UNDERSTANDING NEGATIVE SAMPLING IN KNOWLEDGE GRAPH EMBEDDING

Jing Qian[1, 2], Gangmin Li[1], Katie Atkinson[2] and Yong Yue[1]

[1]Department of Intelligent Science, School of Advanced Technology,
Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu Province, China
[2]Department of Computer Science, University of Liverpool, Liverpool, United Kingdom

ABSTRACT

Knowledge graph embedding (KGE) is to project entities and relations of a knowledge graph (KG) into a low-dimensional vector space, which has made steady progress in recent years. Conventional KGE methods, especially translational distance-based models, are trained through discriminating positive samples from negative ones. Most KGs store only positive samples for space efficiency. Negative sampling thus plays a crucial role in encoding triples of a KG. The quality of generated negative samples has a direct impact on the performance of learnt knowledge representation in a myriad of downstream tasks, such as recommendation, link prediction and node classification. We summarize current negative sampling approaches in KGE into three categories, static distribution-based, dynamic distribution-based and custom cluster-based respectively. Based on this categorization we discuss the most prevalent existing approaches and their characteristics. It is a hope that this review can provide some guidelines for new thoughts about negative sampling in KGE.

KEYWORDS

Negative Sampling, Knowledge Graph Embedding, Generative Adversarial Network

1. INTRODUCTION

A knowledge graph (KG) refers to a network in which nodes are real entities or abstract concepts and edges are their in-between relations. Many KGs have been well developed, such as NELL [1], Freebase [2] and YAGO [3]. They store and tell ground-truth facts in the triple form, expressed as (head entity, relation, tail entity) or (subject, predicate, object). Knowledge graph embedding (KGE) aims to encode components of a KG into a low-dimensional continuous vector space to support the downstream graph operations and knowledge reuse. A variety of KGE models have been successively proposed and deployed in recent years.

Inspired by word embedding [4], people turned to distributed representation of entities and relations rather than discrete representation. One-hot encoding is broadly used to convert features or instances into vectors, it has great interpretability but incapable of capturing latent semantics since it is impossible to compute the similarity between orthogonal vectors. To overcome the problems associated with the one-hot encoding, more embedding techniques have been developed. In KGE, not only the conventional translational distance-based TransE [5], semantic matching-based RESCAL [6] but also the state-of-the-art attention-based KBAT [7] and GAATs [8], they are devoted to learning better knowledge representation and serving knowledge graph completion tasks.

DOI: 10.5121/ijaia.2021.12105

Different embedding models are embodied in their own defined scoring functions that return a score to measure the plausibility of a given triple. These models require certain amount of training and verification as in the routine. Noise contrastive estimation (NCE) [9]is one of the common methods to accelerate training. It transforms the density estimation into a binary classification problem by discriminating real samples from noise samples [10]. Mikolov et al. [4] simplifies NCE to negative sampling and applies it in word embedding to reduce computational complexity that results from large vocabulary size. KGE extends this strategy with the aim of ranking observed ("positive") instances higher than unobserved ("negative") ones to accomplish model training [11]. As seen in translational distance-based models [5, 12-15], they are optimized through partitioning scores of positives and negatives with an adaptive margin. Most KGs contain only ground-truth triples, for the sake of space efficiency. Following the thought of NCE to improve the training efficiency of KGE models, a large number of negative samples are required. Negative sampling thus becomes a critical point in knowledge representation learning. Uniform sampling [5, 13] is one of the most commonly used negative sampling approaches, it corrupts positive triples by replacing the head or tail entities with those that are uniformly sampled from the entity set of the KG. However, such generated negative triples are too easy to be discriminated and make little contribution towards the training for most of the time. Different from that sampling with equal probability in random uniform mode, Bernoulli sampling [12] applies different probabilities in head and tail replacement to address the issue of false negatives. KBGAN [16] and IGAN [17] are two typical GAN-based negative sampling approaches that take advantage of the generative adversarial network (GAN), they adversarially train the generator to provide better-quality negatives by applying a pre-trained KGE model as the discriminator. TransE-SNS [18] and NSCaching [19] attempt to gather candidate entities of negative sampling into custom clusters. Furthermore, enlightened by CKRL [20], NKRL [21] puts forward a confidence-aware negative sampling approach. Yang et al. [22] recently derives the general form of an effective negative sampling distribution, which is of pioneering significance. They are the first to deduce the correlation between positive and negative sampling distribution. Trouillon et al. [23] further studies the number of negatives generated for each positive triple, and elicits that fifty negative samples per positive is a good choice for balancing accuracy and duration.

