Re-evaluating Embedding-Based Knowledge Graph Completion Methods

Farahnaz Akrami
University of Texas at Arlington
farahnaz.akrami@uta.edu

Lingbing Guo
State Key Laboratory for Novel Software Technology,
Nanjing University
lbguo.nju@gmail.com

Wei Hu
State Key Laboratory for Novel Software Technology,
Nanjing University
whu@nju.edu.cn

Chengkai Li
University of Texas at Arlington
cli@uta.edu

ABSTRACT

Incompleteness of large knowledge graphs (KG) has motivated many researchers to propose methods to automatically find missing edges in KGs. A promising approach for KG completion (link prediction) is embedding a KG into a continuous vector space. There are different methods in the literature that learn a continuous representation of KG (latent features of KG). The benchmark dataset FB15k has been widely employed to evaluate these methods. However, It has been noted that FB15k contains many pairs of edges in which a pair represents the same relationship in reverse directions. Therefore, the inverse of numerous test triples occurs in the training set. To address this problem, FB15k-237, a subset of FB15k, was created by removing those inverse-duplicate relations to form a more challenging, realistic dataset. There is not any study that investigates how the aforementioned bias in this widely used benchmark dataset affects the results of embedding-based knowledge graph completion methods and whether their promising results are largely due to the bias. Motivated by this question, we conducted extensive experiments and report the link prediction results on FB15K and FB15k-237 using several embedding-based methods. We compare the results of different methods to see how their performances change in absence of inverse relations. Our experiment results demonstrate that the performance of embedding models in link prediction task diminishes tremendously when the inverse relationships do not exist anymore.

KEYWORDS

knowledge graph completion; link prediction; knowledge graph embedding

ACM Reference Format:
Farahnaz Akrami, Lingbing Guo, Wei Hu, and Chengkai Li. 2018. Re-evaluating Embedding-Based Knowledge Graph Completion Methods. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269266

1 INTRODUCTION

Large-scale knowledge graphs (KG) such as Freebase [2], DBpedia [1] and NELL [4] store real-world facts in the form of triples (head entity, relation, tail entity), denoted (h, r, t). They are an important resource for many AI-related applications such as question answering, web search, and fact checking, to name just a few. Despite their large sizes, KGs are usually far from complete, which hampers their usefulness in the aforementioned applications. To address this problem, various KG completion methods have been proposed. Existing methods can be categorized into two groups [11]: (1) latent feature models, also known as embedding models, such as TransE [3] and RESCAL [12] that embed a KG into a continuous vector space, (2) observed feature models that use observable properties of graphs, e.g., rule mining systems [6] and path ranking algorithms [8].

Among existing KG completion methods, embedding models that learn continuous representation of entities and relations have been quite popular. They usually embed an entity h (t) into a multidimensional vector $\mathbf{h}$ ($\mathbf{t}$) while a relation is represented as an operation (e.g., translation [3]) that combines $\mathbf{h}$ and $\mathbf{t}$. In TransE [3]—one of the simplest and most efficient embedding models with high accuracy—the embedding is learned in such a way that if (h, r, t) holds then $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ where $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ are the vector representations of h, r, and t, respectively.

Models such as TransH [16] and TransR [9] have been proposed to further address TransE's limitations such as its inadequacy in dealing with 1-to-n, n-to-1 and n-to-n relationships. Most embedding models were evaluated on the *link prediction* task—predicting the missing h or t in triple (?, r, t) or (h, r, ?). They were evaluated using a benchmark dataset FB15k, which is a subset of Freebase created by Bordes et al. [3]. Toutanova and Chen [14] noted that this widely-used dataset contains many inverse triples, i.e., it contains many pairs of (h, r, t) and (t, r^{-1}, h) where r^{-1} is inverse of r. With regard to link prediction, it seems unnecessary to learn embeddings of entities and relations given the existence of such inverse triples. A simple rule, r (h, t) $\leftarrow$ r^{-1} (t, h), can probably predict the missing h or t with better accuracy.

