CEUR-WS.org/Vol-3447/Text2KG_Paper_5.pdf

A Comprehensive Methodology for Re-evaluation of Knowledge Graph Embedding Models

Bhushan Zope[1,*], Sashikala Mishra[1], Sanju Tiwari[2], Deepali Vora[1] and Ketan Kotecha[3]

[1] *Symbiosis Institute of Technology, Symbiosis International (Deemed University) (SIU), Lavale, Pune 412115, India*

[2] *Universidade Autonoma de Tamaulipas, Mexico*

[3] *Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis Institute of Technology, Symbiosis International (Deemed University) (SIU), Lavale, Pune 412115, India*

Abstract

Knowledge Graph (KG) has emerged as a favored tool in many areas of research and industry. One of the research areas in the KG domain is knowledge graph embedding, which involves mapping entities and relationships to low-dimensional vectors. Many knowledge graph embedding models have been proposed in the literature. However, minimal efforts have been made to investigate the reproducibility of these models. This research focuses on a reproducibility study of four state-of-the-art knowledge graph embedding models viz. CompGCN, NodePiece, PairRE, and TorusE. The PairRE results are 80% comparable to the corresponding published results on the Hit@10 parameter. On the other hand, for the MRR parameter, TorusE provided 95% comparable findings to results reported in the accompanying publication. This research has also demonstrated that reproducibility is a significant challenge in knowledge graph embedding research and highlighted the importance of transparency and standardization in this field.

Keywords

Knowledge Graph Embedding Models, Natural Language Processing, Knowledge Representation, Reproducibility, Reliability,

1. Introduction

Google, in 2012, proposed the knowledge graph as a large-scale knowledge base consisting of many entities and relationships among them. Recently KG has been viewed as a structured way of representing human knowledge by both industry and academia. Hence Knowledge graphs, big data, and deep learning are shaping the development of AI.

Most of the KGs are constructed manually or semi-automatically. Hence many entities and relations remain undiscovered. Thus incompleteness has become a most important drawback of KG [1, 2, 3] and makes it impractical to use in certain industrial applications. Knowledge

TEXT2KG 2023: Second International Workshop on Knowledge Graph Generation from Text, May 28 - Jun 1, 2023, co-located with Extended Semantic Web Conference (ESWC), Hersonissos, Greece

[*] Corresponding author.

bhushan.zope@hotmail.com (. B. Zope); sashikala.mishra@sitpune.edu.in (S. Mishra); sanju.tiwari.2007@gmail.com (S. Tiwari); deepali.vora@sitpune.edu (D. Vora); director@sitpune.edu.in (K. Kotecha)
0000-0003-2636-223X (. B. Zope); 0000-0002-5433-4917 (S. Mishra); 0000-0001-7197-0766 (S. Tiwari); 0000-0003-3969-9800 (D. Vora); 0000-0003-2653-3780 (K. Kotecha)

© 2023 Copyright for this paper Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Graph Completion (KGC) tries to solve this problem by identifying the missing entities or relations. Most of the research in KGC focuses on finding the low-level embedding for entities and relations. These models are called Knowledge Graph Embedding Models (KGEMs).

Many neural-network-based methods [4, 5, 6, 7, 8] have been proposed for KGEM, showing promising results. However, they have a complex scoring function in the form of a black-box neural network. Due to this black-box nature, it becomes difficult sometime to correlate the architectural changes with better performance. To alleviate these problems, Ali et al. [9] has performed extensive experimentation over 21 KGEM models with various training approaches, loss functions, and many other hyperparameters. However, four latest KGEM models, viz. TorusE[10], PairRE[11], NodePiece[12], and CompGCN[13] are not considered in that research work.

Keeping research work by Ali et al. [9] as a base for our study, we have performed similar experimentation on those four KGEM models. Hence this study aims to replicate the same result under similar circumstances to examine whether embedding models are particularly successful for link prediction tasks. Thorough experimentation is done on those KGEMs on four widely popular datasets, and reported the results on four well-known performance metrics. Pykeen library has been used for the implementation. Our experimentation results show that reproducing the results published in respective research papers is difficult, and performance varies drastically with slight changes in hyper-parameters. These observations point to the need for more study on KGEM algorithms and their evaluation on a complex, practical dataset.

This paper is organized in the following way: Section 2 introduces the basics of KG and its embedding concepts along with discussion on interaction models. Section 3 discusses the experimental setup and a detailed discussion on evaluation metrics and the dataset. Section 4 presents the results of this experimentation along with a discussion and implication. Finally, the conclusion and future work is discussed in section 5.

2. Knowledge Graph Embedding Models

KGs are being widely used for storing knowledge in the form of vertices and edges. All the important entities in the text are treated as vertices, while the relation between two entities is expressed as the edge. Figure 1 shows the sample knowledge graph of drug discovery. Such KGs can then be processed to derive insights from the vast amount of heterogeneous data or to perform the inference on it. KGs thus possess the possibility of becoming the 'brains' of machines. Hence A collection of triples (h, r, t) may be used to describe KG, where h and t stand for the head node and tail node, respectively, and r indicates the relationship between them. If $\mathbb{E}$ is an entity collection, and $\mathfrak{R}$ is a collection of relations, KG can be defined as $\mathbb{K} \subset \mathbb{E} \times \mathfrak{R} \times \mathbb{E}$.

In KGEM, entities and relations are represented in vector space, preserving the latent structure, as demonstrated in figure 2. One can manipulate the KG entities and relations through vector manipulation. Due to simplicity in manipulation, KGEM has gained attention recently due to its applicability in recommendation systems [14, 15], information retrieval [16, 17], question answering [18] etc. In the subsequent section, a few interaction models that have been considered for the experimentation are explained.

