experimental Settings

5.1.1. Dataset

Two well-known knowledge graphs are used for our experiments: WordNet [33] and Freebase [21]. From these two knowledge graphs, two data sets are extracted for the evaluation of the proposed method. The data sets are WN18RR and FB15K-237. WN18RR are derived from WordNet while FB15K-237 are from Freebase. WN18RR and FB15K-237 are generated by removing near-duplicate and inverse-duplicate relations from WN18 and FB15k, respectively. These data sets have been commonly used in many previous studies [9,10,34], and the simple statistics on them is given in Table 1.

Table 1. Simple statistics on data sets.

Dataset	WN18RR	FB15K-237
# Entities	93,003	14,541
# Relations	11	237
# Training Triples	86,835	272,115
# Validation Triples	3034	17,535
# Test Triples	3134	20,466

5.1.2. Evaluation Task and Protocol

The effectiveness of the proposed method is shown through a *link prediction* [9] task, which aims at predicting a missing entity h or t when there exists a missing entity in a knowledge graph. The link prediction is evaluated with two metrics: Hits@10 and mean reciprocal ranking (MRR). The Hits@10 is the proportion of correct triples ranked in the top 10, while the mean reciprocal ranking measures the average reciprocal rank of all correct entities. To avoid underestimating the performances of embeddings, we use the “Filter” evaluation setting [9]. That is, the triples that are already included within training, validation, and test sets are filtered out before ranking.

Bernoulli sampling is used for generating negative triples following the previous study of Wang et al. [13]. The reason why the sampling is adopted is that it helps reducing false negative triples since it replaces head or tail entities with different probabilities for one-to-many, many-to-one, and many-to-many relations.

5.1.3. Implementation

Two well-known translation-based knowledge graph embeddings are adopted to compare the proposed loss with the margin-based ranking loss. The adopted embeddings are TransE [9] and TransD [10]. To minimize both the margin-based ranking risk (Equation (3)) and the N -pair translation risk (Equation (4)), we use Adam optimizer [35] with learning rate 0.001 and two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Minibatch size is set to the number of the training triples divided by 100, and the epoch limit is set to 1000. The Xavier uniform initializer [36] is used to initialize the vector representations of entities and relations. In order to find other hyper-parameters, we performed an exhaustive grid search over the following settings: a dimension of vector representations $k \in \{25, 50, 100\}$, and a norm for the score function $L \in \{\ell_1, \ell_2\}$. The best hyper-parameters are tuned by the Hits@10 metric on a validation set. As a result, the dimension size is set to 50 for WN18RR, while it is 100 for FB15K-237. ℓ_1 is used as the norm for a score function in all data sets.

5.2. Experimental Results

We first examine the effectiveness of negative triples. Thus, the change of performances is investigated according to N , the number of negative triples. The result is shown in Figure 2. Figure 2a,b

are the performances of TransE on WN18RR and FB15K-237 with a various number of negative triples, while Figure 2c,d show those of TransD. In all figures, the X-axis is the number of epochs, and the Y-axis represents the Hits@10. Here, '1' (expressed as the blue curve) implies that only one negative triple is considered in training the embedding models. Therefore, the embedding model with '1' is equivalent to that trained with the margin-based ranking loss. Similarly, $N(N > 1)$ means N negative triples are used. Thus, the performances of N are obtained using the N -pair translation loss. As these figures show, all non-blue curves are always above the blue curve regardless of epochs. That is, the performances of the N -pair translation loss are higher than that of the margin-based ranking loss. This proves that considering multiple negative triples in every parameter update leads to significant performance improvement over adopting one negative triple.

Figure 2. Hit@10 performances according to various numbers of negative triples. (a) WN18RR—TransE, (b) FB15K-237—TransE, (c) WN18RR—TransD, (d) FB15K-237—TransD.

Another thing to note in Figure 2 is that the Hits@10's of the N -pair translation loss rise up much faster than that of the margin-based ranking loss (the blue curve). That is, the N -pair translation loss converges faster than the margin-based ranking loss. In more detail, the margin-based ranking loss does not converge until 1000 epochs in all figures, while the N -pair translation loss converges to a stable Hits@10 before 400 epochs. This verifies that the N -pair translation loss is helpful for fast convergence. The convergence speed is an important feature for translation-based knowledge graph embeddings because a number of other knowledge graph embedding models use a translation-based model as its base model. That is, when a new knowledge graph is given, such embedding models require fast vector representation of the knowledge graph as a prerequisite. Therefore, fast convergence of a translation-based knowledge graph embedding by the proposed N -pair translation loss is worthy.