CIKM '18, October 22–26, 2018, Torino, Italy
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269266

In this paper, we aim to shed light on whether the embedding models are highly effective for link prediction in less straightforward scenarios. Toward this end, it is imperative to use a more challenging, realistic dataset. Toutanova and Chen [14] constructed such a dataset, FB15k-237, by excluding inverse relations from FB15k. Although embedding models have promising results on FB15k, no prior study provided a comprehensive investigation of their performance on FB15k-237. For instance, Toutanova and Chen [14] proposed an observed feature model NLFeat and compared it with only two embedding models E [13] and DistMult [17] on FB15k-237.

We conducted comprehensive experiments using a wide range of embedding-based knowledge graph completion methods on FB15k-237 and measured their performance using multiple standard metrics. We also reproduced their results on FB15k. The results of our experiments show that the good performance of current embedding models degrades significantly after removing inverse triples. Methods such as DistMult, ANALOGY [10] and ComplEx [15] have significantly outperformed TransE on FB15k but only attained similar or even worse performance on FB15k-237. For example, on FB15k, the FHits@10 (an often-used evaluation metric in [3] and others) of ANALOGY vs. TransE is 84.3% vs. 61.8%, while it is 37.4% vs. 42.5% on FB15k-237. Using another popular measure FMRR, ConvE [5] is the method with the best results on FB15k-237. However, its performance on FB15k was considerably stronger. These observations suggest the necessity of more research on embedding-based KG completion methods and evaluating them on a more challenging, realistic dataset.

2 BACKGROUND

Embedding-based methods employ two steps: (1) defining a scoring function to measure the plausibility of triples (*h*,*r*,*t*), and (2) learning the representations (i.e., embeddings) of entities and relations by solving an optimization problem of maximizing the scores of correct triples while minimizing the scores of incorrect ones. In TransE, the scoring function is:

$$f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|^2_{\ell_{1/2}} \tag{1}$$

which gives higher scores to correct triples and lower scores to incorrect ones.

TransH is similar to TransE but each relation is modeled as a vector $\mathbf{d}_r$ on a relation-specific hyperplane with $\mathbf{w}_r$ as normal vector where $\|\mathbf{w}_r\| = 1$, and the embeddings $\mathbf{h}$ and $\mathbf{t}$ are projected to the relation-specific hyperplane to obtain $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r$. In this way, the embeddings of entities are learned based on the relation that they are involved in.

TransR learns embeddings in two different vector spaces $\mathbb{R}^d$ and $\mathbb{R}^k$ for entities and relations, respectively. It defines a projection matrix $\mathbf{M}_r$ to map entity embedding to relation vector space. The score function of TransR is defined as:

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

In TransD [7], an improvement of TransR, the entities and the relation in a triple (*h*,*r*,*t*) are represented by two groups of vectors: one being $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$, and the other being $\mathbf{h}_p$, $\mathbf{r}_p$ and $\mathbf{t}_p$. The latter group is used to create the projection matrices:

$$\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p^\top + \mathbf{I} \qquad \mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p^\top + \mathbf{I} \tag{3}$$

As in TransR, these projection matrices are used to map entity vectors $\mathbf{h}$ and $\mathbf{t}$ to the relation vector space.

Besides these embedding methods that employ additive operation for composition of two entity vectors, there are bilinear approaches such as RESCAL, DistMult and ComplEx which use multiplicative operation (element-wise dot product). RESCAL represents a relation as a matrix which describes the interactions between latent representations of entities. The score of a triple in this method is defined as:

$$f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^\top \mathbf{W}_r \mathbf{t} \tag{4}$$

DistMult is similar to RESCAL but restricts relations to diagonal matrices to decrease the number of relation parameters. ComplEx is an extension of DistMult that uses complex numbers instead of real numbers to handle symmetric and anti-symmetric relations. ANALOGY further optimizes the embeddings of entities and relations with respect to their analogical properties. They argue that if two subgraphs g_1 and g_2 are analogous then missing triples in one of them can be inferred by their counterparts in the other one. ConvE is a neural network model that uses 2D convolutional layers over embeddings. NLFeat is a graph feature model that utilizes simple observed features of entities and relations. NeuralLp is a rule mining system which learns first-order logical rules from KGs. Although NLFeat and NeuralLP are not embedding-based methods, we include them in evaluation since they were among the few of which the results on FB15k-237 were reported in the literature.