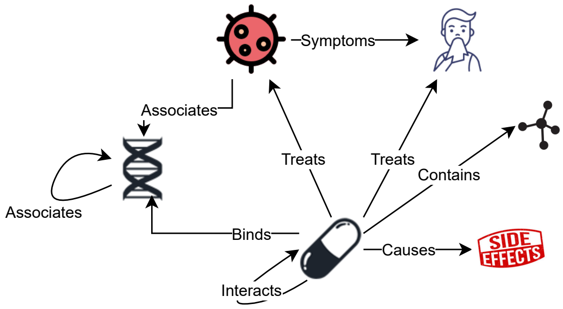


Figure 1: Model Knowledge Graph

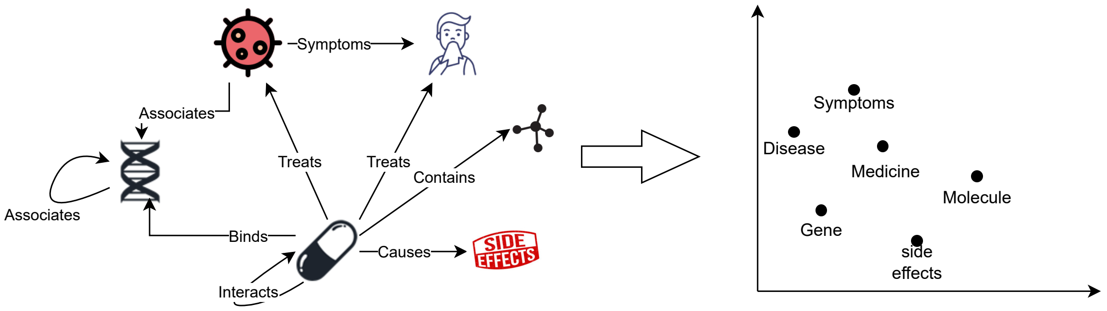


Figure 2: Knowledge graph embedding

2.1. Interaction model

Interaction models calculate the plausibility of the fact (h, r, t) when embedding for the head entity, relation, and the tail entity is given. Thus interaction model can be summarized as a mapping function $f : \mathbb{E} \times \mathfrak{R} \times \mathbb{E} \rightarrow \mathbb{R}$, that gives the plausibility score of the triple $(h, r, t) \in \mathbb{K}$ Generally, interaction models can be categorized into two, viz., 1. Translation models 2. Semantic matching models.

Translation Model finds the distance between entities and uses it as a scoring function. Figure 3 illustrates the translation principal, which models this problem as the minimization of $(h + r) - t$.
Canonical methods that come under this category are TransE[19] and its variants like TransR[20], TransH[21], TransD[22]; RotatE[23], HakE[24], MuRE[25], KG2E[26], PairRE[11]. All these methods follow the general principle of translation while adopting different representation spaces.

Semantic Matching Model preserves the latent semantic by using a similarity-based scoring function. Entities and semantically close relations are mapped to nearby points on vector space as demonstrated in figure 4. Usually, semantic matching is done using factorization or a neural network approach. RESCAL[27], ComplEx[28], QuatE[29], TuckER[30], DISTMULT[31], SimplE[32], etc. are few factorization based approaches. Whearas, ProjE[33], ERMLP[34],

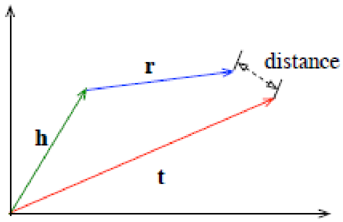


Figure 3: Translation Model: TransE[19]

NTN[35], ConvKB[4], ConvE[5], etc. are neural network based models.

For this study four latest KGEM models viz. TorusE, PairRE, CompGCN, and NodePiece are considered. In the following subsection, these models are explained in depth.

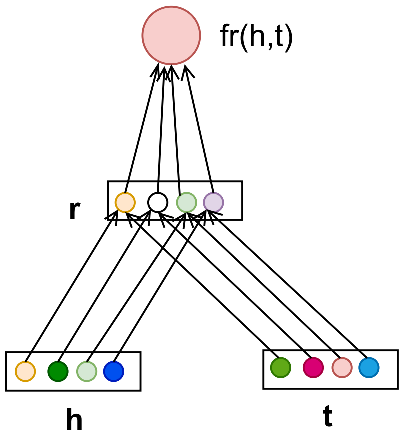


Figure 4: Semantic matching model: Distmult[31]

2.1.1. TorusE

Since regularization affects the TransE algorithm negatively as it forces embedding to be on a sphere of embedding space, reducing its link prediction capability [36], TorusE[10] avoids the regularization by projecting n-dimensional Euclidean space into n-dimensional torus space. TorusE also, as shown in figure 5, follows the underlying translational principle of TransE i.e., $[h] + [r] \approx [t]$. It uses equation 1 as a scoring function on the n-dimensional torus.

$$Min_{(x,y)\in([h]+[r])\times[t]} \|x - y\|_i \tag{1}$$

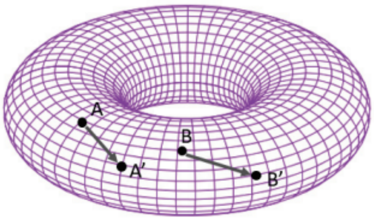


Figure 5: Entities A, A', B, and B' are represented on Torus Space [10]

2.1.2. PairRE

Two major challenges in embedding any relations are: 1. Handling the complex relations. 2. Preserving the inherent property of relation. To overcome both challenges, the PairRE model proposed in [11] considers relation vector as a vector pair $[r^H, r^T]$. As shown in figure 6, the Hadamard Product of these head and tail entity vectors project them in euclidean space and plausibility of the triple (h, r, t) is calculated from a distance between projected vectors. Thus the vector pair $[r^H, r^T]$ is adjusted so that $h \circ r^H \approx t \circ r^T$ if triple is present. Otherwise, $h \circ r^H$ and $t \circ r^T$ should not be close. Thus scoring function becomes the minimization of equation 2

$$f_r(h, t) = -\left\| h \circ r^H - t \circ r^T \right\| \tag{2}$$

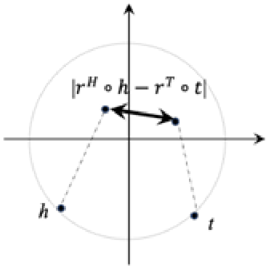


Figure 6: Entities are projected on the unit circle. They are projected inside the circle using the relation vectors. [11]

2.1.3. CompGCN

Most of the multi-relation GCN methods suffer from the over-parameterization problem. CompGCN [13], which is a generalization of multi-relation GCN, alleviates these problems by using the same embedding space for entities and relations. Subsequently, using many composition operators on them.