In this paper, we summarize current negative sampling approaches in knowledge representation learning and sketch out them into three categories based on their sample source: sampling from static distribution, sampling from dynamic distribution and sampling from custom clusters respectively. A majority of researches about KGE focus on proposing new embedding models or evaluating their performance in downstream tasks, such as knowledge graph completion [24], question-answering [25] and recommendation [26]. We argue that despite the broad agreement that negative sampling is of great importance in the training of KGE models, it is under explored and needs more attention and efforts. In the representative surveys about knowledge representation learning [27, 28], negative sampling is mentioned but only in a short space. To the best of our knowledge, this paper is the first work to systematically and exhaustively overview existing negative sampling approaches in the field of KGE.

The remainder of this review is organized as follows. Section 2 provides a brief definitions and notations, and assumptions necessary for understanding the existing KGE models. A variety of embedding models that are proposed in knowledge representation learning are briefly covered in Section 3, these models are further elaborated according to our categorization schema in Section 4. Finally, we present our conclusion remarks and future research suggestions.

2. DEFINITIONS, NOTATIONS AND ASSUMPTIONS

All the negative sampling approaches are based on a simple KGE model. That is, in a standard KG, $\mathbb{E}$ represents the set of entities, $\mathbb{R}$ represents the set of relations. $\mathbb{D}^+$ and $\mathbb{D}^-$ are sets of the positive triples $\tau^{+} = (h, r, t)$ and the counterpart negative triples respectively. The following formula sets out the components of the set $\mathbb{D}^{-}$. In general cases, one KGE model can be explained by its own defined scoring function $f_r(h, t)$ where h and t belong to $\mathbb{E}$ and r belongs to $\mathbb{R}$. The relation $r$maps the head entity h to its tail entity t. The plausibility of each possible triple is measured by the scoring function. The higher the plausibility is, the more probability for the triple being a piece of truth.

$$\tau^{-} \in \mathbb{D}^{-}$$

$$\begin{aligned}\mathbb{D}^{-} = \{(h', r, t) | h' \in \mathbb{E} \wedge h' \neq h \wedge (h, r, t) \in \mathbb{D}^{+}\} \\ \cup \{(h, r, t') | t' \in \mathbb{E} \wedge t' \neq t \wedge (h, r, t) \in \mathbb{D}^{+}\} \\ \cup \{(h, r', t) | r' \in \mathbb{D} \wedge r' \neq r \wedge (h, r, t) \in \mathbb{D}^{+}\}\end{aligned}$$

KGE models are trained under the open world assumption (OWA) [29] or the closed world assumption (CWA) [30]. The CWA states that facts that are not observed in D^{+} are false, while the OWA is relaxed to assume that unobserved facts can be either missing or false. Most models prefer the OWA due to the incompleteness nature of KGs. The CWA has two main drawbacks, worse performance in downstream tasks and scalability issues caused by tremendous negative samples [27].

3. KGE Models

Knowledge graph embedding is also called knowledge representation learning that aims to embed triples (h, r, t) into a low-dimensional continuous vector space and take advantage of numerical representation that is processed by machine learning and deep learning models. Different KGE models encode latent semantics into the embedding vectors in different ways [27], which are reflected in the manually defined scoring functions f_r (h, t)that calculate the credibility scores for given triples. Generally, the Translational Distance-based and the Semantic Matching-basedare two mainstream types of KGE models. In addition, another two frameworks, basing on neural networks and incorporating additional information, have also been considered in recent years.

