Improved Knowledge Base Completion by the Path-Augmented TransR Model

Wenhao Huang[1,2](✉), Ge Li[1,2], and Zhi Jin[1,2]

[1] Key Laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing, China
wenhao.huang@pku.edu.cn, {lige,zhijin}@sei.pku.edu.cn
[2] Software Institute, Peking University, Beijing, China

Abstract. Knowledge base completion aims to infer new relations from existing information. In this paper, we propose path-augmented TransR (PTransR) model to improve the accuracy of link prediction. In our approach, we build PTransR based on TransR, which is the best one-hop model at present. Then we regularize TransR with information of relation paths. In our experiment, we evaluate PTransR on the task of entity prediction. Experimental results show that PTransR outperforms previous models.

Keywords: Knowledge base completion · Relation path · Link prediction

1 Introduction

Large scale knowledge bases such as WordNet [1] and FreeBase [2] are important resources for natural language processing (NLP) applications like web searching [3], automatic question answering systems [4], and even medical informatics [5]. Formally, a *knowledge base* is a dataset containing triples of two entities and their relation. A triplet (h, r, t), for example, indicates that the *head entity* h and the *tail entity* t have a relation r. Despite massive triplets a knowledge base contains, evidence in the literature suggests that existing knowledge bases are far from complete [6,7].

In the past decades, researchers have proposed various methods to automatically construct or populate knowledge bases from plain texts [6,8], semi-structured data on the Web [9,10], etc. Recently, studies have shown that embedding the entities and relations of a knowledge base into a continuous vector space is an effective way to integrate the global information in the existing knowledge base and to predict missing triplets without using external resources (i.e., additional text or tables) [7,11–14].

Bordes et al. [11] propose the TransE approach, which translates entities' embeddings by that of a relation, to model knowledge bases. That is to say, the relation between two entities can be represented as a vector offset, similar to

G. Li et al. (Eds.): KSEM 2017, LNAI 10412, pp. 149–159, 2017.
DOI: 10.1007/978-3-319-63558-3_13

word analogy tasks for word embeddings [15] and sentence relation classification by sentence embeddings [16]. For one-to-many, many-to-one, and many-to-many relations, however, such straightforward vector offset does not make much sense. Considering a head entity `China` and the `country-city` relation, we can think of multiple plausible tail entities like `Beijing`, `Tianjin`, and `Shanghai`. These entities cannot be captured at the same time by translating the head entity and relation embeddings. Therefore, researchers propose to map entities to a new space where embedding translation is computed, resulting in TransH [7], TransR [12], and other variants. Among the above approaches, TransR achieves the highest performance on established benchmarks.

One shortcoming of the above methods is that only the direct relation (i.e., one-hop relation) between two entities is considered. In a knowledge base, some entities and relations only appear a few times; they suffer from the problem of data sparsity during training. Fortunately, the problem can be alleviated by using multi-hop information in a knowledge base. Guu et al. [13] present a random walk approach to sample entities with composited relations. Likewise, Lin et al. [14] propose a path-augmenting approach that uses multi-hop relations between two entities to regularize the direct relation between the same entity pair. Their experiments show the path-augmented TransE model (denoted as PTransE) outperforms the one-hop TransE model.

In this paper, we are curious whether we can combine the worlds, i.e., whether the path-augmenting technique is also useful for a better one-hop "base" model. Therefore, we propose to leverage TransR [12] as our cornerstone, but enhance it with path information as in [14], resulting a new variant, PTransR. We evaluate our model on the FreeBase dataset. Experimental results show that modeling relation paths is beneficial to the base model TransR, and that PTransR also outperforms PTransE in entity prediction. In this way, we achieve the state-of-the-art link prediction performance in the category that uses only the knowledge base itself (i.e., without additional textual information).

The rest of this paper is organized as follows. In Sect. 2, we describe the base model TransR and then discuss the path-augmented variant PTransR. In Sect. 3, we compare our PTransR model with other baselines in an entity prediction experiment; we also have in-depth analysis regarding different groups of relations, namely 1-to-1, 1-to-n, n-to-1, and n-to-n relations. In Sect. 4, we briefly review previous work in information extraction. Finally, we conclude our paper in Sect. 5.