Basically CompGCN views the KG as $(\mathbb{E}, \mathfrak{R}, \mathbb{Z}, \mathbb{X})$, where $\mathbb{E}$ and $\mathfrak{R}$ are as defined earlier, while $\mathbb{Z} \in \mathbb{R}^{|\mathbb{E}| \times d_0}$ and $\mathbb{X} \in \mathbb{R}^{|\mathbb{R}| \times d_0}$ mean d_0 size input feature of entity and relation respectively.

GCN uses equation 3 to update the embedding for nodes. Since it doesn't involve relation-specific input features, it suffers from over-parameterization.

$$\mathbb{Z}_v = f(\sum_{(u,r)\in\eta(v)} W_r Z_u) \tag{3}$$

where $\eta(v)$: Neighbourhood of v
W_r : Trainable matrix representing all the relation types.
To reduce the problem of over-parameterization, CompGCN uses the composition operator Φ over the node in a neighborhood with respect to relation r. For better information flow, CompGCN assumes bi-directional edges in the graph. Hence, the relation set is expanded by adding inverse relation for each relation type. To incorporate these points, CompGCN uses equation 4.

$$\mathbb{Z}_v = f(\sum_{(u,r)\in\eta(v)} W_{\lambda(r)}\Phi(Z_u, X_r)) \tag{4}$$

where, $W_{\lambda(r)}$ specifies the relation type parameter.
Z_u and X_r are initial representation of u and r respectively.

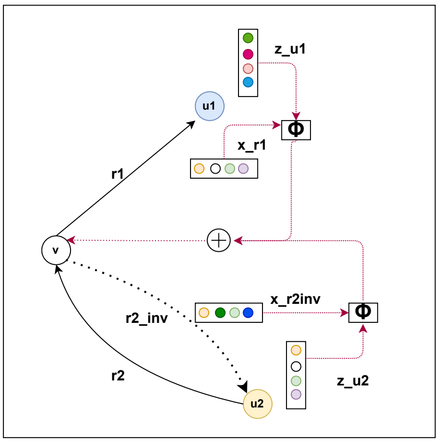


Figure 7: Working of CompGCN

Figure 7, explains the intuition of equation 4 in detail. As seen in 7, u1 and u2 are neighbors of v; hence they are considered for updating the node v's embedding. Equations 5 and 6 demonstrate the difference between the workings of GCN and CompGCN with respect to the graph given in figure 7.

$$\mathbb{Z}_v = f(W_r Z_{u1} + W_r Z_{u2}) \tag{5}$$

$$\mathbb{Z}_v = f(W_{\lambda(r1)}\Phi(Z_{u1}, X_{r1}) + W_{\lambda(r2_{inv})}\Phi(Z_{u2}, X_{r2_{inv}})) \tag{6}$$

2.1.4. Node Piece

Large Language Models (LMM) like BERT and GPT don't employ shallow embedding by finding the embedding for all the words. However, it learns the embedding for a few words and tries to find the embedding for other words by using the learned embedding. This reduces the number of parameters drastically.
Taking a cue from this, NodePiece [12] uses a few selected entities (anchor nodes) and all relations as vocabulary set $\mathbb{V}$. Hence, $|\mathbb{V}| \ll |\mathbb{E}|$. Similar to CompGCN, NodePiece also assumes bi-directional relations; hence inverse relation of each relation type is added to the relation set.

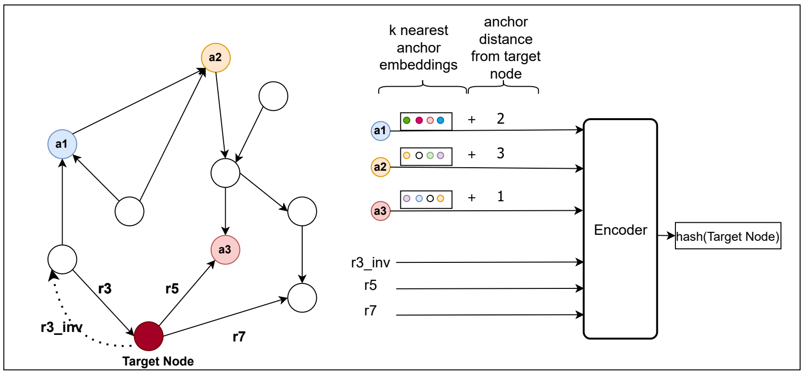


Figure 8: Working of NodePiece

Figure 8 explains the procedure to calculate the hash function for each entity. To calculate the hash for entity e, k closest anchors and all m incident relation types are considered. The encoder function for the same is given in equation 7.

$$hash(e) = \left[\{a_i\}^k, \{\Delta_{a_i}\}^k, \{r_j\}^m\right] \tag{7}$$

where
$\{a_i\}^k$ and $\{\Delta_{a_i}\}^k$ represents embedding for k anchors and their distance from target node respectively.
$\{r_j\}^m$ represents embedding for all m incident relations.

3. Experimental setup

Models selected for this study belong to different categories; separate experimental setups are considered by looking at the advantages and prerequisites of each model. KG is split into training and testing parts in every experimentation, and then Hyper-Parameter Optimization (HPO) is performed on TorusE, PairRE, CompGCN, and NodePiece models. As prescribed by Ali et al. [37], the detailed experimental setup is designed, which is also discussed in figure 9.

