A Novel Negative Sample Generating Method for Knowledge Graph Embedding

Yi Zhang
College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China
Wuhan Digital Engineering Research Institute, China Shipbuilding Industry Corporation, Wuhan, 430074, China
yzhang85@hrbeu.edu.cn

Wanhua Cao
College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China
Wuhan Digital Engineering Research Institute, China Shipbuilding Industry Corporation, Wuhan, 430074, China
clarkmss@163.com

Juntao Liu
Wuhan Digital Engineering Research Institute, China Shipbuilding Industry Corporation, Wuhan, 430074, China
prolay@163.com

Abstract

In order to extract relation and reason efficiently in knowledge graphs, entities and relations are encoded into a continuous low-dimensional semantic space. In the negative samples generating stage, most knowledge graphs embedding methods pay more attention to replace head or tail entities for high-efficient training, and seldom to replace relations. These negative sample Generating methods contribute little for relation prediction. In this paper, we propose a new negative triplet generating method. Compared to those methods which replace only the entities such as TransE, TransH and TransR, the proposed method replaces both the entities and the relations in appropriate proportions. Experimental results on several classical data sets such as WN18 and FB15K show that the proposed method significantly outperforms original methods in the accuracy of the relation link prediction with little impact on the entity link prediction. According to the experimental results, we also find that the ability of the relation link prediction increases and the ability of the entity link prediction decreases when the proportion of the relations replacement probability increases.

1 Introduction

In recent years, knowledge graphs have been used as important resources in many fields, such as interactive retrieval [6], intelligence analysis [8] and intelligent question answering [11]. Some famous knowledge graphs, such as FreeBase [1], WordNet and YAGO [15], store vast amounts of structured data usually in the form of triplets (*head entity*, *relation*, *tail entity*) (abridged as (h,r,t), each of which indicates that there is a *relation* between the *head* and the *tail* in real world. For example, the triple (*Beijing*, *the capital of*, *China*) indicates that there is a relationship between the head entity "*Beijing*" and the tail entity "*China*" as "*the capital of*". Although current knowledge graphs contain millions of entities and billions of relational facts, there are still huge amounts of unobserved relational facts. Knowledge representation learning, as an efficient way to automatically predict the unknown relational facts, completes knowledge graphs. It embeds the semantic information of the entities and relations in knowledge graphs into the dense low-dimensional real-valued vectors, which can fully reveal the semantic relation between the entities and the relations. Because of the advantage, many methods of knowledge representation learning are proposed. These methods are usually divided into two categories: Tensor Factorization Based Methods and Mapping Based Methods. The former contains RESCAL [13], DistMult [20], HolE [12], etc., and the latter contains Unstructured Model (UM) [2], Structured Embedding (SE) [4], and Translation-based methods [3, 17, 10, 18].

Among these methods, the margin-based score function [4] usually is utilized as the training objective, which separate a positive triplet and its corresponding negative triplet. But there is a difficulty in training with the margin-based score function. There are only positive triplets in knowledge graphs. Usually, a negative triplet is constructed by replacing the head or tail entity of a positive triplet randomly. Because of the simplicity and easy implementation, it's widely used by some translation-based knowledge representation models, such as TransE, TransH, TransR, etc. However, this simple randomly replacing method may introduce many false negative triplets in the training process. Here, the false negative triplet is the positive triplets which are mistakenly treated in training as a negative example. Because it is not possible to traverse all the positive triplets when constructing negative examples, false negative triplets cannot be avoided, and they can only be reduced by reasonable sampling. The TransH improves the negative triplets constructing method by setting different probabilities for replacing the head or tail entity. In TransH, according to the mapping property of the relations, such as reflexive, one-to-

International Conference on Embedded Wireless Systems and Networks (EWSN) 2019
25–27 February, Beijing, China

ISBN: 978-0-9949886-3-8

many, many-to-one and many-to-many, the head or tail entities are replaced in different probabilities when constructing negative triplets. Thus, the head entity in one-to-many relations will get more chance to be replaced, and the tail entity in many-to-one relations will get more chance to be replaced. This improvement, widely used in many methods [10, 20], can effectively reduce the probability of generating false negative triplets, and improve the accuracy of the entity link prediction. To achieve better results in the relation link prediction, some presentation models, such as TransG [18], IKRL [19], replace the relation of a triplet to construct the negative sample for training. However, because of the imbalance of randomly sampling and the unsuitable proportion of replacing probabilities between the relation and entity in a triplet, it may introduce many false negative triplets into training and lead some errors for link predictions.