2 Our Approach

In this section, we present our PTransR model in detail. In Subsect. 2.1, we introduce the TransE model and explain how TransR overcomes the weakness of TransE. Then, we augment TransR model with path information in Subsect. 2.2.

2.1 Base Model: TransR

As said in Sect. 1, embedding entities and their relation into vector spaces can effectively exploit internal structures that a knowledge base contains, and thus is helpful in predicting missing triplets without using additional texts.

The first model in such research direction is TransE [11]. It embeds entities and their relation in a same low-dimensional vector space; the two entities' embeddings are translated by a relation embedding, which can be viewed as an offset vector. In other words, for a triplet (h, r, t), we would like $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. (Here, bold letters refer to the embeddings of head/tail entities and the relation.) The plausibility of a triplet (h, r, t) is then evaluated by a scoring function

$$f_r(\mathbf{h}, \mathbf{t}) \triangleq \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|, \tag{1}$$

where $\|\cdot\|$ denotes either ℓ_1-norm or ℓ_2-norm. $f_r(\mathbf{h}, \mathbf{t})$ is expected to be small if (h, r, t) is a positive triplet.

To further analyze the performance of TransE, Bordes et al. [11] divide relations into four groups, namely 1-to-1, 1-to-n, n-to-1, and n-to-n, according to the mapping properties of a relation. For example, `country-city` is a 1-to-n relation, because a country may have multiple cities, but a city belongs to only one country. The weakness of TransE is that entity embeddings on the `many` side tend to be close to each other, which is the result of expecting $f_r(\mathbf{h}, \mathbf{t})$ to be small for all positive triplets. Therefore, it is hard for TransE to distinguish among the entities which are on the `many` side.

To solve the above problem, TransR [12] embeds entities and relations into two separate spaces: the *entity space* and the *relation space*. It uses relation-specific matrices $\mathbf{M}_r$ to map an entity from its own space to the relation space, given by $\mathbf{h}_r = \mathbf{M}_r\mathbf{h}$ and $\mathbf{t}_r = \mathbf{M}_r\mathbf{t}$, so that translation can be accomplished by regarding relation embedding as an offset vector, i.e., $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$. To achieve this goal, TransR defines the scoring function as

$$f_r(\mathbf{h}, \mathbf{t}) \triangleq \|\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\|_2^2. \tag{2}$$

To train the model, we shall generate negative samples and use the hinge loss. The overall cost function of the TransR model is

$$\mathcal{L}_{\text{TransR}} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} \max\left\{0, \gamma + f_r(\mathbf{h}, \mathbf{t}) - f_r(\mathbf{h}', \mathbf{t}')\right\}, \tag{3}$$

where negative samples are constructed as

$$S' = \{(h', r, t) | (h', r, t) \notin S\} \bigcup \{(h, r, t') | (h, r, t') \notin S\}.$$

Results in link prediction show that TransR outperforms other models on established benchmarks, indicating TransR is the best one-hop model at present. However, TransR fails to utilize the rich path information, which will be dealt with in the following subsection.

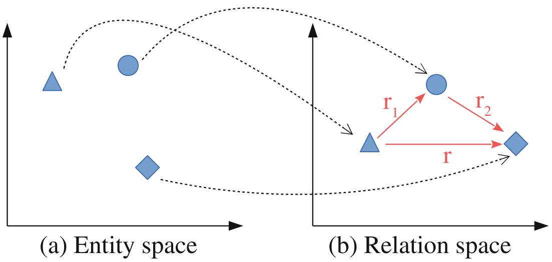

Fig. 1. An illustration of the PTransR model.

2.2 Path-Augmented TransR: PTransR

Using path information to regularize one-hop models can be beneficial [13,14]. Here, we adopt the path modeling method in PTransE, and extend the TransR model to path-augmented TransR (denoted as PTransR) (Fig. 1).