Datasets: Four Well known datasets are used in this research.

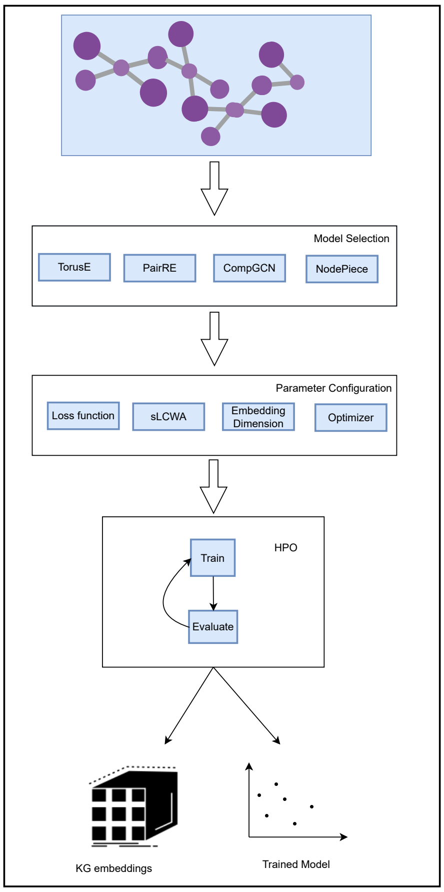


Figure 9: Experimental setup [37]

1. **Kinships**: The dataset Kinships[38] has 104 entities representing the people of some tribe, and those people are related to each other with 26 different relationships. It has 10,686 triples.

2. **Nations**: The Nations [39] dataset is one of the oldest knowledge graph datasets comprising 14 countries and their various relationships.

3. **WN18**: WN18 [40] dataset is part of WordNet dataset. There are 40,943 sunsets (i.e., entities) and 18 conceptual semantic relationships. WN18 has a test leakage, inverse relations present in WN18 are removed, and WN18RR[41] dataset is formed. Though WN18RR is a better version of WN18, most of the methods have used WN18; hence for consistency purposes, we have used the WN18 dataset for detailed analysis.

4. **FB15k-237**: Since FB15K [19] also has similar leakage issue, FB15k-237 [42] dataset is formed by removing the inverse relations. It has 14541 entities and 237 relations.

Table 1
Datasets

Dataset	Number of relation	Number of Entities	Number of triples
FB15K [19]	1345	14,951	592,213
FB15K-237 [42]	237	14,541	272,115
WN18 [40]	18	40,943	151,442
WN18RR [41]	11	40,559	92,583
Kinships [38]	26	104	10,686
Nations [39]	56	14	11,191

Table 1 summarizes the datasets. CompGCN and NodePiece are applicable to graphs with symmetric relations only. hence these datasets are extended by adding the inverse relations for those models only. This can be achieved by simply adding triple $(t, r_{inv}, h), \forall(h, r, t) \in \mathbb{K}$.

Evaluation Metric KGEMs are mostly evaluated using link prediction tasks. There are many metrics available to evaluate knowledge graph embedding; however, Mean Rank(MR), Adjusted Mean Rank (AMR), Mean Reciprocal Rank (MMR), and Hits@K are used more frequently in the literature.

Mean Rank : MR is the arithmetic mean of ranks of all triples $(h, r, t) \in \mathbb{T}_{eval}$. It is given in the equation 8:

$$MR = \frac{1}{|\mathbb{T}_{eval}|} \sum_{t \in \mathbb{T}_{eval}} rank(t) \tag{8}$$

Adjusted Mean Rank: As explained in [43], MR is flawed since getting a low rank with fewer possible candidates is easy. AMR neutralizes this by taking the ratio of MR with the expected mean rank. Thus making it useful to compare the results of two different-sized datasets.

Mean Reciprocal Rank: MRR, also known as Inverse harmonic mean rank, is the mean of the reciprocal of a rank. Thus defined as equation 9

$$MMR = \frac{1}{|\mathbb{T}_{eval}|} \sum_{t \in \mathbb{T}_{eval}} \frac{1}{rank(t)} \tag{9}$$

[44] and [45] argues that MMR is theoretically incorrect as ranks are in an ordinal scale and finding their reciprocal is wrong. However, these arguments are countered by [46]. Moreover, MMR is a frequently used metric. Unlike the Hits@K metric, it doesn't ignore the changes in high-rank values. At the same time, unlike MR, it is more sensitive to changes in low-rank values than high-rank values. Thus it gives more importance to small ranks while remaining less affected by outliers.

Hits@K: Hits@K (Mostly K= {1,3,5,10}) is a simple metric that measures the ratio of test triples appearing in top K entries. Mathematically it can be represented as equation 10.

$$Hits@K = \frac{\{t \in \mathbb{T}_{eval} | rank(t) \leqslant k\}}{|\mathbb{T}_{eval}|} \tag{10}$$

One of the biggest drawbacks of the Hits@K metric is it considers the entries appearing in top k ranks but ignores all the cases where $rank(t) > K$. In consequence, it doesn't make any difference to Hits@K whether $rank(t) = K + 1$ or $K + m$ where m » 1. Thus Hits@K is practically useless for comparing different models. However, most of the published articles include Hits@K as an evaluation metric. Hence, this metric has been used for reproducibility study.
With this experimental setup, a series of trials have been performed. The following section discusses the outcome of these trials.

4. Result and Discussions

This research objective was to assess whether the findings of four current Knowledge Graph Embedding techniques were reproducible and provide comprehensive results of these models for their comparability with newer models. In order to assess reproducibility, this study replicated the key findings from ToursE, PairRE, CompGCN, and NodePiece. Analysis revealed that results are reproducible only in some cases, indicating a substantial challenge in reproducing results across scientific research. This section is divided into two parts. Section 4.1 compares the obtained results with published results. It also discusses the probable reasons for the deviations in results. Section 4.2 gives the results[1] for the selected models for various datasets and for different evaluation metrics.

