Knowledge Graph Embeddings for Link Prediction: Beware of Semantics!

Nicolas Hubert, Pierre Monnin, Armelle Brun, Davy Monticolo

Knowledge Graph Embeddings for Link Prediction: Beware of Semantics!

Nicolas Hubert[1,2,*], Pierre Monnin[3], Armelle Brun[1] and Davy Monticolo[2]

[1]Université de Lorraine, CNRS, LORIA, France
[2]Université de Lorraine, ERPI, France
[3]Orange, France

Abstract

The task of predicting links in knowledge graphs (KGs) can be tackled using knowledge graph embedding models (KGEMs). Such models project entities and relations of a KG into a low-dimensional vector space that preserves as much as possible the properties of the graph. The performance of KGEMs for link prediction is traditionally assessed using rank-based metrics that evaluate the ability of models to give high scores to ground-truth entities. However, other scored entities are left unconsidered by these metrics. This constitutes a shortcoming in some application domains where it may be required to ensure consistency among the top-scored entities. To this aim, in this paper we propose to measure the ability of popular KGEMs to capture the semantic profile of relations. In particular, we use Sem@K, a semantic-oriented metric that assesses whether top-scored entities are semantically valid. Our experiments show that agnostic KGEMs are actually able to learn the semantic profile of relations. This raises the opportunity of using Sem@K as an additional training criterion.

Keywords

Knowledge Graph Embeddings, Link Prediction, Rank-Based Metrics, Semantic-Oriented Metrics

1. Introduction

A knowledge graph (KG) is a collection of triples (h, r, t) where h (head) and t (tail) are two entities of the graph, and r is a predicate (also called relation) that qualifies the relationship holding between them. KGs support several tasks including entity matching, question answering, and link prediction [1]. The latter is the focus of this paper. Given a triple $(?, r, t)$ (resp. $(h, r, ?)$), link prediction (LP) consists in predicting the most plausible head h (resp. tail t). Knowledge Graph Embedding Models (KGEMs) address this particular task by projecting entities and relations of the KG into a low-dimensional vector space that preserves as much as possible the properties of the graph [2]. Training a KGEM firstly requires corrupting existing triples by replacing either their head h or their tail t with another entity to generate negative counterparts.

DL4KG 2022: Workshop on Deep Learning for Knowledge Graphs, held as part of ISWC 2022: the 21st International Semantic Web Conference, October 23 - 27, 2022

*Corresponding author.
nicolas.hubert@loria.fr (N. Hubert); pierre.monnin@orange.com (P. Monnin); armelle.brun@loria.fr (A. Brun); davy.monticolo@univ-lorraine.fr (D. Monticolo)
0000-0002-4682-422X (N. Hubert); 0000-0002-2017-8426 (P. Monnin); 0000-0002-9876-6906 (A. Brun); 0000-0002-4244-684X (D. Monticolo)

CEUR Workshop Proceedings (CEUR-WS.org)

This procedure is called negative sampling [3, 4, 5]. Secondly, the KGEM iteratively learns to assign higher scores to true triples than to their negative counterparts.

The performance of Knowledge Graph Embedding Models (KGEMs) for LP is ultimately evaluated using rank-based metrics such as Hits@K, Mean Rank (MR), and Mean Reciprocal Rank (MRR) that evaluate whether ground-truth entities are indeed given higher scores [1, 2]. However, various works recently raised some caveats about such metrics [6, 7, 8]. Indeed, they not only lack a theoretical grounding (see Section 2.1) but are also not well-suited for drawing comparisons across datasets [6]. More importantly, they only provide a partial picture of KGEM performance [6]. Indeed, LP can lead to nonsensical triples, such as (`BarackObama`, `isFatherOf`, `USA`), being predicted as highly plausible facts, although they violate constraints on the domain and range of relations [5, 9]. KGEMs with such issues may nevertheless reach a satisfying performance in terms of rank-based metrics since these violations are not taken into account by the metrics.