In this paper, we propose a novel method to generate negative samples by introducing flexible relation replacements. Furthermore, by analyzing the impact of the proportion of the probability of relation and entity replacement on the performance of link prediction, we propose the proper proportion of probabilities to replace relations and entities according to the number of entities and relations in the knowledge graph. And this proportion and our negative samples generation method are verified through the experiments. In order to evaluate the performance of link prediction of knowledge representation models with our training method more comprehensively, we adopt both the entity and relation prediction protocol. Experiments show that our negative examples generation method can improve the performances of relation prediction, meanwhile, with a little influence on entity prediction. Indeed, we find that, in general, when the probability of relation replacement decreases, the accuracy of relation link prediction decreases, and the accuracy of entity link prediction increases. When the probability of relation replacement increases, the accuracy of relation link prediction increases, and the accuracy of entity link prediction decreases.

The remainder of this paper is organized as follows. In Section 2, the related work are summarized. Then, there is an elaboration on our negative samples generating method in Section 3. And the proposed negative sample generating method is compared with the original methods in three models to verify the effectiveness of the improvement in Section 4. Finally, the conclusion and the future research directions are given.

2 Related Work

The negative sample generation is crucial for the training of knowledge representation models. The higher quality of negative samples, the more it can contribute towards the training efficiency. Given a positive triplet $(h,r,t) \in S$, we can simply generate some negative triplets by replacing either the head entity h or the tail entity t with a random entity sampled uniformly from the entity set E. So we can get a negative triplet (h',r,t) or (h,r,t'), and the set of these negative triplets is marked as S'. Note that we should ensure that this negative triplet does not appear in set S. Then we can describe the negative triplet set as follows, i.e.,

$$\begin{aligned} S' = \{\,(h',r,t)|\, h,h',t \in E \wedge r \in R \\ \wedge (h,r,t) \in S \wedge (h',r,t) \notin S\} \\ \cup \{\,(h,r,t')|\, h,t,t' \in E \wedge r \in R \\ \wedge (h,r,t) \in S \wedge (h,r,t') \notin S\}\,. \end{aligned} \tag{1}$$

In the training procedure, this method denoted as "unif" just simply generates negative triplets by uniformly replacing head or tail entities. However, this method is conventional, and it might introduce false-negative samples which are unobserved real fact. For example, given a triplet of (*KobeBryant*,*Career*,*Player*), this method may generate a false-negative example (*YaoMing*,*Career*,*Player*). This triplet may not exist in positive sample sets, but it is real.

By setting different probabilities for replacing the head and the tail, TransH improves the approach of negative example generation to reduce such false-negative examples and denotes it as "bern". Specifically, it give more chance to replace the "one" side, i.e. more opportunities are allocated for head entity replacements if the relation is 1-to-N, and more opportunities are allocated for tail entity replacements if the relation is N-to-1. For example, the relation Gender is an N-to-1 relation. Given a triplet (*YaoMing*, *Gender*,*Male*), we can ensure the tail-replaced negative triplets (*YaoMing*,*Gender*,*Female*) and (*YaoMing*,*Gender*,*China*) both are true negative samples, but the head-replaced one (*KobeBryant*,*Gender*,*Male*) can almost be considered a false-negative sample. Obviously, the triplets which contains the N-to-1 relation, such as *Gender*, are more likely to generate real negative triplets by replacing the tail entity. In this way, it can reduce the chance of generating false-negative samples.

In TransG [18], negative triplets are generated not only by replacing the head or tail entities but also by replacing the relation r with a random relation uniformly sampled from the relation set R, i.e.,

$$\begin{aligned} S' = \{\,(h',r,t)|\, h,h',t \in E \wedge r \in R \\ \wedge (h,r,t) \in S \wedge (h',r,t) \notin S\} \\ \cup \{\,(h,r,t')|\, h,t,t' \in E \wedge r \in R \\ \wedge (h,r,t) \in S \wedge (h,r,t') \notin S\} \\ \cup \{\,(h,r',t)|\, h,t \in E \wedge r,r' \in R \\ \wedge (h,r,t) \in S \wedge (h,r',t) \notin S\}\,. \end{aligned} \tag{2}$$

They just simply generate negative triplets by replacing entities and relations with equal probability, and ignore the impact of the proportion of entity and relations replacement probabilities on the performance of entity and relation prediction.