3 EVALUATION

Motivated by the lack of a comprehensive comparison of performance of KG embedding models, we did extensive experiments to empirically compare the results of the most important models with and without inverse relations. The aim of our experiments is to investigate how much the removal of inverse relations affects the performance of these approaches.

3.1 Datasets

The experiments are conducted on two datasets FB15k and FB15k-237. FB15k, a subset of Freebase constructed by Bordes et al. [3], is extensively employed for evaluating KG embedding approaches. It contains 592,213 triples with 14,951 entities and 1,345 relations, which were randomly split into training, test and validation sets. As mentioned in Section 1, FB15k includes many pairs of inverse triples, e.g., (*x*, */film/actor/film./film/performance/film*, *y*) and (*y*, */film/film/starring./film/performance/actor*, *x*). Therefore, the inverse triples of numerous test triples exist in the training set. Toutanova and Chen [14] created FB15k-237 from FB15k by removing inverse and near-duplicate relations. They noted that the training dataset of FB15k has 81% test leakage as inverse relations, i.e., inverse triples of 81% of the test triples exist in the training set. The resulting FB15-237 contains 310,116 triples (reduced from 592,213 in FB15k) with 14,541 entities and 237 relations (reduced from 1,345 in FB15k).

3.2 Experiment Setups

Our experiments were conducted using source codes of various methods from several places, including the OpenKE repository [1]

[1] https://github.com/thunlp/OpenKE

which covers implementations of TransE, TransH, TransR, TransD, RESCAL, DistMult, and ComplEx, as well as the source codes released for ANALOGY [2] (which covers DistMult and ComplEx as well), ComplEx [3] (which covers DistMult as well), and ConvE [4] (which covers DistMult and ComplEx as well). All source codes and data used in our experiments are available at https://github.com/idirlab/kgcompletion.

To evaluate different embedding methods, the widely used link prediction task as described in [3] is used. The goal of link prediction is to predict missing h or t in a triple (h,r,t). For each test triple (h,r,t), the head entity h is replaced with each other entity in the dataset, to form *corrupted* triples. The original test triple and its corresponding corrupted triples are ranked by their dissimilarity scores according to the score functions mentioned in Section 2. The same procedure is repeated by replacing the tail entity t.

The accuracy of different embedding models is measured using Mean Rank (MR), Hits@10 and Mean Reciprocal Rank (MRR), as in [3]. MR is the mean of the test triples' ranks. Hits@10 is the percentage of test triples that are ranked within top 10. MRR is the average multiplicative inverse of the ranks of the test triples. Higher Hits@10 and MRR and lower MR are desirable. Besides these raw metrics, we also used their corresponding *filtered* metrics [3], denoted FMR, FHits@10, and FMRR, respectively. They are measured after removing those corrupted triples that appear in training, test or validation sets. In this way, a model is not penalized for ranking other correct triples higher than a test triple.

3.3 Results

Table 1 displays the results of link prediction on FB15k and FB15k-237 for all compared methods, using both raw and filtered metrics. The upward (downward, resp.) arrow beside a measure indicates that methods with greater (smaller, resp.) values by that measure possess higher accuracy. For each method, the table shows the original publication where it comes from. The values in black color are the results listed in the original publication, while a hyphen under a measure indicates that the original publication did not list the corresponding value. The values in other colors are obtained through our experiments using various source codes: blue for results obtained by implementations from the OpenKE repository; green for results from codes provided by Trouillon et al. [15] which introduced ComplEx and also supplied an implementation of DistMult; red for Liu et al. [10] which introduced ANALOGY and also provided implementations of DistMult and ComplEx; and orange for code from Dettmers et al. [5] which introduced ConvE and supplied DistMult and ComplEx implementations.

Below we summarize and explain the results in Table 1 based on the the best performance achieved by our experiments.