Few works propose to go beyond the mere traditional quantitative performance of KGEMs and address their ability to capture the semantics of the original KG [10, 11, 12]. This is why we advocate for additional qualitative and semantic-oriented metrics to supplement traditional rank-based metrics. According to Berrendorf *et al.* [6], this would give a more complete picture of the performance of a KGEM. More precisely, such semantic-oriented metrics allow to investigate dimensions that remain unexplored in the literature and yet deserve greater attention. For example, given the domain and range of a relation, such metrics would allow to assess the need for post-filtering candidate entities that do not belong to the expected types. Indeed, if the KGEM is not able to assign higher ranks to entities of the correct types without supervision, a post-filtering phase may be needed to ensure consistency in the predictions.

Accordingly, in this work, our goal is to assess the ability of popular KGEMs to capture the semantic profile (i.e., domain and range) of relations in a link prediction task. To do so, we build on Sem@K, the semantic-oriented metric that we introduced in [13] for a recommendation task. In this previous work, Sem@K was used to evaluate the ability of KGEMs to recommend items that are of the expected type. Here, we extend the scope of the metric to fit the more generic LP task.

The remainder of the paper is structured as follows. Related work is presented in Section 2. In Section 3, we detail the KGEMs used in this work as well as the semantic-oriented metric Sem@K that we tailor for the LP task. Dataset descriptions, experimental settings and key findings are provided in Section 4. Lastly, Section 5 outlines future directions.

2. Related Work

2.1. Evaluating KGEM Performance for Link Prediction

KGEM performance is almost exclusively assessed using the following rank-based metrics: Hits@K, Mean Rank (MR), and Mean Reciprocal Rank (MRR) [7]. We recall their definitions and discuss their limits below. The use of such metrics stems from the fact that training and evaluating a KGEM requires generating negative triples [7]. Thus, positive triples are scored against negative ones to determine whether the model is able to predict plausible facts. More specifically, given a ground-truth triple (h, r, t), all possible triples $(?, r, t)$ and $(h, r, ?)$ are

generated with all the entities observed in the KG. Then, such triples are scored by the KGEM and their scores are compared with the score given to the ground-truth triple.

Hits@K (Equation (1)) accounts for the proportion of ground-truth triples appearing in the first K top-scored triples:

$$\text{Hits@}K = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \mathbb{1} \left[\text{rank}(q) \leq K \right] \tag{1}$$

where $\mathcal{B}$ is the batch of ground-truth triples, $\text{rank}(q)$ is the position of the ground-truth triple q in the sorted list of triples, and $\mathbb{1} \left[\text{rank}(q) \leq K \right]$ yields 1 if q is ranked between 1 and K, 0 otherwise. This metric is bounded in the $[0, 1]$ range and its values increases with K, where the higher the better.

Mean Rank (MR) (Equation (2)) corresponds to the arithmetic mean over ranks of the ground-truth triples:

$$\text{MR} = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \text{rank}(q) \tag{2}$$

This metric is bounded in the $[0, |\mathcal{E}|]$ interval, where $|\mathcal{E}|$ stands for the number of entities in the KG, where the lower the better.

Mean Reciprocal Rank (MRR) (Equation (3)) corresponds to the arithmetic mean over the reciprocals of ranks of the ground-truth triples:

$$\text{MRR} = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \frac{1}{\text{rank}(q)} \tag{3}$$

Contrary to MR, MRR is a metric bounded in the $[0, 1]$ interval, where the higher the better. Because this metric does not use any threshold K compared to Hits@K, it is less sensitive to outliers. In addition, it is often used for performing early stopping and for tracking the best epoch during training [6, 7].