In order to improve the quality negative training samples, the Type-Constraints [9] adopt a strategy by constraining the entities' range to generate negative samples. Inspired by generative adversarial networks (GANs) [7], KBGAN [5] adopts some knowledge embedding models as the generator to construct better quality negative samples. To achieve the excellent efficiency of training, ComplEx [16] investigates the influence of the proportion of positive and negative samples. The investigation shows that more negatives samples generated per positive one can lead to better results, and 50 negative samples per positive is a good trade-off between accuracy and training time.

In this work, we introduce the relation replacement into the negative sample generation, moreover, set different replacement probabilities for entities and relations to achieve better results on both entity and relation predictions.

3 The Proposed Negative Samples Generating Method

In this section, a novel method of generating negative samples is proposed to improve the training procedures of knowledge representation models. In the proposed method, the negative samples set S' is generated by replacing the head entity h, the tail t or the relation r in the positive triplet $(h,r,t) \in S$, and only one item in each observed triplet is replaced at a time. How to set the probability of relation and entity replacement is the focus of this paper. We solve this problem in two steps. First, the entity replacement probabilities and relation replacement ones are set according to the numbers of entities and relations in knowledge graphs, next the head and tail entity replacement probabilities are set according to two statistics among all triplets of the relation r: 1) the average number of tail entities per head entity, and 2) the average number of head entities per tail entity. The detailed description is as follows.

In many knowledge representation models, the margin-based score function is usually utilized with negative sampling as objective for training:

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S'} \left[\gamma + f_r(h,t) - fr\left(h',t'\right)\right]_+ \quad (3)$$

Here, (h,r,t) is a positive example in training sets which is extracted from knowledge graphs, (h',r',t') is the negative one, γ is the margin separating positive and negative triplets, $[x]_+$ denotes the positive part of x, and S' denotes the negative set. Different from the assumption of other loss functions which considers the negative examples should be false, this loss function just assumes that negative examples are less valid than positive ones. And it makes negative examples get lower scores than positive ones.

The optimization of formula (3) can be carried out by stochastic gradient descent (SGD) [14] in minibatch mode. The detailed training procedure is described as follows. First, all embedding vectors of entities and relations are initialized randomly from uniform distributions. Next, at each iteration, the embedding vectors of entities and relations are normalized. Then, a small set of positive facts is sampled from the set S, and for each of them, a negative triplet is generated accordingly. The positive facts and the generated negative ones are treated as training examples of the minibatch. And the embedding vectors of entities and relations in the minibatch are then updated by a gradient descent with constant learning rate.

In order to improve the training procedure of knowledge graphs embedding methods, such as the TransE, TransH and TransR, we extend the negative example space by corrupting triplets with relation replacements. So our negative triplet sets can also be described as formula (2) in Section 2. Then, a novel method is proposed to generate negative triplets, and applied to three representation models, including TransE, TransH and TransR. Inspired by the method of TransH, we set different probabilities for replacing the relation, head entity or tail entity in the positive triplet.

First, we define the probability of relation replacement as a real value variable α, and $\alpha \in [0,1]$. So the probability of entity replacement could be easily deduced as $1-\alpha$, i.e.,

$$\mathbf{P}_{head} + \mathbf{P}_{tail} = 1 - \alpha \quad (4)$$

where $\mathbf{P}_{head}$ denotes the probability of head entity replacement, and $\mathbf{P}_{tail}$ denotes the probability of tail entity replacement.

Then, inspired by TransH, we assign different probabilities for head/tail entity replacement. The probability of head entity replacement is obtained as below:

$$\mathbf{P}_{head} = (1-\alpha)\frac{tph}{tph+hpt} \quad (5)$$

,and the probability of tail entity replacement is obtained as below:

$$\mathbf{P}_{tail} = (1-\alpha)\frac{hpt}{tph+hpt} \quad (6)$$

Next, we need to explore the value of α, i.e., the probability of relation replacement $\alpha \in [0,1]$. If $\alpha = 0$, the relations are not replaced when the negative triplets are generated, and our negative samples generation method is equivalent to original method. If $\alpha = 1$, our method generates negative triplets only by replacing relations in the positive samples. The number of relations, in general, is much smaller than entities in knowledge graphs. Thus, in this case, the negative sample space is reduced so much that the models cannot get sufficiently trained.

