

TransHExt: a Weighted Extension for TransH on Weighted Knowledge Graph Embedding

Kong Wei Kun¹, Xin Liu², Teeradaj Racharak¹ and Le-Minh Nguyen¹

¹*School of Information Science, Japan Advanced Institute of Science and Technology*

²*Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST)*

Abstract

Many methods have been proposed for embedding ordinary knowledge graphs (KG), which greatly promote various applications. However, current knowledge graph embedding algorithms cannot encode weighted knowledge graphs (WKG), which are a generalized form of the ordinary KG. This paper gives a promising approach that enables to extend existing models to encode weighted knowledge graphs. Taking TransH as an example, we propose TransHExt. TransHExt shows competitive performance in both link prediction task and weight prediction task.

Keywords

weighted Knowledge graph embedding, weighted extension

Knowledge graphs (KG) are thriving and promoting many downstream applications in semantic web such as question answering over RDF triples [1] and related areas such as academic search [2] and social relationship recognition [3]. Facts encoded in KG are formalized as triple (h, r, t) , in which h denotes a head entity, t denotes a tail entity, and r denotes a relation between h and t .

Weighted knowledge graphs (WKG) generalize (crisp) knowledge graphs by associating a weight in $(0, 1]$ to each triple. This formalism have been used to represent uncertainty [4], and even out-of-band knowledge [5] in a growing number of scenarios. The weighted triples can better model the interactions between the entities, such as the interactions of proteins in STRING [5].

Most knowledge graph embedding (KGE) models are targeted at learning the representation of KGs without weights. To leverage the existing models, such as TransH [6] to embed facts encoded in WKGs, we propose WeExt, an extension method that adds on a weight prediction module to a KGE model (called the base model). To illustrate our intuition, we select TransH and show its extension called TransHExt in this work. Our experiments show that TransHExt outperforms TransH on link prediction task and UKGE [7] on weight prediction task.

The 21st International Semantic Web Conference, October 23–27, 2022, Hangzhou, China


✉ kong.diison@jaist.ac.jp (K. W. Kun); xin.liu@aist.go.jp (X. Liu); racharak@jaist.ac.jp (T. Racharak); nguyenml@jaist.ac.jp (L. Nguyen)

🌐 <https://staff.aist.go.jp/xin.liu/> (X. Liu); <https://sites.google.com/view/teeradaj> (T. Racharak)

🆔 0000-0002-2958-9969 (K. W. Kun); 0000-0002-2336-7409 (X. Liu); 0000-0002-8823-2361 (T. Racharak); 0000-0002-2265-1010 (L. Nguyen)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

1. WeExt: Weighted Extension Framework

The framework of WeExt is shown in Figure 1. We introduce an add-on weight prediction module consisting of preprocessing and a neural weight predictor (denoted by nwp) to predict the weight for a given triple. The preprocessing is defined as the scoring function of a base model without the norm. We implement the neural weight predictor using a multi-layer feed-forward neural network.