As mentioned in Section 1, these metrics present some caveats. LP is often used in a knowledge base completion perspective, where the Open World Assumption (OWA) prevails. KGs are incomplete and, due to the OWA, an unobserved triple used as a negative one can still be positive. It follows that traditional evaluation methods based on rank-based metrics may systematically underestimate the true performance of a KGEM [9]. In addition, the aforementioned rank-based metrics have intrinsic and theoretical flaws, as pointed out in several works [6, 7, 8]. For example, Hits@K does not take into account triples whose rank is larger than K. As such, a model scoring the ground-truth in position $K + 1$ would be considered equally good as another model scoring the ground-truth in position $K + d$ with $d \gg 1$. It follows that Hits@K is not a suitable metric for drawing comparisons between models [7]. MR alleviates this concern as it does not consider any threshold K. Therefore, MR allows to compare KGEM performance on the same dataset. Nonetheless, MR is sensitive to the number of KG entities (see Equation (2)) [6]: a MR of 10 indicates very good performance if the set of entities is in the thousands, but it would indicate poor performance if the set of entities is much more restricted. Therefore, MR does not allow comparisons across datasets.

2.2. Combining Embeddings with Semantics

The possibility of using additional semantic information to enhance KGEM performance has been extensively studied. A significant part of the literature incorporates semantic information to constrain the negative sampling procedure and generate meaningful negative triples [5, 4]. For instance, type-constrained negative sampling (TCNS) [4] replaces the head or the tail of a triple with a random entity belonging to the same type as the ground-truth entity. Jain *et al.* [5] go a step further and use ontological reasoning to iteratively improve KGEM performance by retraining the model on inconsistent predictions. Semantic information can also be embedded in the model itself [14, 15]. In [15], the proposed KGEM embeds both entities and entity types, which allows entities to have different vector representations depending on their respective types.

Recall that embedding models project entities and relations of a KG into a vector space. As such, the semantics of the original KG may not be fully preserved [5, 10]. As stated by Paulheim [10], because embeddings are not meant to preserve the semantics of the KG, they are not interpretable and this can severely hinder explainability in domains such as recommender systems. Consequently, Paulheim [10] advocates for *semantic embeddings*. Similarly, Jain *et al.* [11] perform a thorough evaluation of popular KGEMs to better assess their semantic awareness. A key finding is that in most datasets, the semantic representation of some entities is easier to grasp than for other ones. For instance, the task of finding semantically similar entities does not always provide satisfying results when working with entity embeddings [11].

Although some aforementioned approaches leverage the semantics of entities and relations to improve KGEM performance in terms of rank-based metrics, their ability to generate sensical predictions is not directly addressed. This encourages further assessment of the semantic capabilities of KGEMs. In our work, we directly address this issue by assessing to what extent KGEMs are able to give high scores to triples whose head (resp. tail) belongs to the domain (resp. range) of the relation.

3. Measuring Semantic Awareness: our Proposal

Section 3.1 summarizes the KGE models used in this work. Then, Section 3.2 details the semantic-oriented metric used to experimentally evaluate and compare these models and highlight their semantic awareness.

3.1. Knowledge Graph Embedding Models

As in [16], we study three highly popular KGEMs, namely TransE [3], DistMult [17], and ComplEx [18].

TransE is the earliest translational model. It learns representations of entities and relations such that for a triple (h, r, t), $\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{e}_t$, where $\mathbf{e}_h$, $\mathbf{e}_r$ and $\mathbf{e}_t$ are the head, relation and tail embeddings, respectively. The scoring function is $f(h, r, t) = -d(\mathbf{e}_h + \mathbf{e}_r - \mathbf{e}_t)$ with d a distance function, usually the $L1$ or $L2$ norm. TransE does not properly handle 1-to-N, N-to-1, nor N-to-N relations [1] and yet has been found to be very efficient in multi-relational settings [19].

DistMult is a semantic matching model. It is characterized as such because it uses a similarity-based scoring function and matches the latent semantics of entities and relations by leveraging their vector space representations. More specifically, DistMult is a bilinear diagonal model that uses a trilinear dot product as its scoring function: $f(h, r, t) = \langle \mathbf{e}_h, \mathbf{W}_r, \mathbf{e}_t \rangle$. It is similar to RESCAL [20] – the very first semantic matching model – but restricts relation matrices $\mathbf{W}_r \in \mathbb{R}^{d \times d}$ to be diagonal. As the scoring function of DistMult is commutative, all relations are considered symmetric. This assumption does not hold in general. However, DistMult still achieves state-of-the-art performance in most cases [21].