Translational distance-based models. The main idea behind the translation-based models is to measure the distance between the head entity and the tail entity after projecting the KG into the vector space. Inspired by translation invariance in word embedding vectors, TransE [5] considers the relation vector as a transition from the head to the tail, i.e. h + r ≈ t. The equation holds when (h, r, t) exists in the KG. The distance between h + r and t reflects the degree of confidence for the given triple. TransH [12] improves TransE to make it capable of modeling multiple relations, like “1-to-N”, “N-to-1” and “N-to-N”. Some other variants such as TransR [13], TransD [14] and TransG [15], they extend TransE by embedding entities into various spaces.

Semantic matching-based models. Compared to the translational distance-based models, semantic matching-based models focus on modeling the latent semantics embodied in vectorized entities and relations by means of matrix decomposition. RESCAL [6] is one of the most representative KGE models that define the scoring function based on matching semantics. The relations are encoded in the mapping matrix M_r to connect the head and the tail while the matrix product hM_rt measures the plausibility of triples. Furthermore, DistMult [31] simplifies RESCAL by limiting M_r to be a diagonal matrix, and ComplEx [23] extends DistMult to depict antisymmetric relations in the complex number field.

Neural network-based models. Applying neural networks in knowledge representation learning has also gained wide attention. MLP [32] feeds vectors of entities and relations into a fully-connected layer to capture implied semantics. ConvE [33] attempts to form its scoring function with 2D convolution. RSN [34] introduces a recurrent skip mechanism in order to benefit the embedding of KGs. In addition, KG-BERT [35] bases on Transformer (BERT) to integrate knowledge representation learning and natural language processing. In the emerging field of graph neural networks (GNNs), R-GCN [36] builds a graph convolutional network framework to encode the multi-relational data of KGs.

Auxiliary-dependent models. Apart from triples that compose KGs, some additional information can be incorporated as the auxiliary to enhance semantic representation learning. Guo et al. [37] takes the entity type into account and assumes that entities of the same type ought to be closer to each other in the embedded space. PTransE [38] attempts to model multi-hop relations using addition, multiplication and RNN rules so that the relation paths between entities can be reflected in the calculations among vectors. Besides, Wang et al. [39] considers text information and puts forward a joint model to accomplish the embedding process. Guo et al. [40] comes up with a rule-based KGE model by combining some rule information.

Negative sampling is a variation of NCE, that firstly proposed in the word2vec tool [41]. Knowledge representation learning follows this strategy. A majority of studies above focus on inventing novel embedding models, and adopt some random sampling approaches to provide negative training samples [27]. Till now, a few works have been devoted to improving the quality of negatives in embedding KGs. We outline these studies with the aim of gaining more attention to negative sampling. In addition, application scenarios and future trends about conventional and the state-of-the-art KGE models can be found in the representative surveys [27, 28, 42].

4. NEGATIVE SAMPLING

All the above models require negative samples during training. The thought of negative sampling was firstly raised in probabilistic neural models of language and labelled as importance sampling [43]. Mikolov et al. [4] emphasizes it as a simplified version of NCE [9] to benefit the training of word2vec. NCE is used to overcome the computational difficulty associated with probabilistic models of language since they involve evaluating partition functions by summing over all the words, which may be a huge vocabulary. Evolved from NCE, negative sampling transforms the difficult density estimation problem into a binary classification problem that distinguishes real samples from noise samples, which simplifies the computation and accelerates the training. Instead of normalizing the partition function into a probability distribution based on the entire vocabulary, separating the “true” samples from those that are sampled from the noise distribution is beneficial for asymptotically estimating the “true” distribution with high efficiency and low computational cost.