4.1. Comparison with published results

The PyKEEN framework, a Python toolkit for KGEMs, has been used for all work. In addition, the Optuna package is used for hyperparameters optimization. The evaluation metrics used are rank-based. Hence it becomes very necessary to select the strategy to break a tie. Due to the absence of any strategy reported in the selected publications, the 'realistic rank' method on 'head' and 'tail' prediction is used.

In this research, 160 experiments, ten each for every model-dataset pair, were carried out. Each experiment is repeated multiple times for better reporting of a study. Moreover, CompGCN and NodePiece methods are also used to train a model on the WN18RR dataset. But results of WN18RR and WN18 were found to be similar. Table 2 shows the best results [2] obtained for the respective setup. For the FB15k237 dataset, results for the PairRE method are 80% and 70% of published results for Hit@10 and MRR, respectively. Moreover, with 95% comparable results, the TorusE method has also performed exceptionally well with respect to the MRR parameters. However, the results were not near to the published results for other methods. Reproducing the same result under similar conditions is challenging. Also, due to different implementations and various interpretations about the link prediction task evaluation metric, it becomes difficult

[1] The source code and results obtained have been made available at https://github.com/bhushan-zope/Reproducibility-Study

[2] Results in bracket uses WN18RR dataset

Table 2

Results on WN18/WN18RR and FB15K237 dataset; where **O** represents Obtained results while **P** refers to published results. Moreover, results are not available for WN18 of PairRE model

Model		WN18 (WN18RR)		FB15K237	
		Hits @10	MRR	Hits @10	MRR
TorusE	O	0.127	0.080	0.400	0.242
	P	0.954	0.619	0.832	0.256
PairRE	O	0.891	0.734	0.437	0.248
	p	NA	NA	0.544	0.351
CompGCN	O	0.008 (0.006)	0.004 (0.001)	0.298	0.165
	P	0.546	0.479	0.535	0.355
NodePiece	O	0.107 (0.027)	0.048 (0.007)	0.289	0.182
	p	0.515	0.403	0.420	0.256

to compare two previously published results [9]. This difference in the observed and reported results could be attributed to various factors discussed below.

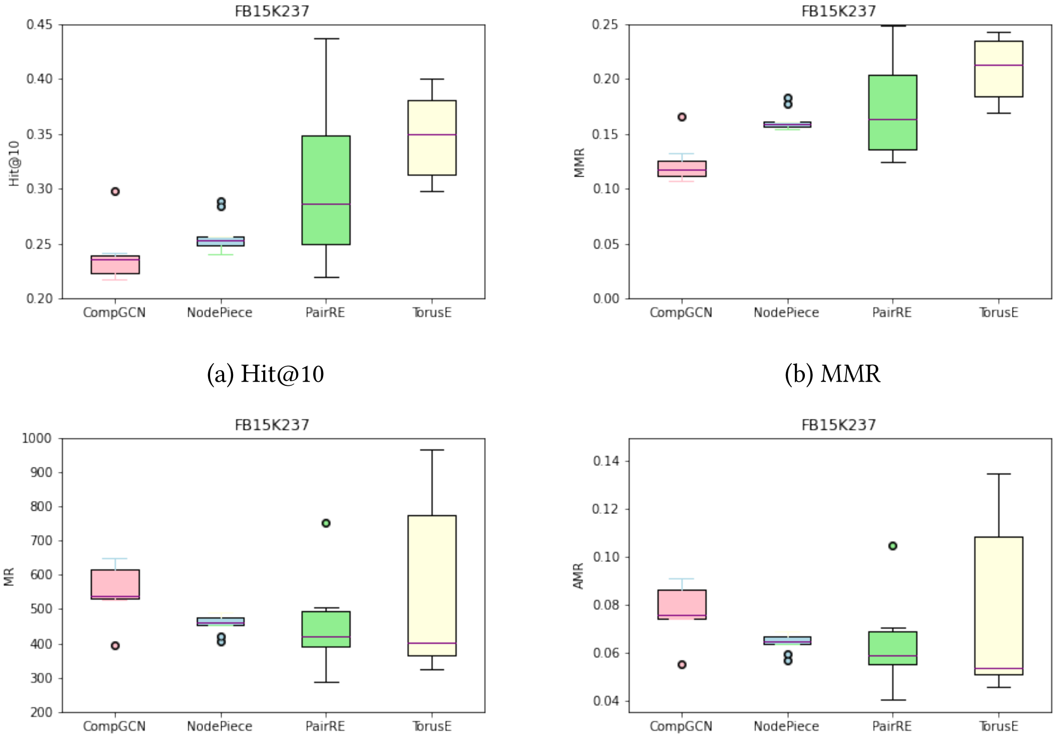
1. **Difference in ranking approach:** As discussed in [47], [48], various authors have implemented the ranking metrics differently. If more than one triple has the same rank, which should be ranked higher, it is an important question to answer and affects the overall scores. Yet, no author has declared their ranking approach in the publication. However, in this research, a realistic ranking approach is used, which is the mean of the pessimistic and optimistic ranking approaches.

2. **Difference in the implementation:** This complete study is based on the PYKEEN library, whereas all the respective authors have implemented the methods independently.

3. **Missing finer details:** In most cases, the hyper-parameter values have not been reported. Additionally, some publications have used grid search for hyper-parameters. However, the best configuration is not reported, leaving scope for various possible combinations to experiment with. These hyper-parameters initialized to different values can significantly affect the results [9].

Additionally, WN18 is based on WordNet, which consists of English words and their semantic relationships. It mainly consists of semantically similar relations. Hence due to *Humongous relations* (i.e., all the relations of the same type) and *Data Imbalance* (i.e., some relations occur more frequently), the results of all the methods are better for the FB15K-237 dataset than the WN18 dataset.