ComplEx is also a semantic matching model. It extends DistMult by using complex-valued vectors to represent entities and relations: $\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t \in \mathbb{C}^d$. As a result, ComplEx is better able to model antisymmetric relations than DistMult [22]. Its scoring function uses the Hadamard product: $f(h, r, t) = \mathrm{Re}\left(\mathbf{e}_h \odot \mathbf{e}_r \odot \overline{\mathbf{e}}_t\right)$ where $\overline{\mathbf{e}}_t$ denotes the conjugate of $\mathbf{e}_t$.

3.2. A Semantic-Oriented Metric for Measuring Link Prediction Performance

The standard LP evaluation protocol consists in reporting aggregated results, considering the rank-based metrics presented in Section 2.1. As mentioned in Section 1, these metrics only provide a partial picture of KGEM performance [6]. To give a more comprehensive assessment of KGEMs, we aim at jointly assessing their semantic awareness using Sem@K [13]. In [13], Sem@K is specifically defined for the recommendation task seen as predicting tails for a unique target relation. In this work, we extend Sem@K to the more generic LP task, where not only tails but also heads are corrupted and all relations are considered equally.

This adapted version of Sem@K (Equation (4)) accounts for the proportion of triples that are semantically valid in the first K top-scored triples:

$$\mathrm{Sem@}K = \frac{1}{|\mathcal{B}|} \sum_{q \in \mathcal{B}} \frac{1}{K} \sum_{q' \in \mathcal{S}_q^K} \mathrm{compatibility}(q, q') \tag{4}$$

where, given a ground-truth triple $q = (h, r, t)$, $\mathcal{S}_q^K$ is the top-K candidate triples scored by a given KGEM (i.e. by predicting the tail for $(h, r, ?)$ or the head for $(?, r, t)$). The operator $\mathrm{compatibility}(q, q')$ (Equation (5)) assesses whether the candidate triple q' is semantically compatible with its ground-truth counterpart q. In this work, by semantic compatibility we refer to the fact that the predicted head (resp. tail) belongs to the domain (resp. range) of the relation:

$$\mathrm{compatibility}(q, q') = \begin{cases} 1, & \text{if } \mathrm{type}(q'_h) = \mathrm{domain}(q_r) \wedge \mathrm{type}(q'_t) = \mathrm{range}(q_r) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\mathrm{type}(e)$ returns the type of entity e and $\mathrm{domain}(r)$ (resp. $\mathrm{range}(r)$) is the domain (resp. range) of the relation r. q_r, q'_h, and q'_t denote the ground-truth relation, the head and the tail of the ranked triple q', respectively. Note that type hierarchy is not considered in this work.

Sem@K is bounded in the $[0, 1]$ interval. Compared to Hits@K (Equation (1)), Sem@K is non-monotonic: increasing K can lead to either lower or higher Sem@K values.

Table 1
Characteristics of EduKG, FB15K-237, and KG20C datasets. $|\mathcal{E}|$, $|\mathcal{R}|$, and $|\mathcal{C}|$ stand for the number of entities, relations, and classes, respectively. $|\mathcal{T}_{train}|$, $|\mathcal{T}_{valid}|$, and $|\mathcal{T}_{test}|$ stand for the number of train, validation, and test triples, respectively.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{T}_{train}\rvert$	$\lvert\mathcal{T}_{valid}\rvert$	$\lvert\mathcal{T}_{test}\rvert$
EduKG	5,452	27	9	25,411	4,279	4,258
FB15K-237	14,541	237	532	271,575	15,360	17,960
KG20C	16,362	5	5	48,213	3,670	3,724

4. Experiments

In this section, we assess TransE, DistMult, and ComplEx performance in terms of MRR and Sem@K on real-world and public datasets.