Taking into account the feature of knowledge graphs in terms of the number of entities and relations, we also propose $\alpha = \frac{N_{relation}}{N_{entity}+N_{relation}}$, here $N_{relation}$ is the number of relations, and N_{entity} is the number of entities. For short, $\frac{N_{relation}}{N_{entity}+N_{relation}}$ is denoted as $\frac{r}{e+r}$. This value measures the diversity of relations of the data sets. In section 4, the experimental results indicate the impact of the probability of relation replacement α on the performance of entity and relation prediction and verify that the $\frac{r}{e+r}$ is an appropriate probability of relation replacements in next.

4 Experiments

In this section, the proposed negative samples generating method is verified on FB15K and WN18 datasets for the tasks of entity prediction and relation prediction. The impact of the probability of relation replacement is also investigated. First, the datasets FB15K and WN18 are introduced. Then, in the experimental setup, our evaluation protocol and implementation are described. In particular, in order to analyze the impact of replacing probability on the entity and relation prediction, we select the relation replacement probabilities from a list of real values between 0 and 1.0. The experimental results show the effectiveness of our improvement in negative samples generating method.

4.1 Data sets

In this paper, we use two public benchmark datasets, F-B15K and WN18, to evaluate our method and study the impact of the relation-entity replacing probability assignment on the performance of entity and relation prediction of some representation models, such as TransE, TransH and TransR. Both WN18 and FB15k are released in [3]. Following [3], we split those triples into train, validation and test set. Table 1 summarizes the statistics of the two datasets.

Table 1. Statistics of the datasets

Dataset	#Rel	#Ent	#Train	#Valid	#Test
WN18	18	40,943	141,442	5,000	5,000
FB15K	1,345	14,951	483,142	50,000	59,071

4.2 Experimental setup

Evaluation protocol For evaluation, we follow the identical protocol with [3]. For each test triplet (h,r,t), we corrupt this triplet and replace its head entity h by each entity e in the entity set E in turn. We first calculate the scores of function f_r of these corrupted triplets, and then sort their scores by ascending order; the rank of the correct triplet is finally stored. Similarly, the tail entity or the relation in each test triplet (h,r,t) is replaced, and the rank of the correct triplet is stored. We calculate the average rank of the testing triplets which is called *mean rank*, denoted as *MR*. Simultaneously, we calculate the proportion of correct triplet ranked in top 10 which is denoted as *Hits@10* during the testing procedure described above. In this paper, we will report evaluation results just in the setting named "Filter" which used in [3].

To evaluate the ability of the model in knowledge graphs completion more comprehensively, besides the entity predictions above, we evaluate the relation prediction additionally. Similarly, we also consider these two measures: (1) mean rank of correct relations, denoted as *MR*; (2) the proportion of correct answers ranked in top 2 for relations, denoted as *Hit@2*.

Implementation We investigate the performance of the representation models, including TransE, TransH and TransR, with the training improvement proposed in this paper. In the experiments, myTransE denotes the TransE model with the training improvement proposed in this paper. And the meanings of myTransH and myTransR are similar to that of myTransE. For a more complete performance comparison, we implement TransE [3], TransH [17] and TransR [10] and our results in additional experiments following the same experimental settings in these papers.

We select the dimension of vectors k among $\{20,50,100\}$, the margin γ among $\{1,2,4,8\}$, the learning rate λ for SGD among $\{0.1,0.01,0.001\}$, the mini-batch size B among $\{100,400,800,1440,1600\}$, the probability of relation replacement α among $\{0,\frac{r}{e+r},0.2,0.5,0.8,1\}$. We take the L_1 distance to measure dissimilarity of the score functions. Then, for the two data sets, training is limited to 1,000 epochs. The best configuration is obtained by early stopping (100 epochs) using the lowest mean rank of entity prediction on the validation sets, under raw setting, in the case $\alpha = 0$. The experiment of relation prediction is only verified on FB15K, because the number of relations in WN18 is so small, only 18, that it's difficult to demonstrate the performances of relations predicting without enough samples.

4.3 Results

We compare our negative samples generation method with the early methods "unif"and "bern" in different representation models, including TransE, TransH and TransR, and report the results obtained by these models. The task use a part of the results published in the original papers for comparisons, and the "*" denotes the models which we implement for comparison.