Graph representation learning is similar to language modeling when regarding nodes as the words and neighbors as the context. Negative sampling is also applied in KGE so that the knowledge representation can be learnt through discriminating positive triples from negative triples that are generated by perturbing the positive ones, rather than modeling conditional on all nodes. In KGE, poor or too obviously incorrect negative triples fail in facilitating the capture of latent semantics and easily bring about the zero loss problem. In contrast, high-quality negatives will ensure that the training smoothly moves on and the learnt knowledge representation performs better in a myriad of downstream tasks.

Recognising the importance and benefits of negative sampling, we systematically collect the existing sampling approaches, study them and most importantly, categorize them from three distinct perspectives, i.e. static distribution-based, dynamic distribution-based and custom cluster-based. Brief comments on the characteristics and pros and cons are provided.

4.1. Static Distribution-Based Sampling

Static distribution-based negative sampling approaches are commonly used because of their simplicity and efficiency. Static distribution contains uniform distribution, Bernoulli distribution and improved Bernoulli distribution that considers relation replacement. However, ignoring the dynamics in the negative sampling distribution is likely to bring about the vanishing gradient problem and impede the model training.

4.1.1. Uniform sampling

Uniform sampling [5]is the earliest, easiest and most widely-used negative sampling approach in knowledge representation learning. It refers to constructing negative triples by replacing either the head h or the tail t of a positive triple with the entity that is randomly sampled from the entity set E according to uniform distribution. However, in most cases, the uniformly sampled entity is unrelated with the corrupted positive triple, then the formed negative triple is too wrong to benefit the training. Taking the triple (London, locatedIn, UnitedKingdom) as an example, its tail entity UnitedKingdom needs to be replaced to produce counterpart negative triples. Under the uniform sampling schema, the generated negatives could be (London, locatedIn, apple) or (London, locatedIn, football). These low-quality triples will be easily distinguished by the KGE model merely in terms of different entity types, which can slow down the convergence of model training[44]. Similarly, IGAN emphasizes the zero loss problem in the random sampling mode, and explains the little contribution made by the low-quality negatives. Translation-based KGE models prefer adopting a marginal loss function with a fixed margin to discriminate positive triples from negative ones. Unreliable negatives tend to be out of the margin, which easily gives rise to zero loss. Another severe drawback of uniform sampling lies in false negative samples. To replace the head in (DonaldTrump, Gender, Male) with JoeBiden, (JoeBiden, Gender, Male) is still a true (false negative)fact.

4.1.2. Bernoulli Sampling

To alleviate the problem of false negatives, Bernoulli negative sampling [12]suggests replacing head or tail entities with different probabilities according to the mapping property of relations. That is, to give more chance of replacing the head in one-to-many relations and the tail in many-to-one relations. In the mathematical explanation, to set the probability $tph/((tph+hpt))$ for replacing the head and the probability $hpt/((tph+hpt))$ for replacing the tail after denoting tphas the average number of tail entities per head entity and hptas the average number of head entities per tail entity. Gender is a typical many-to-one relation. Replacing the tail in (DonaldTrump, Gender, Male) with high probability that is computed by Bernoulli distribution is unlikely to bring about false negative triples. Furthermore, it may generate high-quality negatives if setting extra constraints on the entity type.

Improvement in the Bernoulli sampling. Zhang et al. [45]extends Bernoulli sampling by considering relation replacement following the probability $\alpha=r/((r+e))$, here r is the number of relations and e is the number of entities. The rest $1-\alpha$ is divided by head entity replacement and tail entity replacement according to Bernoulli distribution. Such changes enhance the ability of KGE models in relation link prediction.