4.1. Datasets

The experiments are carried out on EduKG[1] [23], FB15K-237 [24], and KG20C [25], three KGs that have been chosen for their adequate entity typing: given a head-relation pair (h, r) (resp. a relation-tail pair (r, t)), the missing tail (resp. head) can only be of one single type. In addition, these datasets comprise a different number of entities, relations, and entity types as depicted in Table 1. Consequently, they allow us to study whether the performance of a KGEM in terms of Sem@K is dataset-dependent. Note that relations with less than 10 semantically valid heads or tails are removed from the test set, so that Sem@10 is not wrongfully lowered by an insufficient number of candidates for either the domain or range of such relations.

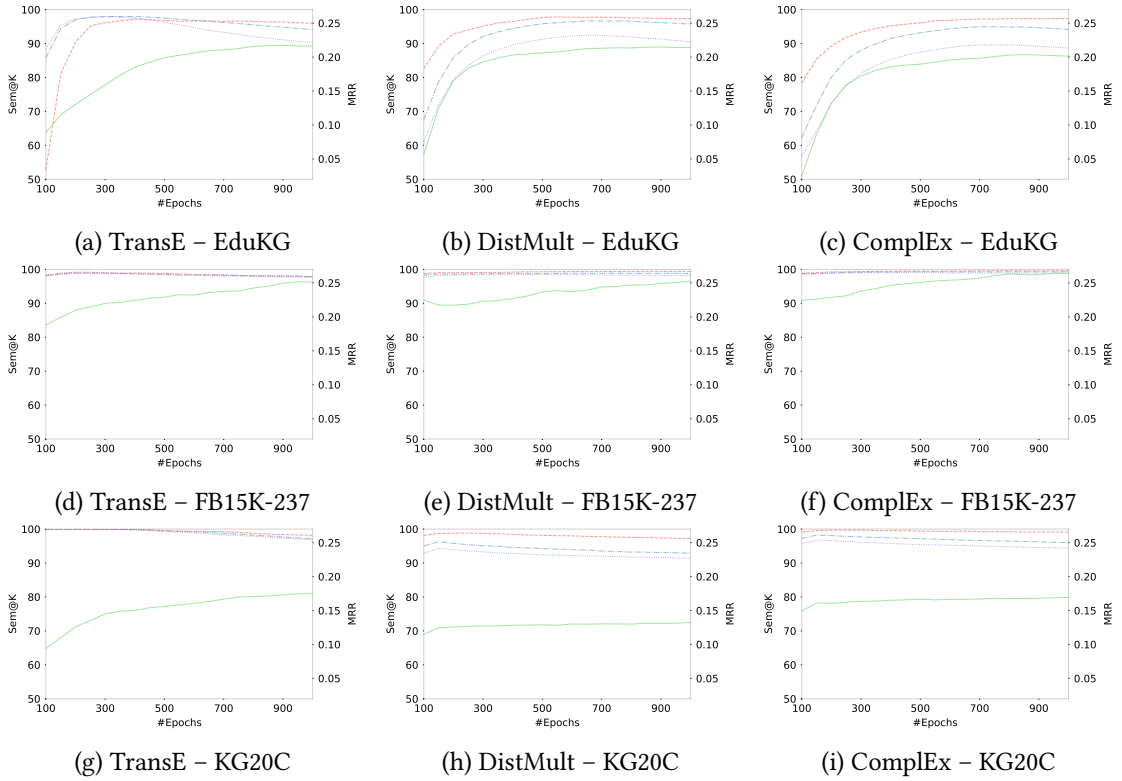
4.2. Experimental Setup

TransE, DistMult, and ComplEx are implemented using PyTorch[2]. All models are trained during 1,000 epochs, with one generated negative triple for each positive triple. The max-margin loss function and Adam optimizer are used, as in [16]. To choose the best hyperparameters[2] for each model and dataset, grid-search was performed on the validation sets, with possible values for the embedding dimension $k \in \{10, 20, 50, 100, 200, 300\}$, learning rate $\eta \in \left[10^{-4}, 10^{-1}\right]$, and margin for the loss function $\gamma \in \{1, 2, 5, 10, 15, 20\}$. In line with [22], we experimentally found that no regularization was needed, as a good choice of γ prevents KGEMs from overfitting. Recall that given a ground-truth triple (h, r, t), all possible triples $(?, r, t)$ and $(h, r, ?)$ are generated with all the entities observed in the KG and scored by the KGEMs.