(1) By conducting experiments on both FB15k and FB15k-237, our main goal is to see how the performance of different embedding models change after removal of inverse relations. The overall observation is that the performance of all methods worsens considerably on FB15k-237. For instance, the FMRR of ConvE—one of the best performing methods under many of the metrics—has decreased from 68.9 (on FB15k) to 31 (on FB15k-237), and its FMR also became much worse, from 51.2 to 277. This result suggests that embedding-based methods may only perform well in predicting the inverse relations. However, it is possible to leverage a simple rule-based approach to conduct link prediction when the inverse of a triple is available, as explained in Section 1. Besides being simple, a rule-based system has another advantage in that it can be applied to entities unavailable in the training set. On the other hand, embedding models do not enjoy such a trait. The results of two observed feature models, NLFeat and NeuarlLP, demonstrate their strength in link prediction on FB15k. NLFeat has the highest FMRR among all the embedding models.

(2) Many methods were proposed as superior successors of TransE—the very first embedding model for KG completion—and indeed outperformed TransE on FB15k. However, by raw metrics MR, Hits@10 as well as filtered metric FMR, they all have worse results than TransE on FB15k-237, except for almost equal performance in a couple of cases. By raw metric MRR and filtered metric FHits@10, still they were outperformed by TransE on FB15k-237, except for ConvE. In terms of FMRR, methods such as DistMult, ComplEx, and ANALOGY significantly outperformed TransE on FB15k: TransE has an FMRR of 30.7, in comparison with DistMult (70.5), ComplEx (72.4) and ANALOGY (72.2). But their performance advantage over TransE under FB15k-237 is much smaller: TransE with an FMRR of 18.0, in comparison with DistMult (29.6), ComplEx (28.6) and ANALOGY (21.3). We hypothesize that these methods actually improved the results of link prediction for inverse triples and hence, after removing those triples they could not make much improvement on link prediction results. Yang et al. [17] empirically showed that embeddings learned by using multiplicative models significantly outperform additive models, which is also verified by our experiment results on FB15k. However the results on FB15k-237 do not show high superiority of them over additive models. This suggests their limitations in modeling the less straightforward portion of KGs.

(3) ConvE has the best performance under many metrics. However, there is a wide margin between its performance on FB15k and FB15k-237. For instance, its FMRR on FB15k vs. FB15k-237 is 68.9 vs. 31. This implies the necessity of more improvements of embedding models on FB15k-237. Most of the previous embedding models tried to be straightforward and efficient by using few parameters and simple operators. In this way they will be easy to train and hence scalable on large KGs. However, as the results of our experiments show, this simplicity decreases their true power in modeling KGs. On the other hand, increasing the number of parameters can lead to overfitting. Dettmers et al. [5] argued that deep, multi-layer models such as ConvE have better modeling power for complex graphs and the experiment results appear to verify it. ConvE employs a multi-layer convolutional neural network, for which different techniques exist to avoid overfitting while training the model. This probably explains why ConvE had the best overall results.

(4) For ComplEx and DistMult, we have presented the results from several different implementations which have quite different performance values. Similarly many of the implementations in our experiments produced better results than those reported in