TransE and its training improvement The experimental results of TransE and its training improvement, denoted as myTransE, on both WN18 and FB15k are shown in Table 2. The best hyperparameters on FB15K are $\lambda = 0.001$, $k = 50$, $\gamma = 1$, $B = 1600$, and the best hyperparameters on WN18 are $\lambda = 0.001$, $k = 50$, $\gamma = 4$, $B = 400$.

TransH and its training improvement The TransH model with advanced training is denoted as myTransH and it has the improvements of negative samples generation in accordance with myTransE. The best hyperparameters are $\lambda = 0.001$, $k = 50$, $\gamma = 2$, $B = 400$ on FB15K and $\lambda = 0.001$, $k = 50$, $\gamma = 4$, $B = 100$ on WN18. The results of entity and relation prediction are shown in Table 3.

TransR and its training improvement We improve the training procedure of TransR model with our negative samples generation as myTransE and myTransH, and denote it as myTransR. Then, the best configurations of myTransR are obtained in the same way as myTransE and myTransH, in the case of $\alpha = 0$. So the best hyperparameters are $\lambda = 0.001$, $k = 50$, $\gamma = 1$, $B = 4800$ on FB15K and $\lambda = 0.001$, $k = 50$, $\gamma = 4$, $B = 1440$ on WN18. The results of entity and relation prediction are shown in Table 4.

From the experimental results in Table 2 - Table 4, we investigate the effect of different α on prediction performances of myTransE, myTransH and myTransR. We can observe that: (1) compared the entity prediction results on these two datasets, the impact of α on the performance of entity prediction is much larger on WN18 than that on FB15K. The reason may be the number of FB15K relations is much larger than WN18, and the number of FB15K entities is less than WN18; (2) when α approaches to 1, the probability of entity replacement decreases, and the performance of entity prediction becomes worse on both WN18 and FB15K; (3) when α approaches to 0, the probability of entity replacement increases, and the performance of entity prediction of becomes better on both WN18 and FB15K; especially, in the case of $\alpha = \frac{r}{e+r}$, the performance of entity prediction is close to that of TransE, TransH and TransR under "bern" setting on both WN18 and FB15K, and sometimes, maybe better than them. It can be concluded that the relation prediction performance is improved by the proposed negative sample generating method. Even a very small value of α can greatly improve the ability of relation prediction of myTransE, myTransH and myTransR. From a comprehensive experimental analysis, $\alpha = \frac{r}{e+r}$ is a good compromise of entity prediction and relation prediction.

Table 2. Evaluation results on entity prediction of TransE and its training improvement

Datasets	**WN18**		**FB15K**			
Metric	Ent Link		Ent Link		Rel Link	
	MR	Hit@10	MR	Hit@10	MR	Hit@2
TransE Unif [3]	251	89.2	**125**	47.1	38.93*	68.8*
TransE Bern*	**249**	92.2	127	**65.3**	31.42	75.9
myTransE($\alpha = 0$)	258	92.2	126	**65.3**	32.02	76.1
myTransE($\alpha = \frac{r}{e+r}$)	270	**92.2**	**128**	**64.4**	**1.87**	**87.4**
myTransE($\alpha = 0.2$)	322	88	133	63.2	1.63	90.2
myTransE($\alpha = 0.5$)	471	84.4	141	59.9	1.41	90.8
myTransE($\alpha = 0.8$)	780	73.9	173	52.4	1.25	91.7
myTransE($\alpha = 1$)	4312	29.1	876	21.5	1.39	93.3

Table 3. Evaluation results on entity prediction of TransH and its training improvement

Datasets	**WN18**		**FB15K**			
Metric	Ent Link		Ent Link		Rel Link	
	MR	Hit@10	MR	Hit@10	MR	Hit@2
TransH Unif [17]	303	86.7	84	58.5	13.93*	79.1*
TransH Bern [17]	388	82.3	87	64.4	16.12*	81.9*
myTransH($\alpha = 0$)	**251**	**92.8**	120	**62.8**	14.2	82.6
myTransH($\alpha = \frac{r}{e+r}$)	**269**	**92.6**	**117**	**62.2**	**1.61**	**90.9**
myTransH($\alpha = 0.2$)	314	88.1	119	61.2	1.42	92.1
myTransH($\alpha = 0.5$)	413	83.2	133	57.9	1.22	93.8
myTransH($\alpha = 0.8$)	808	68.8	167	51.1	1.01	94.4
myTransH($\alpha = 1$)	5239	15.2	922	21.2	1.14	94.1

Table 4. Evaluation results on entity prediction of TransR and its training improvement