4.3. Results

Figure 1 presents a complete picture of the evolution of MRR, Sem@1, Sem@5, and Sem@10 according to the number of epochs. Table 2 reports the values of these metrics for specific epochs. Only the key findings are discussed below.

[1] purl.org/edukg/doc
[2] The code of our experiments and the best hyperparameters are available at purl.org/dl4kg-2022.



(a) TransE – EduKG (b) DistMult – EduKG (c) ComplEx – EduKG

(d) TransE – FB15K-237 (e) DistMult – FB15K-237 (f) ComplEx – FB15K-237

(g) TransE – KG20C (h) DistMult – KG20C (i) ComplEx – KG20C

Figure 1: Evolution of MRR (—), Sem@1 (- -), Sem@5 (-·-), and Sem@10 (····) on EduKG, FB15K-237, and KG20C using TransE, DistMult, and ComplEx.

Analysis of Sem@K w.r.t the number of epochs. In this paragraph, Sem@K is analyzed according to the number of epochs (Figure 1). Sem@K reaches high values during the first epochs, regardless of the model and dataset at hand. Most notably, using TransE on KG20C (Figure 1g), Sem@K reaches its maximum values after only 100 epochs with a near-perfect ability to infer the semantic profile of the relations, as evidenced in Table 2. Therefore, it seems that agnostic models are quickly able to learn the domain and range of relations.

It is worth mentioning that the decline of Sem@K appears to be less marked with lower K values. To substantiate this claim, let us consider the Sem@K inflexion point – after which Sem@K starts to decrease – and calculate the difference between the best achieved Sem@K and Sem@K at epoch 1,000. For example, the best Sem@K values using TransE on EduKG are 97.2%, 98.0%, and 97.9% for $K = 1$, $K = 5$, and $K = 10$, respectively. Sem@1, Sem@5, and Sem@10 at epoch 1,000 are 96.0% (-1.2 pts), 94.1% (-4.0 pts), and 90.2% (-7.8 pts), respectively. Except for TransE on FB15K-237 and DistMult on KG20C, all the other combinations of model/dataset lead to the same conclusion: Sem@10 systematically decreases more than Sem@5, just as Sem@5 compared to Sem@1. Figure 1a substantiates this claim as this statement is clearly observed.

Table 2
Comparison of MRR, Sem@1, Sem@5, and Sem@10 values (reported as S@1, S@5,and S@10, respectively) between the best epoch in terms of MRR and the best epoch in terms of Sem@5. In both cases, Epoch denotes the epoch of the reported values.

		Best Epoch (MRR)					Best Epoch (Sem@5)				
		Epoch	MRR	S@1	S@5	S@10	Epoch	MRR	S@1	S@5	S@10
EduKG	TransE	900	0.217	96.4	94.8	91.1	350	0.173	96.7	98.0	97.8
	DistMult	900	0.215	97.4	96.1	91.3	650	0.213	97.7	96.6	92.4
	ComplEx	850	0.204	97.3	94.9	89.4	700	0.198	97.2	95.0	89.6
FB15K-237	TransE	950	0.252	98.1	97.9	97.7	250	0.215	99.2	99.0	98.8
	DistMult	1,000	0.253	99.4	98.8	98.2	750	0.245	99.4	98.9	98.3
	ComplEx	950	0.264	99.6	99.2	98.8	500	0.263	99.6	99.3	98.9
KG20C	TransE	1,000	0.175	98.1	97.2	96.8	100	0.094	100.0	99.9	99.9
	DistMult	1,000	0.132	97.2	92.9	91.3	150	0.125	98.7	96.2	94.3
	ComplEx	1,000	0.169	99.1	96.0	94.4	150	0.161	99.6	98.2	96.8