4.1.3. Probabilistic Sampling

Kanojia et al. [46] proposes probabilistic negative sampling to address the issue of skewed data that commonly exists in knowledge bases. For relations with less data, Uniform or Bernoulli random sampling fails to predict the missing part of golden triples among semantically possible options even after hundreds of epochs of training. Probabilistic negative sampling speeds up the process of generating corrupted triples by bringing in a tuning parameter β known as train bias that determines the probability by which the generated negative examples are complemented with early-listed possible instances. Kanojia et al. evaluates probabilistic negative sampling (PNS) over TransR in link prediction, and elicits that TransR-PNS achieves 190 and 47 position gains in Mean Rank on benchmark datasets WN18 and FB15K [5] respectively compared to TransR using Bernoulli sampling.

4.2. Dynamic Distribution-Based Sampling

Static distribution-based sampling fails in modeling the changes in negative sampling distribution and generating the negative triples with high plausibility dynamically. GAN is short for Generative Adversarial Network [47], it is capable of modeling dynamic distribution. In the GAN-based negative sampling framework, the generator dynamically approximates the constantly updated negative sampling distribution in order to provide high-quality triples while the target KGE model acts as the discriminator to distinguish between positives and negatives. Adversarial training is going on between the generator and the discriminator to optimize the final knowledge representation. Reinforcement learning is required for training GAN [19]. GAN-based framework can be performed on various KGE models as it is independent of the specific form of the discriminator [17]. However, potential risks (training instability and model collapse) embodied in reinforcement learning should not be neglected. Besides, a general estimation about negative sampling distribution, Markov chain Monte Carlo negative sampling [22], that is derived from positive sampling distribution needs to be highly regarded.

4.2.1. KBGAN

KBGAN [16] is the first work to adapt GAN to negative sampling in knowledge representation learning. It considers selecting one of two translational distance-based KGE models (DistMult [31], ComplEx [23]) as the negative sample generator and one of two semantic matching-based KGE models (TransE [5], TransD [14]) as the discriminator for adversarial training. The generator produces the probability distribution over a candidate set of uniformly sampled negativesand selects the one with highest probability to feed into the discriminator. The discriminator minimizes the marginal loss between positive and negative samples to learn the final embedding vectors. KBGAN combines four Generator-Discriminator pairs that show better performance than baselines, which reflects the strengths of the adversarial learning framework.

4.2.2. IGAN

Unlike KBGAN [16] that considers probability-based, log-loss KGE models as the generator, IGAN [17] applies a two-layer fully-connected neural network as its generator to provide negative samples with high quality. The discriminator is still the target KGE model. The embedding vectors of the corrupted positive triple are fed into the neural network and followed by non-linear activation function ReLU. The softmax function is added after to calculate the probability distribution over the whole entity set E instead of a small candidate set in KBGAN. The plausibility of the formed negative is measured by the scoring function of the target KGE model. IGAN can dynamically select negative samples with relatively high quality during adversarial training but suffers from high computational complexity.

Comparison between GAN-based and self-adversarial sampling. Adversarial Contrastive Estimation (ACE) [48] introduces a general adversarial negative sampling framework for NCE that is commonly used in natural language processing. RotatE [49] thinks that such adversarial framework is difficult to optimize since it needs to train the discrete negative sample generator and the embedding model simultaneously, which costs a lot in computation. Therefore, RotatE proposes a self-adversarial sampling approach based on the self-scoring function and introduces αas the temperature of sampling, which avoids the use of reinforcement learning. GAN-based sampling has no advantage in efficiency. In order to reduce the risk of training instability caused by reinforcement learning, both KBGAN and IGAN requires to be pre-trained, which gives rise to extra costs. By comparison, self-adversarial sampling is easier to operate, and experimental results show that it outperforms KBGAN in link prediction.

4.2.3. MCNS

Yang et al. [22] creatively derives that a nice negative sampling distribution that should be positively but sub-linearly correlated to the positive sampling distribution, and raises Markov chain Monte Carlo negative sampling (MCNS). In the proposed Sampled NCE framework, the depth first search (DFS) algorithm is applied to traverse the graph to obtain the Markov chain of the last node, from which negative samples are generated. MCNS uses the self-contrast approximation to estimate positive sampling distribution, and the Metropolis-Hastings algorithm [50]to speed up negative sampling. Embedding vectors are updated by minimizing the hinge loss after inputting the positive sample and the generated negative sample into the encoder of the framework. The importance of negative sampling is proved in the formula derivation. Experiments exhibit that MCNS performs better than all baselines in the downstream tasks and wins in terms of efficiency. The proposal of MCNS is based on the graph structureddata without limitation to knowledge representation learning, which is a generic solution to modelling dynamic negative sampling.