A relation path is a set of relations that connect head entity and tail entity in succession. An n-hop relation path from h to t is defined as $p = \{r_1, r_2, \cdots, r_n\}$, satisfying $h \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_n} t$. If $n = 1$, then $p = r_1$ is a direct (1-hop) relation. To enhance TransR model with multi-hop information, we follow the treatment in PTransE [14] and represent a relation path as an embedding vector by additive compositional methods. Then such multi-hop information is used to regularize one-hop direct relation between the same entity pair. A reliability score is computed to address the strength of regularization by a particular path. The details are described as follows.

To compute the representation of a relation path p that composites primitive relations $r_1, r_2, \cdots, r_n$, i.e., $p = r_1 \circ r_2 \circ \cdots \circ r_n$ (where $\circ$ denotes the composition operation), we add the embeddings of these primitive relations, given by

$$\mathbf{p} = \mathbf{r}_1 + \mathbf{r}_2 + \cdots + \mathbf{r}_n, \tag{4}$$

where bold letters denote the vector of a relation or a path.

The choice of addition as the composition operation is reasonable, because the vector representation of path p should be close to that of direct relation r if it is likely to infer r from p. For example, the representation of path $\xrightarrow{father}\xrightarrow{mother}$ is expected to be close to that of direct relation $\xrightarrow{grandmother}$.

Although a knowledge base may contain a variety of relation paths between two entities, not every path is equally useful for inferring direct relations. For example, the relation path $John \xrightarrow{friend} Tim \xrightarrow{gender} male$ gives little contribution to inferring the gender of $John$.

To evaluate the reliability of a path, PTransE uses a path-constraint resource algorithm (PCRA) [14], which is also applied in our approach. This algorithm first assigns a certain amount of resource (i.e., a value of 1) to the head entity h; then each node distributes resource evenly to its direct child nodes (Fig. 2). The value of p along an entity pair h, t is denoted as $v(p|h, t)$.

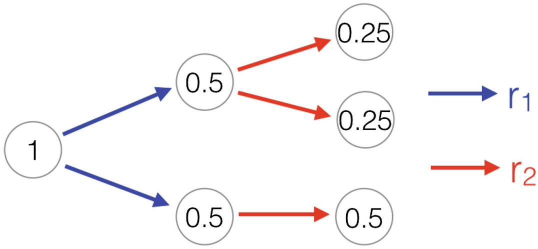

Fig. 2. An illustration of the path-constraint resource algorithm (PCRA).

However, $v(p|h,t)$ alone does not embody the relatedness between a relation path p and a direct relation r. To address this problem, a relatedness measure is defined as $P_r(r|p) = P_r(r,p)/P_r(p)$, where $P_r(p)$ is the sum of $v(p|h,t)$ for every training triplet (h,r,t) with p as a relation path from h to t. $P_r(r,p)$ is the sum of $v(p|h,t)$ for every training triplet (h,r,t) with r being a direct relation and p as a relation path. The overall reliability of a path p on a triplet (h,r,t) is given by

$$R(p|h,r,t) = P_r(r|p) \cdot v(p|h,t). \tag{5}$$

PTransR's scoring function $f_{\mathrm{PTransR}}(h,r,t)$ is composed of two scores:

$$f_{\mathrm{PTransR}}(h,r,t) = E(h,r,t) + E(\mathbf{P}|h,r,t). \tag{6}$$

$E(h,r,t)$ is the same as the scoring function of TransR (Eq. 2) which evaluates the plausibility of (h,r,t) without considering relation paths from h to t. Following PTransE, $E(\mathbf{P}|h,r,t)$ is defined as

$$E(\mathbf{P}|h,r,t) = \frac{1}{Z}\sum_{p\in\mathbf{P}} E(p|h,r,t), \tag{7}$$

$$E(p|h,r,t) = R(p|h,r,t)\|\mathbf{p}-\mathbf{r}\|_2^2 = P_r(r|p)v(p|h,t)\|\mathbf{p}-\mathbf{r}\|_2^2, \tag{8}$$

where Z is a normalizing factor for $R(p|h,t)$ and $\mathbf{P}$ is the set of all paths from h to t. $E(\mathbf{P}|h,r,t)$ evaluates the plausibility of (h,r,t) with the consideration of relation paths from h to t. The overall loss function of PTransR is