Moreover, figure 10 represents the central tendency of the results obtained. From figure 10a and figure 10b, it is evident that the results for CompGCN and NodePiece are away from the central tendency. The interquartile range(IQR) indicates the reliability of the results. More points clustered around the median give a narrow IQR indicating that the data is more reliable or mored representative of the population. Hence, wide IQR for TorusE (as shown in figure 10c and figure 10d) indicates that the results obtained might not be very reliable; on the other hand, narrow IQR for CompGCN, NodePiece, and PairRE(up to a certain extent) makes their results more reliable. A highly skewed distribution around the median also indicates the results' poor quality.



(a) Hit@10

(b) MMR

(c) MR

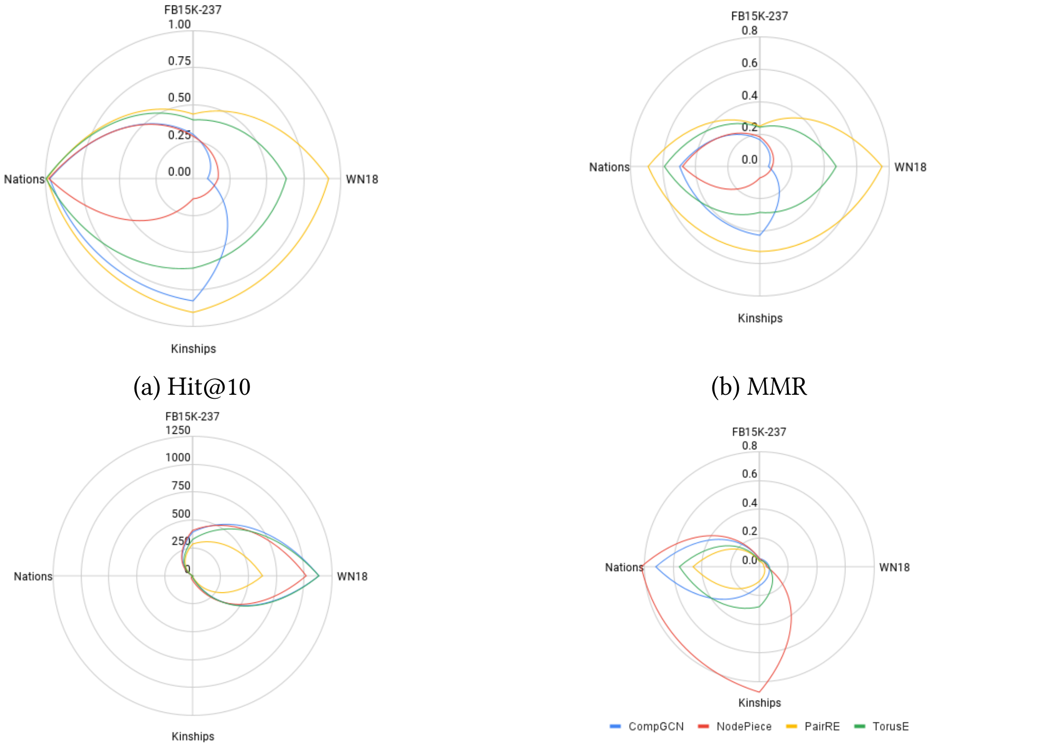
(d) AMR

Figure 10: Performance Distribution of all models for each parameter on FB15K237 dataset

4.2. Comprehensive results

Due to the availability of various performance parameters and datasets, researchers in the KGEM domain have used various combinations of them. As a result, it becomes practically impossible to compare the two KGEMs. To eliminate this problem, another objective of this study was to present the result of selected models on more datasets with various parameters. Section 3 explains the considered datasets and evaluation metrics in detail. This section discusses the results of ToursE, PairRE, CompGCN, and NodePiece methods with those datasets and evaluation metrics.

Results listed in appendix A show that the models' performance varied significantly depending on the dataset and the performance parameter being used. Figure 11 shows the complete results of all the models on all four datasets using all four parameters. The blue line in figure 11 represents the results of CompGCN, whereas the red, yellow, and green lines represent the results for NodePiece, PairRE, and TorusE, respectively.



(a) Hit@10

(b) MMR

(c) MR

(d) AMR

Figure 11: Comprehensive results of all models on all datasets using parameters (a) Hit@10, (b) MMR, (c) MR, (d) AMR.

Unlike MR and AMR, larger values represent better results for Hit@10 and MMR parameters. Thus in figure 11a and 11b, the yellow line covers most of the area; at the same time, in figure 11c and 11d, the yellow line covers the smallest area compared to the other three lines representing an excellent performance of PairRE method on all the datasets. Similar observations can be drawn on other models from figure 11. Moreover, It is surprising to notice the inconsistent gain on some datasets. For example, in PairRE, there is ≈a 30% increase in the hit@10 metric compared to TorusE on WN18 and Kinships dataset. However, there is only a 3% increase on the FB15K237 dataset and a 1% decrease on the Nations dataset. This also leads to less confidence in the methods.

Overall, the findings can be summarized as follows:

1. Results of PairRE method were satisfactorily close to the results mentioned in its publication Chao et al. [11]. However, the results of other methods were difficult to reproduce.
2. Compared to the other three methods, PairRE performed outstandingly well on all datasets considering all metrics.

3. CompGCN and NodePiece performed worst on WN18 dataset. Nevertheless, PairRE achieved exceptional results on the same dataset.

4. TorusE is more suitable for smaller datasets and its performance degrades for more complex datasets.

These findings suggest that the choice of model and performance parameter can significantly impact the effectiveness of knowledge graph embedding models. Our study provides valuable insights for researchers and practitioners seeking to use these models for knowledge graph completion tasks.