4.3. Custom Cluster-Based Sampling

Sampling from custom clusters means that the desired negative sample is selected from a handful of candidates rather than sampled from the whole entity set, which requires to collect entities that meet some custom standards into clusters firstly. For examples, domain sampling [51] suggests to sample from entities of the same domain, and affinity dependent sampling emphasizes the closeness between entities. Two more cluster-based sampling approaches, Trans E-SNS [18] and NSCaching[19], are elaborated in this section. Narrowing the sampling scope makes the target of negative sampling more clear, which gains efficiency. Because KGs grow rapidly and update frequently, renewing the custom clusters continually is essential and difficult.

4.3.1. TransE-SNS

Qin et al. [18]puts forward the entity similarity-based negative sampling (SNS) to mine valid negatives. Inspired by the observation that smaller distance between two entity vectors in the embedding space imply their higher similarity, the K-Means clustering algorithm [52] is used to divide all entities into a number of groups. An entity is uniformly sampled from the same cluster of the replaced head entity to complete the corrupted positive triple and when necessary, the tail entity is replaced in the same manner. The negatives generated in such a way should be highly similar to the given positive triple. Adapting SNS to TransE (TransE-SNS) and then being evaluated in link prediction and triple classification, the experiment demonstrates that SNS enhances the ability of TransE.

4.3.2. NS Caching

High-quality negative samples tend to get high plausibility measured by the scoring functions. Motivated by the skewed score distribution of negative samples, Zhang et al. [19]attempts to only track helpful and rare negatives of high plausibility using a cache. NSCaching can be considered in the same group with GAN-based approaches since they all parametrize the dynamic distribution of negative samples. To be precise, NSCaching is a distilled version of GAN-based strategy, because it has fewer parameters, it does not need to be trained through reinforcement learning, and it also avoids the model collapse that might be brought by GAN. After storing the high-quality negative triples in the cache, NSCaching samples from the cache according to uniform distribution and applies importance sampling to update it. With more concentrated sampling and more concise training, NSCaching performs better than GAN-based approaches in terms of efficiency and effectiveness.

4.4. Other Novel Approaches

There is another negative sampling strategy which cannot be definitely sorted into the three categories specified above but it ought to be mentioned since it accomplishes negative sampling and noise detection simultaneously. Because human knowledge is innumerable and changeable, bypassing crowdsourcing and manual efforts in building KGs is the mainstream. Noises and conflicts are inevitably involved due to the auto-construction, explosive growth and frequent updates of KGs. Xie et al. [20] initially proposes a novel confidence-aware knowledge representation learning framework (CKRL), and Shan et al. [21] extends this idea to negative sampling in noisy knowledge representation learning (NKRL). CKRL detects noises but applies uniform negative sampling that easily causes zero loss problems and false detection issues. NKRL proposes a confidence-aware negative sampling approach to address these problems, and the concept of negative triple confidence it introduces is conducive to generate plausible negatives by measuring their quality. NKRL also modifies the triple quality function defined in CKRL with the aim of reducing the false detection problems and improving ability of detection noises. Both CKRL and NKRL are performed on translation-based KGE models, and NKRL outperforms CKRL in link prediction.

By looking into the defined negative sampling distribution in NKRL, we find that it is similar to the self-adversarial sampling in RotatE [49] since they both sample according to the self-scoring function of the current embedding model.