$$\mathcal{L}_{\mathrm{PTransR}} = \sum_{(h,r,t)\in S} [L(h,r,t) + \frac{1}{Z}\sum_{p\in\mathbf{P}} L(p|h,r,t)], \tag{9}$$

$$L(h,r,t) = \sum_{(h',r,t')\notin S} \max\left\{0, \gamma_1 + E(h,r,t) - E(h',r,t')\right\}, \tag{10}$$

$$L(p|h,r,t) = \sum_{(h,r',t)\notin S} \max\left\{0, \gamma_2 + E(p|h,r,t) - E(p|h,r',t)\right\}, \tag{11}$$

PTransR learns entity and relation embeddings by minimizing $\mathcal{L}_{\mathrm{PTransR}}$.

2.3 Training Details

We train PTransR by mainly following PTransE [14].

Initial Vectors and Matrices. Following TransR, initial vectors and matrices for PTransR are obtained from TransE. The configuration of TransE is: margin $\gamma = 1$, learning rate $\alpha = 0.01$, method = unif, and epoch = 1000.

Negative Samples. We sample negative triplets by randomly replacing head entity h or tail entity t or relation r. For example, (h, r, t)-derived negative triplets are (h', r, t), (h, r, t'), and (h, r', t), where $(h, r, t) \in S$ and (h', r, t), (h, r', t), $(h, r, t') \notin S$.

Vector Representation Constraints. Following TransR, to regularize the representations, we impose the following constraints on the entity and relation embeddings.

$$\| \mathbf{h} \| = \| \mathbf{t} \| = \| \mathbf{r} \| = 1, \| \mathbf{M}_r \mathbf{h} \| \leq 1, \| \mathbf{M}_r \mathbf{t} \| \leq 1. \tag{12}$$

Path Selection. PTranE restricts the length of path to less than 3. Its results show that 3-hop paths do not make significant improvement, compared to 2-hop paths. For efficiency, we only consider 2-hop relation paths.

Inverse Relation. As inverse relations sometimes contain useful information, for each training triplet (h, r, t), (t, r^{-1}, h) is added to the training set.

3 Evaluation

In this section, we present results of our experiment. We first briefly introduce the dataset and the task of entity prediction. Then we show the experimental results and analyze the performance.

3.1 Dataset

FB15k is a commonly used dataset in knowledge base completion. Table 1 shows statistics of FB15k. FB15k dataset contains factual information in our world, e.g., `location/country/language_spoken`. As FB15k has various kinds of relations, it is suitable for the evaluation of PTransR. Therefore, we choose FB15k as our experimental dataset.

Table 1. Statistics of the FB15k dataset.

Dataset	Relation	Entity	Train	Valid	Test
FB15k	1,345	14,951	483,142	50,000	59,071

3.2 Experimental Settings

We evaluate PtransR on the task of entity prediction. Entity prediction aims at predicting the missing entity in an incomplete triplet, i.e., predicting h given r and t, or predicting t given h and r. Following the settings in TransE, for a triplet (h, r, t), we replace the head entity h with every entity e and compute the score of (e, r, t). Entity candidates are ranked according to their scores. We repeat the same process to predict the tail entity t. Then we use the two metrics in TransE to evaluate the performance: MeanRank (average rank of the expected entity) and Hits@10 (proportion of triplets whose head/tail entity is among top-10 in the ranking). However, there could be several entities that are plausible for the same incomplete triplet. The plausible entities which are ranked before h or t may cause underestimation of performance. One solution is to remove other plausible entities in the ranking, which is referred to as a *filter*. In comparison, the results without removing other plausible entities are referred to as *raw*. A good model should achieve low MeanRank and high Hits@10.