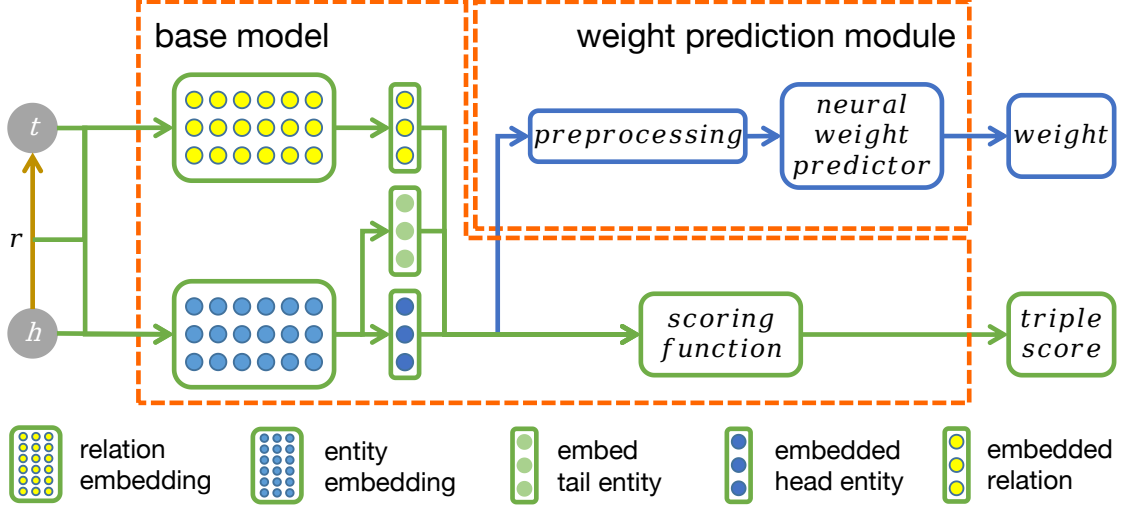


Figure 1: The framework of WeExt. The green components are the base KGE model. The blue components are the weight prediction component.

The preprocessing of the embedded entity and relation vary to the scoring function $f(\cdot)$ of the base KGE model. The preprocessing serves two purposes: one is to back-propagate the loss from the neural weight predictor to all learnable parameters, another is to make the loss from the scoring function and the loss from the neural weight predictor as consistent as possible in the adjustment of the parameters in the back-propagation. For TransH, the scoring function is:

$$s = -\|(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r)\|_2^2$$

where \mathbf{d}_r is the relation-specific translation vector and \mathbf{w}_r is the relation-specific hyperplane. Correspondingly, the predicted weight w_p can be calculated by the weighting prediction component is:

$$w_p = nwp((\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r))$$

We call this model TransHExt, i.e., extending TransH based on WeExt.

1.1. Training Protocol

For a given positive training set

$$S = \{(h_i, r_i, t_i), w_i\}_{i=0}^u,$$

we generate a negative set

$$S' = \{(h', r, t')\} = \{(h'_i, r_i, t_i) \mid h'_i \in E \setminus \{h_i\}\} \cup \{(h_i, r_i, t'_i) \mid t'_i \in E \setminus \{t_i\}\}_{i=0}^u.$$

We measure the accuracy of the neural weight predictor with respect to the error on the weight prediction in each triple of S :

$$\text{acc}(h, r, t, w) = \begin{cases} \frac{w - |w - w_p|}{w}, & w_p \in [0, 2w] \\ 0, & \text{otherwise} \end{cases}$$

We adopt margin ranking loss as the loss function for the proposed models:

$$\mathcal{L} = \sum_{\langle (h,r,t), w \rangle \in S} \sum_{(h',r',t') \in S'} \gamma + f(h, r, t) + \text{acc}(h, r, t, w) - f(h', r', t')$$

where γ is the margin. Since negative triples represent unencoded facts in the real world, it is not necessary to measure the loss of prediction on this set.

1.2. Evaluation Protocol

We evaluate the developed TransHExt on link prediction and weight prediction. For link prediction, given a test triple (h_i, r_i, t_i) , h_i is removed and the head entity is replaced by each of the entities of the dictionary in turn to form all possible triples. Triple scores are calculated by the scoring function $s = f(h, r_i, t_i)$. We sort s by ascending order, and get the rank of the i -th correct triple in its all possible triples rk_i . We adopt mean rank ($\text{MR} = \frac{1}{|u|} \sum_{i=1}^{|u|} \text{rk}_i$), mean reciprocal rank ($\text{MRR} = \frac{1}{|u|} \sum_{i=1}^{|u|} \frac{1}{\text{rk}_i}$) and Hits@N = $\sum_{i=1}^{|u|} \mathbb{I}[\text{rk}_i \leq N]$ to measure the performance of the models on link prediction. The $\mathbb{I}[\cdot]$ is the indicator function, which outputs 1 if the input is true, 0 otherwise. This whole procedure is repeated while removing t_i instead of h_i .

For weight prediction, we predict the weight of a given triple and report the mean squared error (MSE) and mean absolute error (MAE) between the predicted weight and the real weight.