Datasets	**WN18**		**FB15K**			
Metric	Ent Link		Ent Link		Rel Link	
	MR	Hit@10	MR	Hit@10	MR	Hit@2
TransR Unif [10]	**219**	91.7	78	65.5	700.32*	0.74*
TransR Bern [10]	225	92	**77**	**68.7**	599.6*	1.11*
myTransR($\alpha = 0$)	323	**92.9**	120	61.4	601.4	1.04
myTransR($\alpha = \frac{r}{e+r}$)	**313**	**92.8**	**112**	**60.9**	**5.52**	**78.54**
myTransR($\alpha = 0.2$)	396	90.3	113	60.2	4.22	84.52
myTransR($\alpha = 0.5$)	650	89.5	123	58	3.05	88.66
myTransR($\alpha = 0.8$)	568	86.8	140	53.4	2.15	91.13
myTransR($\alpha = 1$)	2333	11.1	531	29.5	1.34	92.41

5 Conclusions

In this paper, we propose a novel method to generate negative samples for knowledge graph representation by introducing flexible relation replacements. And our method improves the accuracy of entity and relation link prediction, e-specially the contribution for relation prediction. Moreover, we study the effect of the proportion of the probabilities of relation and entity replacements in the negative sample generation on the entity and the relation prediction. The experimental results show that setting a suitable probability of the relation replacement for the negative sample generation during training can improve the ability of relation predicting of models significantly. Given a certain data set, it's possible to balance the relation and entity prediction abilities by setting a suitable probability of the relation replacement according to the numbers of entities and relations in this data set. There are several future research directions. First, our method only considers the proportion of the numbers of entities and relations for the negative sample generations. Other information, such as the types of entities and relations, the instance of entities etc., can also be considered into negative samples generations. In future research, we plan to use these information to enrich the representation models. Next, to construct negative triples, one conventional method is to replace the head entity or tail entity or relation of a triplet randomly. But this method could just discriminate easy positive-negative triplet pairs, such as $(MichealJordan, gender, male)$ is positive and $(MichealJordan, gender, female)$ is negative. It's necessary to research how to enhance the discrimination by improving the negative samples generation methods; Finally, using more negative samples in training is also worthy of further exploration.

6 Acknowledgments

This work is supported by National Natural Science Foundation of China (No.61403350 and No.61401228).

7 References

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase:a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250, 2008.

[2] A. Bordes, X. Glorot, and J. Weston. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *International Conference on Neural Information Processing Systems*, pages 2787–2795, 2013.

[4] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *AAAI Conference on Artifi-*

cial Intelligence, AAAI 2011, San Francisco, California, Usa, August, 2012.

[5] L. Cai and W. Y. Wang. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

[6] B. Gonçalves and H. V. Jagadish. Bsmooth: Learning from user feedback to disambiguate query terms in interactive data retrieval. *International Journal of Approximate Reasoning*, 2016.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3:2672–2680, 2014.

[8] Q. I. Guilin, H. Gao, and W. U. Tianxing. The research advances of knowledge graph. *Technology Intelligence Engineering*, 3(1):4–25, 2017.

[9] D. Krompaß, S. Baier, and V. Tresp. Type-constrained representation learning in knowledge graphs. In *International Conference on the Semantic Web - ISWC*, pages 640–655, 2015.

[10] Y. Lin, Z. Liu, X. Zhu, X. Zhu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187, 2015.

[11] K. Liu, J. Zhao, S. He, and Y. Zhang. Question answering over knowledge bases. *IEEE Intelligent Systems*, 30(5):26–35, 2015.

[12] M. Nickel, L. Rosasco, and T. Poggio. Holographic embeddings of knowledge graphs. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 1955–1961, 2016.

[13] M. Nickel, V. Tresp, and H. P. Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on International Conference on Machine Learning*, pages 809–816, 2011.

[14] H. Robbins and S. Monro. *A Stochastic Approximation Method*. Springer New York, 1985.

[15] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago:a core of semantic knowledge. In *International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May*, pages 697–706, 2007.

[16] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, volume 48, pages 2071–2080, 2016.

[17] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[18] 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)*, volume 1, pages 2316–2325, 2016.

[19] R. Xie, Z. Liu, T.-S. Chua, H.-B. Luan, and M. Sun. Image-embodied knowledge representation learning. In *IJCAI*, 2017.

[20] B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*, 2015.