To utilize the inverse relation, instead of only using the score $f_{\mathrm{PTransR}}(h, r, t)$, we use the sum of $f_{\mathrm{PTransR}}(h, r, t)$ and $f_{\mathrm{PTransR}}(t, r^{-1}, h)$ to rank the candidates, i.e.,

$$score(h, r, t) = f_{\mathrm{PTransR}}(h, r, t) + f_{\mathrm{PTransR}}(t, r^{-1}, h). \tag{13}$$

To accelerate the testing process, we use the reranking method in PTransE. We first rank all candidates according to their scores which are computed by the scoring function of TransR, which means that path information is not considered in the first ranking. Then we rerank the top-500 candidates according to the scores computed by the scoring function mentioned above, namely $score(h, r, t)$.

The configurations for experiments are given as follows: learning rate α for SGD among $\{0.01, 0.001, 0.0001\}$, dimension of entity space $\mathbb{R}^k$ and relation space $\mathbb{R}^d$ between $\{20, 50\}$, γ_1 and γ_2 among $\{1, 2, 4\}$, batch size B among $\{480, 960, 4800\}$. The optimal configuration on valid set is $\alpha = 0.001$, $k = d = 50$, $\gamma_1 = \gamma_2 = 1$, and $B = 4800$. The training process is limited to less than 500 epochs.

3.3 Overall Performance

Table 2 shows the experimental results. By comparing the results of PTransR with the results of previous models, we have the following main observations: (1) PTransR outperforms TransR on every metric to a large margin, which shows that path-augmented model can achieve better results than one-hop base model. (2) PTransR outperforms PTransE in MeanRank and is comparable to PTransE in Hits@10, which shows that path-augmented model with a better one-hop base model can achieve better performance.

Table 3 presents the performance on the four relation categories 1-to-1, 1-to-n, n-to-1, and n-to-n, with Hits@10(*filter*) as the metric. From Table 3, we find that, compared to TransR, PTransR shows consistent improvement on all four relation categories. Also, compared to PTransE, PTransR performs better on 1-to-1, 1-to-n and n-to-1 relations, especially on 1-to-n and n-to-1.

Table 2. Evaluation results of entity prediction on FB15k.

Metric	Mean rank		Hits@10(%)	
	Raw	Filter	Raw	Filter
Unstructured (Bordes et al. 2012)	1,074	979	4.5	6.3
RESCAL (Nickel et al. 2011)	828	683	28.4	44.1
SE (Bordes et al. 2011)	273	162	28.8	39.8
SME (linear) (Bordes et al. 2012)	274	154	30.7	40.8
SME (bilinear) (Bordes et al. 2012)	284	158	31.3	41.3
LFM (Jenatton et al. 2012)	283	164	26.0	33.1
TransE (Bordes et al. 2013)	243	125	34.9	47.1
TransH (unif) (Wang et al. 2014)	211	84	42.5	58.5
TransH (bern) (Wang et al. 2014)	212	87	45.7	64.4
TransR (unif) (Lin et al. 2015)	226	78	43.8	65.5
TransR (bern) (Lin et al. 2015)	198	77	48.2	68.7
CTransR (unif) (Lin et al. 2015)	233	82	44.0	66.3
CTransR (bern) (Lin et al. 2015)	199	75	48.4	70.2
PTransE (2-hop) (Lin et al. 2015)	200	54	51.8	83.4
PTransE (3-hop) (Lin et al. 2015)	207	58	51.4	**84.6**
PTransR (2-hop)	**171**	**47**	**53.0**	84.3

Table 3. Evaluation results of different relation catogories.

Tasks	Predicting head (Hits@10)				Predicting tail (Hits@10)			
Relation category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
Unstructured	34.5	2.5	6.1	6.6	34.3	4.2	1.9	6.6
SE	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (linear)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME (bilinear)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif)	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern)	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
CTransR (unif)	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR (bern)	81.5	89.0	34.7	71.2	80.8	38.6	90.1	73.8
PTransE (2-hop)	91.0	92.8	60.9	83.8	**91.2**	74.0	88.9	86.4
PTransE (3-hop)	90.1	92.0	58.7	**86.1**	90.7	70.7	87.5	**88.7**
PTransR (2-hop)	**91.4**	**93.4**	**65.5**	84.2	**91.2**	**74.5**	**91.8**	86.8

3.4 In-Depth Analysis and Discussion

As pointed out in Sect. 1, despite the massive train set of FB15k, some relations cannot be properly captured due to the problem of data sparsity. We separate relations into five groups according to their frequency in the train set, as shown in Table 4. MeanRank(*raw*) of TransR and PTransR is compared in Table 4 and the improvement from TransR to PTransR is presented. First of all, we see PTransR outperforms TransR in all five groups of relations. Second, as relation frequency decreases, the improvement goes up, which means that path information is useful for dealing with the problem of data sparsity.