5. Conclusion

In this work, knowledge graph embedding models' reproducibility was examined. The study's main objective was to replicate the findings of four cutting-edge embedding models on four benchmark datasets. Findings showed that several factors, such as the selection of hyperparameters, the evaluation dataset, and the implementation specifics, significantly impact the reproducibility of these models. In order to ensure reproducibility, the study emphasizes the significance of including thorough explanations of the experimental setup and hyperparameters in research articles. It also highlights the necessity of uniformity in embedding model evaluation to enable comparison and benchmarking across various research.

In summary, the reproducibility analysis of knowledge graph embedding models offers insightful information on the challenges and opportunities of this emerging research area. This work will serve as a catalyst for future studies to enhance the consistency and dependability of knowledge graph embedding models, which will ultimately result in better comprehension and implementation of knowledge graphs in many contexts.

References

[1] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, Z. Duan, Knowledge graph completion: A review, IEEE Access 8 (2020) 192435–192456.

[2] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2724–2743. doi:10.1109/TKDE.2017.2754499.

[3] S. Tiwari, F. N. Al-Aswadi, D. Gaurav, Recent trends in knowledge graphs: theory and practice, Soft Computing 25 (2021) 8337–8355.

[4] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, D. Q. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: NAACL-HLT (2), 2018, pp. 327–333. URL: https://aclanthology.info/papers/N18-2053/n18-2053.

[5] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018.

[6] L. Guo, Z. Sun, W. Hu, Learning to exploit long-term relational dependencies in knowledge graphs, ArXiv abs/1905.04914 (2019).

[7] X. Jiang, Q. Wang, B. Wang, Adaptive convolution for multi-relational learning, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 978–987. URL: https://aclanthology.org/N19-1103. doi:10.18653/v1/N19-1103.

[8] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, D. Q. Phung, A capsule network-based embedding model for knowledge graph completion and search personalization, CoRR abs/1808.04122 (2018). URL: http://arxiv.org/abs/1808.04122. arXiv:1808.04122.

[9] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp, J. Lehmann, Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[10] T. Ebisu, R. Ichise, Toruse: Knowledge graph embedding on a lie group, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2018.

[11] L. Chao, J. He, T. Wang, W. Chu, PairRE: Knowledge graph embeddings via paired relation vectors, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 4360–4369. URL: https://aclanthology.org/2021.acl-long.336. doi:10.18653/v1/2021.acl-long.336.

[12] M. Galkin, E. Denis, J. Wu, W. L. Hamilton, Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=xMJWUKJnFSw.

[13] S. Vashishth, S. Sanyal, V. Nitin, P. Talukdar, Composition-based multi-relational graph convolutional networks, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=BylA_C4tPr.

[14] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2724–2743. doi:10.1109/TKDE.2017.2754499.

[15] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2000–2010. URL: https://doi.org/10.1145/3308558.3313411. doi:10.1145/3308558.3313411.

[16] R. Reinanda, E. Meij, M. de Rijke, Knowledge Graphs: An Information Retrieval Perspective, 2020.

[17] S. Tiwari, F. Ortiz-Rodriguez, B. Villazon, Guest editorial: Special issue on "current topics of knowledge graphs and semantic web", International Journal of Web Information Systems 18 (2022) 237–239.

[18] B. Zope, S. Mishra, K. Shaw, D. R. Vora, K. Kotecha, R. V. Bidwe, Question answer system: A state-of-art representation of quantitative and qualitative analysis, Big Data and Cognitive Computing 6 (2022). URL: https://www.mdpi.com/2504-2289/6/4/109. doi:10.3390/bdcc6040109.

[19] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 26, Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[20] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, AAAI Press, 2015, p. 2181–2187.

[21] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, Proceedings of the AAAI Conference on Artificial Intelligence 28 (2014). URL: https://ojs.aaai.org/index.php/AAAI/article/view/8870. doi:10.1609/aaai.v28i1.8870.

[22] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 687–696. URL: https://aclanthology.org/P15-1067. doi:10.3115/v1/P15-1067.

[23] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=HkgEQnRqYQ.

[24] Z. Zhang, J. Cai, Y. Zhang, J. Wang, Learning hierarchy-aware knowledge graph embeddings for link prediction, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 3065–3072. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5701. doi:10.1609/aaai.v34i03.5701.

[25] I. Balažević, C. Allen, T. Hospedales, Multi-relational poincar\'e graph embeddings, in: Advances in Neural Information Processing Systems, 2019.

[26] S. He, K. Liu, G. Ji, J. Zhao, Learning to represent knowledge graphs with gaussian embedding, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 623–632. URL: https://doi.org/10.1145/2806416.2806502. doi:10.1145/2806416.2806502.

[27] M. Nickel, V. Tresp, H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, Omnipress, Madison, WI, USA, 2011, p. 809–816.

[28] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, 2016, p. 2071–2080.

[29] S. Zhang, Y. Tay, L. Yao, Q. Liu, Quaternion Knowledge Graph Embeddings, Curran Associates Inc., Red Hook, NY, USA, 2019.

[30] I. Balazevic, C. Allen, T. Hospedales, TuckER: Tensor factorization for knowledge graph completion, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5185–5194. URL: https://aclanthology.org/D19-1522. doi:10.18653/v1/D19-1522.

[31] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6575.

[32] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, p. 4289–4300.

[33] B. Shi, T. Weninger, Proje: Embedding projection for knowledge graph completion, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, AAAI Press, 2017, p. 1236–1242.

[34] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 601–610. URL: https://doi.org/10.1145/2623330.2623623. doi:10.1145/2623330.2623623.

[35] R. Socher, D. Chen, C. D. Manning, A. Y. Ng, Reasoning with neural tensor networks for knowledge base completion, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, p. 926–934.

[36] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 494–514. doi:10.1109/TNNLS.2021.3070843.

[37] M. Ali, H. Jabeen, C. T. Hoyt, J. Lehmann, The keen universe, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham, 2019, pp. 3–18.