The last observation is that the optimal number of negative triples is not always the largest number. For the WN18RR data set (Figure 2a,c), the best performance is obtained when 10 negative triples are used. On the other hand, adopting 200 negative triples achieves the best performance for the FB15K-237 data set (Figure 2b,d). This is because negative triples are generated by random

sampling of entities. All negative triples are not tightly related with the positive triple, and thus some of them behave as noise to learning the embeddings. This result is consistent with the previous study by Trouillon et al. [37].

Table 2 presents the experimental results on link prediction for all data sets. The proposed translation-based embeddings by the N -pair translation loss are compared with the state-of-the-art knowledge graph embeddings that use their own negative sampling method. The compared embeddings include KBGAN [15], IGAN [16], and NSCaching [17], and their performances are referred from their reports since we use the same data sets. ‘Margin’ in this table implies the translation-based embedding with the margin-based ranking loss. The proposed N -pair translation loss for TransE achieves 53.0 and 50.5 of Hits@10 on WN18RR and FB15K-237, respectively, which are much higher than those of ‘Margin’. The performance difference between the proposed loss and the margin loss is up to 15 Hits@10. The performances for TransD are similar to those of TransE, where Hits@10 are 49.4 and 50.3 on WN18RR and FB15K-237, respectively. These performances are also superior to those of the margin-based ranking loss. This implies that the use of multiple negative triples at each update is a good way to achieve better performance.

Table 2. Experimental results on link prediction.

Embeddings	Dataset	WN18RR		FB15K-237	
	Metric	MRR	Hits10	MRR	Hits10
TransE	Margin	14.5	37.6	27.0	43.6
	N -pair	23.7	53.0	32.6	50.5
	KBGAN [15]	21.0	47.9	27.8	45.3
	NSCaching [17]	20.5	47.4	30.0	47.4
TransD	Margin	15.0	38.7	25.4	43.3
	N -pair	22.6	49.4	31.8	50.3
	KBGAN [15]	27.7	45.8	21.5	46.9
	NSCaching [17]	20.1	48.4	28.8	48.3

Even when compared to the current state-of-the-art knowledge graph embeddings, the proposed method outperforms them on WN18RR and FB15K-237. Since the proposed loss interacts with *multiple* and *diverse* negative triples at each parameter update, it could cope with various relations in WN18RR and FB15K-237. As a result, the proposed loss could achieve higher performance in these data sets. Furthermore, the proposed method does not require any additional resource while the previous methods need an extra module such as a high-cost generator or cache. In addition, since the previous methods use TransE as a part of their model, they are heavy to train. On the other hand, the proposed method is light to train, because it just modifies the loss function of legacy translation-based embeddings. From all these results, it can be inferred that the N -pair translation loss is helpful in learning vector representations for translation-based embeddings.

6. Conclusions

This paper has proposed simple and effective learning for translation-based knowledge graph embeddings by introducing a new loss function. The proposed loss function receives multiple negative triples per one positive triple and allows the positive triple to be compared against the multiple negative triples at a single parameter update. Therefore, learning vector representations with the loss can utilize the information obtained by interacting with multiple negative triples. The experimental results have shown that the proposed loss function does not achieve only fast convergence, but also produces better vector representations. (Our code is available at <https://github.com/songhyunje/kge>).