2. Experiment and Result

We did experiments on CN15K, and PPI5K [8] datasets. CN15K is a subgraph of ConceptNet [4], containing 15,000 entities and 241,158 weighted triples in English. NL27k is extracted from NELL [9], a weighted KG obtained from webpage reading. NL27k contains 27,221 entities, 404 relations, and 175,412 weighted triples. PPI5k is a subset of the protein-protein interaction knowledge base STRING [5] that contains 271,666 weighted triples for 4,999 proteins and 7 interactions. STRING labels the interactions between proteins with the probabilities of occurrence.

We implemented the neural weight predictor using a 4-layer neural network with 50 inputs, three hidden layers with 300, 100, 50 neurons, and one output layer, also with ReLU activations. We set learning rate λ for the stochastic gradient descent optimizer to 0.0001. We trained the models for 3000 epochs, evaluated the models every 100 epochs, and chose the best result according to MRR. The margin γ was set to 1. The dimension of embeddings was set to 50.

Table 1
Results on Link Prediction

		MR	MRR	Hits@1	Hits@3	Hits@10
CN15K	TransH	1711	0.000585	0.042132	0.085516	0.140822
	TransHExt	1452	0.000689	0.035010	0.086012	0.155301
NL27K	TransH	322	0.003104	0.143687	0.270522	0.390837
	TransHExt	230	0.004351	0.085257	0.261615	0.430276
PPI5K	TransH	49	0.020226	0.002689	0.123945	0.310025
	TransHExt	26	0.038565	0.000093	0.158699	0.400097

Table 2
Results on Weight Prediction

	CN15K		NL27K		PPI5K	
	MSE	MAE	MSE	MAE	MSE	MAE
URGE [10]	10.32	22.72	7.48	11.35	1.44	6
UKGE	8.61	19.9	2.36	6.9	0.95	3.79
TransHExt	4.34	12.68	3.05	9.77	0.33	2.16

The results of link prediction and weight prediction are shown in Table 1 and Table 2, respectively. TransHExt outperforms TransH on link prediction in general, but gets a worse Hits@1 score on CN15K, NL27K and PPI5K. We assume this may be caused by the preprocessing. We will explore more preprocessing methods for WeExt. For weight prediction, TransHExt outperforms both of URGE and UKGE on CN15K and PPI5K, but gets a worse performance than UKGE on NL27K. We hypothesize this is caused by the different weight distribution of the dataset. We will explore more in our future work.

Acknowledgments

This work was supported by JST SPRING, Grant Number JPMJSP2102. This work was also partially supported by JSPS Grant-in-Aid for Early-Career Scientists (Grant Number 22K18004), JSPS Grant-in-Aid for Scientific Research (Grant Number 21K12042), and the New Energy and Industrial Technology Development Organization (Grant Number JPNP20006).

References

- [1] X. Hu, Y. Shu, X. Huang, Y. Qu, Edg-based question decomposition for complex question answering over knowledge bases, in: ISWC, 2021.
- [2] C. Xiong, R. Power, J. Callan, Explicit semantic ranking for academic search via knowledge graph embedding, WWW (2017).
- [3] Z. Wang, T. Chen, J. S. J. Ren, W. Yu, H. Cheng, L. Lin, Deep reasoning with knowledge graph for social relationship understanding, in: IJCAI, 2018.
- [4] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: AAAI, 2017.

- [5] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. C. J. Roth, A. Santos, K. Tsafou, M. Kuhn, P. Bork, L. J. Jensen, C. von Mering, String v10: protein–protein interaction networks, integrated over the tree of life, *Nucleic Acids Research* 43 (2015) D447 – D452.
- [6] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: *AAAI*, 2014.
- [7] X. Chen, M. Chen, W. Shi, Y. Sun, C. Zaniolo, Embedding uncertain knowledge graphs, *AAAI* (2019).
- [8] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: *AAAI*, 2018.
- [9] T. M. Mitchell, W. W. Cohen, E. R. Hruschka, P. P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. C. Wang, D. Wijaya, A. K. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling, Never-ending learning, *Communications of the ACM* 61 (2015) 103 – 115.
- [10] J. Hu, R. Cheng, Z. Huang, Y. Fang, S. Luo, On embedding uncertain graphs, *CIKM* (2017).