Comparison of Sem@K Across Models. In Figure 1, we also note that considering Sem@K decline, DistMult and ComplEx seem to be more robust than TransE according to the number of epochs. For example, comparing with the previously mentioned Sem@K losses using TransE on EduKG, for DistMult these are all below -2.1 pts for Sem@1, Sem@5, and Sem@10. For ComplEx, these are all below -1.1 pts for Sem@1, Sem@5, and Sem@10. It is worth investigating whether there is a theoretical explanation that semantic matching models are more robust to Sem@K decline than translational models. This requires further experiments with additional models and a theoretical demonstration that these differences are due to the nature of the KGEMs. This question is left for future research. Moreover, it is noteworthy that Sem@K may depend on the number of entities belonging to each class. For example, KG20C has more entities and less classes than EduKG. Hence, on KG20C, there is a higher probability of predicting semantically valid triples, which seems to be reflected in the higher Sem@K values.

Comparison of MRR and Sem@K. Overall, the best models in terms of MRR are not necessarily the best regarding Sem@K (Table 2). For instance, considering the results achieved on KG20C at epoch 100, TransE has the lowest MRR of the three models: $\text{MRR}_{\text{TransE}} = 0.094$, $\text{MRR}_{\text{DistMult}} = 0.115$ and $\text{MRR}_{\text{ComplEx}} = 0.149$. However, at this very same epoch, TransE showcases near-perfect Sem@K values, compared to the values achieved with DistMult and ComplEx which remain significantly lower (Figure 1h and Figure 1i). This may not be dataset-dependent, as the same remark applies to EduKG and FB15K-237. In addition, we note that regardless of the model and dataset at hand, Sem@K starts to decrease well before MRR in terms of epochs (Figure 1). This means that while Sem@K starts to decline, MRR values continue rising. As such, maximizing Sem@K leads to non-optimal values for MRR, and vice versa. Therefore, a trade-off is to be considered. Some use cases may require ensuring homogeneity in the types of the top-ranked entities. In such scenarios, it may be advisable to retain Sem@K as an additional criterion for tracking the best epoch and performing early-stopping.

5. Conclusion and Future Directions

In this work, we consider the link prediction task and measure the ability of popular KGEMs to predict entities that are semantically valid. Two main findings deserve to be highlighted. We show that agnostic KGEMs can learn the semantic profile of relations. In some cases, however, this comes at the expense of the KGEM performance in terms of traditional rank-based metrics. Thus, there seems to be a trade-off between the semantic awareness of KGEMs and their ability to give higher scores to ground-truth entities. Consequently, our take-home message is that both the training and evaluation of KGEMs would benefit from including Sem@K as an additional criterion alongside commonly used rank-based metrics. Indeed, their combination would give a more complete picture of KGEM performance, as Sem@K provides information on the nature of the errors made by a link prediction model. In future works, we will extend our analysis using more datasets and models, including Graph Neural Networks. We will also explore how type hierarchies and missing or evolving domain and range of relations can be taken into account.

References

[1] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (2017) 2724–2743.

[2] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Transactions on Knowledge Discovery from Data 15 (2021) 14:1–14:49.

[3] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Conf. on Neural Information Processing Systems (NeurIPS), 2013, pp. 2787–2795.

[4] D. Krompaß, S. Baier, V. Tresp, Type-constrained representation learning in knowledge graphs, in: The Semantic Web - 14th International Semantic Web Conf. (ISWC), volume 9366, Springer, 2015, pp. 640–655.

[5] N. Jain, T. Tran, M. H. Gad-Elrab, D. Stepanova, Improving knowledge graph embeddings with ontological reasoning, in: The Semantic Web - International Semantic Web Conf. ISWC, volume 12922, 2021, pp. 410–426.