Table 4. Evaluation results concerning relations of different frequency in train set.

Relation frequency in train set	1–3	4–15	16–50	51–300	>300
Relation number	291	305	243	271	235
MeanRank of TransR	159	98	54	81	202
MeanRank of PTransR	85	63	41	63	182
Improvement (%)	46.5	35.7	24.1	22.2	9.9

4 Related Work

Relation extraction is an important research topic in NLP. It can be roughly divided into two categories based on the source of information.

Text-based approaches extraction entities and/or relations from plain text. For example, Hearst [17] uses "is a|an" pattern to extract hyponymy relations. Banko et al. [6] proposes to extract open-domain relations from the Web. Fully supervised relation extraction, which classify two marked entities into several predefined relations, has become a hot research arena in the past several years [8,18,19].

Knowledge base completion/population, on the other hand, does not use additional text. Socher et al. [20] propose a tensor model to predict missing relations in an existing knowledge base, showing neural networks' ability of entity-relation inference. Then, translating embeddings approaches are proposed for knowledge base completion [7,11,12,14]. Recently, Wang et al. [21] use additional information to improve knowledge base completion by using textual context.

In this paper, we focus on pure knowledge base completion, i.e., we do not use additional resources. We combine the state-of-the-art one-hop TransR model [12] and path augmentation method [14], resulting in the new PTransR variant.

5 Conclusion

In this paper, we augment one-hop TransR model with path modeling method, resulting in PTransR model. We evaluate PTransR on the task of entity prediction and compare the performance of PTransR with that of previous models.

Experimental results show that path information is useful in solving the problem of data sparsity, and that PTransR outperforms previous models, which makes PTransR the state-of-the-art model in the field that populates knowledge base without using additional text.

Acknowledgments. This research is supported by the National Basic Research Program of China (the 973 Program) under Grant No. 2015CB352201 and the National Natural Science Foundation of China under Grant Nos. 61421091, 61232015 and 61502014.

References

1. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
2. 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, pp. 1247–1250 (2008)
3. Jiang, X., Tan, A.H.: Learning and inferencing in user ontology for personalized semantic web search. Inf. Sci. **179**(16), 2794–2808 (2009)
4. Yao, X., Van Durme, B.: Information extraction over structured data: question answering with freebase. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 956–966 (2014)
5. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. Nat. Genet. **25**(1), 25–29 (2000)
6. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 2670–2676 (2007)
7. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 1112–1119 (2014)
8. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1785–1794 (2015)
9. Chang, C.H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. IEEE Trans. Knowl. Data Eng. **18**(10), 1411–1428 (2006)
10. Zesch, T., Müller, C., Gurevych, I.: Extracting lexical semantic knowledge from Wikipedia and Wiktionary. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, pp. 1646–1652 (2008)
11. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2181–2187 (2015)

13. Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 318–327. Association for Computational Linguistics, September 2015
14. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 705–714 (2015)
15. Mikolov, T., Yih, W.T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751. Association for Computational Linguistics (2013)
16. Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., Jin, Z.: Natural language inference by tree-based convolution and heuristic matching. In: The 54th Annual Meeting of the Association for Computational Linguistics, p. 130 (2016)
17. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Conference on Computational Linguistics-Volume 2, pp. 539–545. Association for Computational Linguistics (1992)
18. dos Santos, C.N., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. arXiv preprint arXiv:1504.06580 (2015)
19. Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., Jin, Z.: Improved relation classification by deep recurrent neural networks with data augmentation. In: COLING (2016)
20. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934 (2013)
21. Wang, Z., Li, J.: Text-enhanced representation learning for knowledge graph. In: International Joint Conference on Artificial Intelligence, pp. 1293–1299 (2016)