[2] https://github.com/quark0/ANALOGY

[3] https://github.com/ttrouill/complex

[4] https://github.com/TimDettmers/ConvE

Table 1: Link Prediction Results

	FB15k						FB15k-237					
	Raw			Filtered			Raw			Filtered		
Model	MR↓	Hits@10↑	MRR↑	FMR↓	FHits@10↑	FMRR↑	MR↓	Hits@10↑	MRR↑	FMR↓	FHits@10↑	FMRR↑
TransE [3]	243.0 201.0	34.9 43.4	— 18.44	125.0 70.2	47.1 61.8	- 30.7	- 440.2	- 29.8	- 11.9	- 250.8	- 42.5	- 18.0
TransH [16]	211.0 213.8	42.5 47.3	- 28.3	84.0 69.3	58.5 70.1	- 16.3	- 511.8	- 29.0	- 10.5	- 309.8	- 42.9	- 16.3
TransR [9]	226.0 236.4	43.8 47.2	- 16.2	78.0 82.7	65.5 71.9	- 29.7	- 544.9	- 27.9	- 9.9	- 337.0	- 42.9	- 16.2
TransD [7]	211.0 209.8	49.4 47.4	- 16.3	67.0 65.4	74.2 70.4	- 28.3	- 506.9	- 29.4	- 10.4	- 305.2	- 42.8	- 16.2
RESCAL [12]	828.0 374.7	28.4 31.1	- 15.2	683.0 220.4	44.1 47.2	- 28.3	- 850.6	- 19.8	- 10.0	- 640.8	- 31.6	- 18.0
DistMult [17]	- 315.0 269.6 279.0 -	- 45.3 50.6 50.0 -	- 20.4 24.6 25.5 -	- 161.6 112.3 120.4 89.9	57.7 70.9 83.3 84.2 81.3	35 41.8 65.4 70.5 64.8	- 993.7 708.8 708.4 -	- 12.4 18.0 22.1 -	- 5.5 7.9 11.7 -	- 783.1 494.0 495.4 391.7	- 25.3 35.2 37.6 46.1	- 13.2 17.5 21.5 29.6
ComplEx [15]	- 347.6 266.2 292.7 -	- 44.3 48.5 49.2 -	24.2 20.4 23.0 24.9 -	- 189.5 106.0 132.9 97.5	84.0 73.0 82.6 82.5 79.2	69.2 51.3 67.5 72.4 62.3	- 1169.2 630.7 708.5 -	- 8.2 18.7 21.1 -	- 3.9 8.1 11.3 -	- 955.1 415.7 495.1 456.5	- 20.7 36.9 36.7 45.7	- 10.9 18.4 20.9 28.6
ANALOGY [10]	- 279.4	- 50.5	25.3 26.0	- 120.9	85.4 84.3	72.5 72.2	- 715.9	- 21.9	- 11.5	- 502.7	- 37.4	- 21.3
ConvE [5]	- 190.8	- 52.5	- 27.2	64.0 51.2	87.3 85.1	74.5 68.9	- 489.3	- 28.4	- 15.4	246.0 277.0	49.1 48.5	31.6 31.0
NLFeat [14]	-	-	-	-	87.0	82.2	-	-	-	-	34.7	22.6
NeuralLp [18]	-	-	-	-	83.7	76.0	-	-	-	-	36.2	24.0

● Published results ● OpenKE : https://github.com/thunlp/OpenKE ● ComplEx : https://github.com/ttrouill/complex
● ANALOGY : https://github.com/quark0/ANALOGY ● ConvE : https://github.com/TimDettmers/ConvE

the original publications. We attribute these differences to different dimensionalities of vectors representing entities and different optimization methods that were employed in the implementations.

4 CONCLUSIONS

In this paper, we showed that the impact of removing inverse triples is significant on performance of embedding models. This indicates the necessity of more improvements and research on KG completion methods that use embedding models. It also becomes apparent that a more challenging, realistic dataset is required for evaluating embedding models. FB15k-237 is a valuable dataset toward that goal but its size is much smaller than that of a real KG.

ACKNOWLEDGMENTS

Akrami and Li are partially supported by NSF grants IIS-1408928 and IIS-1719054. Guo and Hu are partially supported by the National Natural Science Foundation of China grant 61872172. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*. 1247–1250.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 2787–2795.

[4] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

[5] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*.

[6] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*. ACM, 413–422.

[7] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *ACL*. 687–696.

[8] Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning* 81, 1 (2010), 53–67.

[9] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion.. In *AAAI*. 2181–2187.

[10] Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical Inference for Multi-relational Embeddings. In *ICML*. 2168–2178.

[11] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2016), 11–33.

[12] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *ICML*. 809–816.

[13] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *ACL*. 74–84.

[14] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. 57–66.

[15] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*. 2071–2080.

[16] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*. 1112–1119.

[17] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

[18] Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *NIPS*. 2316–2325.