Author Contributions: Conceptualization, H.-J.S. and A.-Y.K.; methodology, H.-J.S. and A.-Y.K.; visualization, A.-Y.K.; validation, H.-J.S., A.-Y.K. and S.-B.P.; funding acquisition, H.-J.S. and S.-B.P.; writing—original draft preparation, H.-J.S.; writing—review and editing, H.-J.S. and S.-B.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This paper was supported by research funds for newly appointed professors of Jeonbuk National University in 2019 and Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y. Learning Structured Embeddings of Knowledge Bases. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011; pp. 301–306.
- Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* **2016**, *104*, 11–33. [\[CrossRef\]](#)
- Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [\[CrossRef\]](#)
- Huang, H.; Heck, L.; Ji, H. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv* **2015**, arXiv:1504.07678
- Riedel, S.; Yao, L.; McCallum, A.; Marlin, B.M. Relation Extraction with Matrix Factorization and Universal Schemas. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–15 June 2013; pp. 74–84.
- Weston, J.; Bordes, A.; Yakhnenko, O.; Usunier, N. Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Washington, DC, USA, 18–21 October 2013; pp. 1366–1371.
- Bordes, A.; Chopra, S.; Weston, J. Question Answering with Subgraph Embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 615–620.
- Cui, W.; Xiao, Y.; Wang, H.; Song, Y.; Hwang, S.W.; Wang, W. KBQA: Learning Question Answering over QA Corpora and Knowledge Bases. *Proc. VLDB Endow.* **2017**, *10*, 565–576. [\[CrossRef\]](#)
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the Advances in Neural Information Processing Systems 26, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 2787–2795.
- Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. 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, Beijing, China, 26–31 July 2015; pp. 687–696.
- Jia, Y.; Wang, Y.; Lin, H.; Jin, X.; Cheng, X. Locally Adaptive Translation for Knowledge Graph Embedding. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 992–998.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
- Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
- Xiao, H.; Huang, M.; Zhu, X. TransG: A Generative Model for Knowledge Graph Embedding. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 2316–2325.
- Cai, L.; Wang, W.Y. KBGAN: Adversarial Learning for Knowledge Graph Embeddings. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 1470–1480.

16. Wang, P.; Li, S.; Pan, R. Incorporating GAN for Negative Sampling in Knowledge Representation Learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 2005–2012.
17. Zhang, Y.; Yao, Q.; Shao, Y.; Chen, L. NSCaching: Simple and Efficient Negative Sampling for Knowledge Graph Embedding. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering, Macau, China, 8–12 April 2019; pp. 614–625.
18. Dash, S.; Gliozzo, A. Distributional Negative Sampling for Knowledge Base Completion. *arXiv* **2019**, arXiv:1908.06178.
19. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
20. Sohn, K. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
21. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Wuhan, China, 14–16 September 2008; pp. 1247–1250.
22. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmman, T.; Sun, S.; Zhang, W. 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, New York, NY, USA, 24–27 August 2014; pp. 601–610.
23. Fader, A.; Soderland, S.; Etzioni, O. Identifying Relations for Open Information Extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Scotland, UK, 27–31 July 2011; pp. 1535–1545.
24. Mitchell, T.; Cohen, W.; Hruschka, E.R., Jr.; Talukdar, P.P.; Betteridge, J.; Carlson, A.; Mishra, B.D.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; et al. Never Ending Learning. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2302–2310.
25. Nguyen, D.Q.; Sirts, K.; Qu, L.; Johnson, M. STransE: A novel embedding model of entities and relationships in knowledge bases. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 460–466.
26. Zhou, X.; Zhu, Q.; Liu, P.; Guo, L. Learning Knowledge Embeddings by Combining Limit-Based Scoring Loss. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1009–1018.
27. Qian, W.; Fu, C.; Zhu, Y.; Cai, D.; He, X. Translating Embeddings for Knowledge Graph Completion with Relation Attention Mechanism. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4286–4292.
28. Ebisu, T.; Ichise, R. Generalized Translation-Based Embedding of Knowledge Graph. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 941–951. [[CrossRef](#)]
29. Zhu, Q.; Zhou, X.; Zhang, P.; Shi, Y. A neural translating general hyperplane for knowledge graph embedding. *J. Comput. Sci.* **2019**, *30*, 108–117. [[CrossRef](#)]
30. Lei, J.; Ouyang, D.; Liu, Y. Adversarial Knowledge Representation Learning Without External Model. *IEEE Access* **2019**, *7*, 3512–3524. [[CrossRef](#)]
31. Han, X.; Cao, S.; Lv, X.; Lin, Y.; Liu, Z.; Sun, M.; Li, J. OpenKE: An Open Toolkit for Knowledge Embedding. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 31 October–4 November 2018; pp. 139–144.
32. Zhang, D.; Li, M.; Jia, Y.; Wang, Y.; Cheng, X. Efficient Parallel Translating Embedding for Knowledge Graphs. In Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, 23–26 August 2017; pp. 460–468.
33. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [[CrossRef](#)]
34. Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, Beijing, China, 31 July 2015; pp. 57–66.

35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
36. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
37. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the 33th International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2071–2080.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).