5. CONCLUSIVE REMARKS

In this paper we have reviewed negative sampling in knowledge graph embedding. It is stated that KGE is a useful way to take advantage of machine learning and neural networks for knowledge graph computation and operations such as subgraph classification, node classification and link prediction. Learnt from language models in natural language processing, we argue that negative sampling is important for KGE as for language modeling. By studying negative sampling, we have sketched out existing well-known negative sampling approaches that are applied in KGE models and categorize them. We aimed to provide a basis for selecting the proper negative sampling approach to train a KGE model to its best.

We find that the existing KGE studies focus on finding new scoring functions to model multi-relational data in KGs, and with simply selecting the random mode for negative sampling. The importance and significance of negative sampling is ignored or failed to be recognized in comparison with the positive sampling.

We hope that this review can be of some help to those who are interested in negative sampling. Subsequent work maylie in comparing the approaches mentioned here through performing KG downstream applications like link prediction on benchmark datasets to find the shortages of current negative sampling and possibly to propose a new strategy for negative sampling after fully understanding the existing ones.

ACKNOWLEDGEMENTS

This work is supported by the AI University Research Centre (AI-URC) of Xi'an Jiaotong-Liverpool University through XJTLU Key Programme Special Fund KSF-P-02 and KSF-A-17. We appreciate their support.

REFERENCES

[1] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an Architecture for Never-Ending Language Learning," (in English), Proceedings of the Twenty-Fourth Aaai Conference on Artificial Intelligence (Aaai-10), pp. 1306-1313, 2010.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in SIGMOD Conference, 2008.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in WWW '07, 2007.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," presented at the Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Lake Tahoe, Nevada, 2013.

[5] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," presented at the Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Lake Tahoe, Nevada, 2013.

[6] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," presented at the Proceedings of the 28th International Conference on International Conference on Machine Learning, Bellevue, Washington, USA, 2011.

[7] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 4710-4723: Association for Computational Linguistics.

[8] R. Wang, B. Li, S. Hu, W. Du, and M. Zhang, "Knowledge Graph Embedding via Graph Attenuated Attention Networks," IEEE Access, vol. 8, pp. 5212-5224, 2020.

[9] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," J. Mach. Learn. Res., vol. 13, no. 1, pp. 307–361, 2012.

[10] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in AISTATS, 2010.

[11] B. Kotnis and V. Nastase, "Analysis of the Impact of Negative Sampling on Link Prediction in Knowledge Graphs," 08/22 2017.

[12] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," presented at the Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada, 2014.

[13] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Entity and Relation Embeddings for Knowledge Graph Completion," in AAAI, 2015.

[14] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "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), Beijing, China, 2015, pp. 687-696: Association for Computational Linguistics.

[15] H. Xiao, M. Huang, and X. Zhu, "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), Berlin, Germany, 2016, pp. 2316-2325: Association for Computational Linguistics.

[16] L. Cai and W. Y. Wang, "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, Volume 1 (Long Papers), New Orleans, Louisiana, 2018, pp. 1470-1480: Association for Computational Linguistics.

[17] P. Wang, S. Li, and R. Pan, "Incorporating GAN for Negative Sampling in Knowledge Representation Learning," in AAAI, 2018.

[18] S. Qin, G. Rao, C. Bin, L. Chang, T. Gu, and W. Xuan, "Knowledge Graph Embedding Based on Adaptive Negative Sampling," Singapore, 2019, pp. 551-563: Springer Singapore.

[19] Y. Zhang, Q. Yao, Y. Shao, and L. Chen, "NSCaching: Simple and Efficient Negative Sampling for Knowledge Graph Embedding," in 2019 IEEE 35th International Conference on Data Engineering (ICDE), 2019, pp. 614-625.

[20] R. Xie, Z. Liu, and M. Sun, "Does William Shakespeare REALLY Write Hamlet? Knowledge Representation Learning with Confidence," ArXiv, vol. abs/1705.03202, 2018.