[6] M. Berrendorf, E. Faerman, L. Vermue, V. Tresp, On the ambiguity of rank-based evaluation of entity alignment or link prediction methods, arXiv preprint arXiv:2002.06914 (2020).

[7] C. T. Hoyt, M. Berrendorf, M. Gaklin, V. Tresp, B. M. Gyori, A unified framework for rank-based evaluation metrics for link prediction in knowledge graphs, arXiv preprint arXiv:2203.07544 (2022).

[8] S. Tiwari, I. Bansal, C. R. Rivero, Revisiting the evaluation protocol of knowledge graph completion methods for link prediction, in: WWW '21: The Web Conf., ACM / IW3C2, 2021, pp. 809–820.

[9] Y. Wang, D. Ruffinelli, R. Gemulla, S. Broscheit, C. Meilicke, On evaluating embedding models for knowledge base completion, in: Proc. of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL, 2019, pp. 104–112.

[10] H. Paulheim, Make embeddings semantic again!, in: Proc. of the ISWC Posters & Demon-

strations, Industry and Blue Sky Ideas Tracks, volume 2180 of *CEUR Workshop Proceedings*, 2018.

[11] N. Jain, J. Kalo, W. Balke, R. Krestel, Do embeddings actually capture knowledge graph semantics?, in: The Semantic Web - 18th International Conf., ESWC, volume 12731 of *LNCS*, Springer, 2021, pp. 143–159.

[12] P. Monnin, C. Raïssi, A. Napoli, A. Coulet, Discovering alignment relations with graph convolutional networks: A biomedical case study, Semantic Web 13 (2022) 379–398.

[13] N. Hubert, P. Monnin, A. Brun, D. Monticolo, New strategies for learning knowledge graph embeddings: The recommendation case, in: EKAW - 23rd International Conf. on Knowledge Engineering and Knowledge Management, Springer, 2022, pp. 66–80.

[14] S. Yang, J. Tian, H. Zhang, J. Yan, H. He, Y. Jin, Transms: Knowledge graph embedding for complex relations by multidirectional semantics, in: Proc. of the Twenty-Eighth International Joint Conf. on Artificial Intelligence, IJCAI, 2019, pp. 1935–1942.

[15] P. Wang, J. Zhou, Y. Liu, X. Zhou, Transet: Knowledge graph embedding with entity types, Electronics 10 (2021) 1407.

[16] B. Kotnis, V. Nastase, Analysis of the impact of negative sampling on link prediction in knowledge graphs, arXiv preprint 1708.06816 (2017).

[17] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: 3rd International Conf. on Learning Representations, ICLR, 2015.

[18] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proc. of the 33rd International Conf. on Machine Learning, ICML, volume 48, 2016, pp. 2071–2080.

[19] G. Chowdhury, M. Srilakshmi, M. Chain, S. Sarkar, Neural factorization for offer recommendation using knowledge graph embeddings, in: Proc. of the SIGIR Workshop on eCommerce, volume 2410, 2019.

[20] M. Nickel, V. Tresp, H. Kriegel, A three-way model for collective learning on multi-relational data, in: Proc. of the 28th International Conf. on Machine Learning, ICML, 2011, pp. 809–816.

[21] R. Kadlec, O. Bajgar, J. Kleindienst, Knowledge base completion: Baselines strike back, in: Proc. of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL, 2017, pp. 69–74.

[22] Z. Sun, Z. Deng, J. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, in: 7th International Conf. on Learning Representations, ICLR, 2019.

[23] N. Hubert, A. Brun, D. Monticolo, New Ontology and Knowledge Graph for University Curriculum Recommendation, in: Proc. of the ISWC Posters & Demo Track, 2022, pp. 1–5.

[24] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proc. of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, Association for Computational Linguistics, 2015, pp. 57–66.

[25] H. N. Tran, A. Takasu, Exploring scholarly data by semantic query on knowledge graph embedding space, in: Digital Libraries for Open Knowledge - 23rd International Conf. on Theory and Practice of Digital Libraries, TPDL, volume 11799, Springer, 2019, pp. 154–162.