[38] W. W. Denham, The detection of patterns in alyawarra nonverbal behavior, 2014.

[39] R. J. Rummel, Dimensionality of nations project: Nation attribute data, 1950-1965, 1992. URL: https://doi.org/10.3886/ICPSR05020.v1. doi:10.3886/ICPSR05020.v1.

[40] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data, Machine Learning 94 (2014) 233–259. URL: https://doi.org/10.1007/s10994-013-5363-6. doi:10.1007/s10994-013-5363-6.

[41] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Association for Computational Linguistics, Beijing, China, 2015, pp. 57–66. URL: https://aclanthology.org/W15-4007. doi:10.18653/v1/W15-4007.

[42] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Association for Computational Linguistics, Beijing, China, 2015, pp. 57–66. URL: https://aclanthology.org/W15-4007. doi:10.18653/v1/W15-4007.

[43] M. Berrendorf, E. Faerman, L. Vermue, V. Tresp, Interpretable and fair comparison of link prediction or entity alignment methods with adjusted mean rank, CoRR abs/2002.06914 (2020). URL: https://arxiv.org/abs/2002.06914. arXiv:2002.06914.

[44] N. Fuhr, Some common mistakes in ir evaluation, and how they can be avoided, SIGIR

Forum 51 (2018) 32–41. URL: https://doi.org/10.1145/3190580.3190586. doi:10.1145/3190580.3190586.

[45] S. S. Stevens, On the theory of scales of measurement, Science 103 (1946) 677–680. URL: https://www.science.org/doi/abs/10.1126/science.103.2684.677. doi:10.1126/science.103.2684.677. arXiv:https://www.science.org/doi/pdf/10.1126/science.103.2684.677.

[46] T. Sakai, On fuhr's guideline for ir evaluation, SIGIR Forum 54 (2021). URL: https://doi.org/10.1145/3451964.3451976. doi:10.1145/3451964.3451976.

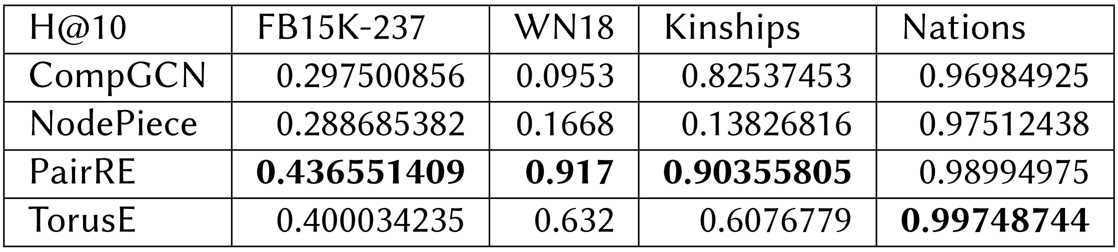
[47] Z. Sun, S. Vashishth, S. Sanyal, P. Talukdar, Y. Yang, A re-evaluation of knowledge graph completion methods, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5516–5522. URL: https://aclanthology.org/2020.acl-main.489. doi:10.18653/v1/2020.acl-main.489.

[48] M. Berrendorf, E. Faerman, L. Vermue, V. Tresp, Interpretable and fair comparison of link prediction or entity alignment methods with adjusted mean rank, CoRR abs/2002.06914 (2020). URL: https://arxiv.org/abs/2002.06914.

A. Appendix

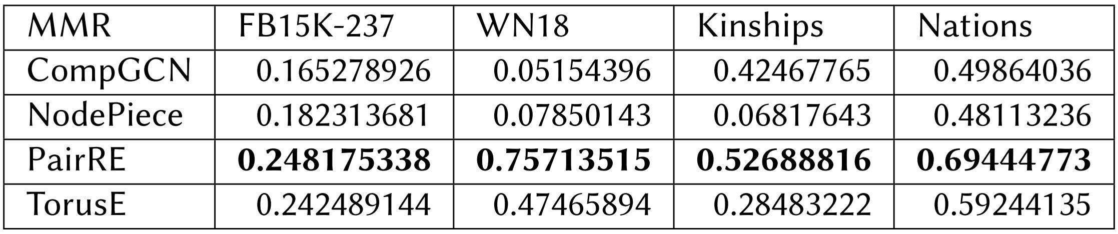
Result Table:

Table 3
Hit@10



H@10	FB15K-237	WN18	Kinships	Nations
CompGCN	0.297500856	0.0953	0.82537453	0.96984925
NodePiece	0.288685382	0.1668	0.13826816	0.97512438
PairRE	**0.436551409**	**0.917**	**0.90355805**	0.98994975
TorusE	0.400034235	0.632	0.6076779	**0.99748744**

Table 4
MMR



MMR	FB15K-237	WN18	Kinships	Nations
CompGCN	0.165278926	0.05154396	0.42467765	0.49864036
NodePiece	0.182313681	0.07850143	0.06817643	0.48113236
PairRE	**0.248175338**	**0.75713515**	**0.52688816**	**0.69444773**
TorusE	0.242489144	0.47465894	0.28483222	0.59244135

Table 5
MR

MR	FB15K-237	WN18	Kinships	Nations
CompGCN	394.4117126	1130.12012	6.31086159	3.5351758
NodePiece	405.4782715	1021.547	41.70298	3.67412925
PairRE	**288.2659912**	**629.351074**	**4.74485016**	**2.24623108**
TorusE	324.5093994	1132.15527	13.3066483	2.71859288

Table 6
AMR

AMR	FB15K-237	WN18	Kinships	Nations
CompGCN	0.055199518	0.06918933	0.13077983	0.7217235
NodePiece	0.056748328	0.04992046	0.87392746	0.82055553
PairRE	**0.040343994**	**0.03075482**	**0.09832741**	**0.45857911**
TorusE	0.045416406	0.06007955	0.27575335	0.55501409