[21] Y. Shan, C. Bu, X. Liu, S. Ji, and L. Li, "Confidence-Aware Negative Sampling Method for Noisy Knowledge Graph Embedding," 2018 IEEE International Conference on Big Knowledge (ICBK), pp. 33-40, 2018.

[22] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang, "Understanding Negative Sampling in Graph Representation Learning," Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020.

[23] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," presented at the Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, New York, NY, USA, 2016.

[24] S. Kazemi and D. Poole, "SimplE Embedding for Link Prediction in Knowledge Graphs," in NeurIPS, 2018.

[25] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge Graph Embedding Based Question Answering," Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019.

[26] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[27] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications," IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 12, pp. 2724-2743, 2017.

[28] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition and Applications," ArXiv, vol. abs/2002.00388, 2020.

[29] L. Drumond, S. Rendle, and L. Schmidt-Thieme, "Predicting RDF triples in incomplete knowledge bases with tensor factorization," 03/26 2012.

[30] R. Reiter, "Deductive Question-Answering on Relational Data Bases," in Logic and Data Bases, H. Gallaire and J. Minker, Eds. Boston, MA: Springer US, 1978, pp. 149-177.

[31] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," CoRR, vol. abs/1412.6575, 2015.

[32] X. Dong et al., "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 08/24 2014.

[33] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D Knowledge Graph Embeddings," ArXiv, vol. abs/1707.01476, 2018.

[34] L. Guo, Z. Sun, and W. Hu, "Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs," in ICML, 2019.

[35] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for Knowledge Graph Completion," ArXiv, vol. abs/1909.03193, 2019.

[36] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," Cham, 2018, pp. 593-607: Springer International Publishing.

[37] S. Guo, Q. Wang, B. Wang, L. Wang, and L. Guo, "Semantically Smooth Knowledge Graph Embedding," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 2015, pp. 84-94: Association for Computational Linguistics.

[38] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling Relation Paths for Representation Learning of Knowledge Bases," ArXiv, vol. abs/1506.00379, 2015.

[39] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph and Text Jointly Embedding," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1591-1601: Association for Computational Linguistics.

[40] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Knowledge Graph Embedding with Iterative Guidance from Soft Rules," ArXiv, vol. abs/1711.11231, 2018.

[41] C. Dyer, "Notes on Noise Contrastive Estimation and Negative Sampling," ArXiv, vol. abs/1410.8251, 2014.

[42] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun, "Knowledge Representation Learning: A Quantitative Review," ArXiv, vol. abs/1812.10901, 2018.

[43] Y. Bengio and J. Senecal, "Adaptive Importance Sampling to Accelerate Training of a Neural Probabilistic Language Model," IEEE Transactions on Neural Networks, vol. 19, no. 4, pp. 713-722, 2008.

[44] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823.

[45] Y. Zhang, W. Cao, and J. Liu, "A Novel Negative Sample Generating Method for Knowledge Graph Embedding," presented at the Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks, Beijing, China, 2019.

[46] V. Kanojia, H. Maeda, R. Togashi, and S. Fujita, "Enhancing Knowledge Graph Embedding with Probabilistic Negative Sampling," Proceedings of the 26th International Conference on World Wide Web Companion, 2017.

[47] I. J. Goodfellow et al., "Generative adversarial nets," presented at the Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, 2014.

[48] A. Bose, H. Ling, and Y. Cao, "Adversarial Contrastive Estimation," ArXiv, vol. abs/1805.03642, 2018.

[49] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," ArXiv, vol. abs/1902.10197, 2019.

[50] N. Metropolis, A. W. Rosenbluth, M. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," Journal of Chemical Physics, vol. 21, pp. 1087-1092, 1953.

[51] Q. Xie, X. Ma, Z. Dai, and E. Hovy, "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), Vancouver, Canada, 2017, pp. 950-962: Association for Computational Linguistics.

[52] J. Hartigan and M. C. Wong, "Statistical algorithms: algorithm AS 136: a K-means clustering algorithm," 1979.