

# Sem@K: Is my knowledge graph embedding model semantic-aware?

Nicolas Hubert<sup>a,b,\*</sup>, Pierre Monnin<sup>c</sup>, Armelle Brun<sup>b</sup> and Davy Monticolo<sup>a</sup>

<sup>a</sup> *Université de Lorraine, ERPI, France*

*E-mails: nicolas.hubert@univ-lorraine.fr, davy.monticolo@univ-lorraine.fr*

<sup>b</sup> *Université de Lorraine, CNRS, LORIA, France*

*E-mails: nicolas.hubert@univ-lorraine.fr, armelle.brun@loria.fr*

<sup>c</sup> *Orange, France*

*E-mail: pierre.monnin@orange.com*

**Abstract.** Using knowledge graph embedding models (KGEMs) is a popular approach for predicting links in knowledge graphs (KGs). Traditionally, the performance of KGEMs for link prediction is assessed using rank-based metrics, which evaluate their ability to give high scores to ground-truth entities. However, the literature claims that the KGEM evaluation procedure would benefit from adding supplementary dimensions to assess. That is why, in this paper, we extend our previously introduced metric Sem@K that measures the capability of models to predict valid entities w.r.t. domain and range constrains. In particular, we consider a broad range of KGs and take their respective characteristics into account to propose different versions of Sem@K. We also perform an extensive study of KGEM semantic awareness. Our experiments show that Sem@K provides a new perspective on KGEM quality. Its joint analysis with rank-based metrics offer different conclusions on the predictive power of models. Regarding Sem@K, some KGEMs are inherently better than others, but this semantic superiority is not indicative of their performance w.r.t. rank-based metrics. In this work, we generalize conclusions about the relative performance of KGEMs w.r.t. rank-based and semantic-oriented metrics at the level of families of models. The joint analysis of the aforementioned metrics gives more insight into the peculiarities of each model. This work paves the way for a more comprehensive evaluation of KGEM adequacy for specific downstream tasks.

**Keywords:** Knowledge Graph Embedding, Link Prediction, Model Evaluation, Semantic-Oriented Metrics

## 1. Introduction

A knowledge graph (KG) is commonly seen as a directed multi-relational graph in which two nodes can be linked through potentially several semantic relationships. More formally, a knowledge graph  $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$  where  $\mathcal{E}$ ,  $\mathcal{R}$  and  $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$  are a set of entities (nodes), relations (edge labels) and triples, respectively. A KG is represented as a collection of such triples - a.k.a. facts - denoted as  $(h, r, t) \in \mathcal{T}$  where  $h \in \mathcal{E}$  and  $t \in \mathcal{E}$  are two entities of the graph and are respectively named the head and tail of the triple, while  $r \in \mathcal{R}$  is a predicate that qualifies the nature of the relationship holding between these entities. For instance, in the sample KG depicted in Fig. 1:  $\mathcal{E} = \{\text{BarackObama}, \text{MichelleObama}, \text{EmmanuelMacron}, \text{USA}, \text{France}\}$  and  $\mathcal{R} = \{\text{spouseOf}, \text{presidentOf}, \text{supports}, \text{livesIn}\}$ .

KGs are inherently incomplete, incorrect, or overlapping and thus major refinement tasks include entity matching, question answering, and link prediction [1, 2]. The latter is the focus of this paper. Link prediction (LP) aims at

---

\*Corresponding author. E-mail: nicolas.hubert@univ-lorraine.fr.

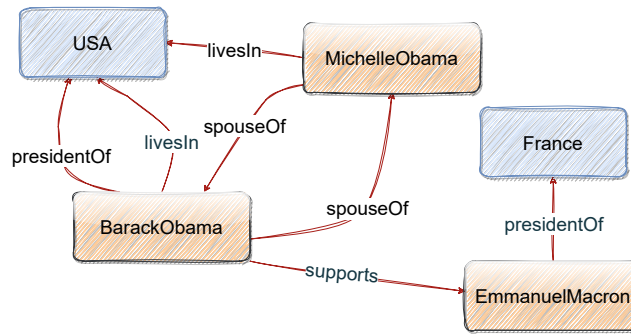


Fig. 1. Excerpt of a KG containing some influential political figures and relations holding between them

completing KGs by leveraging existing facts to infer missing ones. In the LP task, one is provided with a set of incomplete triples, where the missing head (resp. tail) needs to be predicted. This amounts to holding a set of triples  $\mathcal{T}'$  where, for each triple, either the head  $h$  or the tail  $t$  is missing. This task can be subdivided into a head prediction phase - which consists in predicting the most plausible head  $h$  for each  $(?, r, t)$  - and a tail prediction phase - which consists in predicting the most plausible tail  $t$  for each  $(h, r, ?)$ . In the sample KG depicted in Fig. 1, an example of triple to be predicted during the tail prediction phase would be  $(EmmanuelMacron, livesIn, ?)$ , where the expected tail to be inferred is *France*. Training a Knowledge Graph Embedding Model (KGEM) firstly requires corrupting existing triples by replacing either their head  $h$  or their tail  $t$  with another entity to generate negative counterparts. This procedure is called negative sampling [3–5]. Secondly, the KGEM iteratively learns to assign higher scores to true triples than to their negative counterparts.

The performance of KGEM for LP is evaluated using rank-based metrics such as Hits@ $K$ , Mean Rank (MR), and Mean Reciprocal Rank (MRR) that assess whether ground-truth entities are indeed given higher scores [2, 6]. However, various works recently raised some caveats about such metrics [7–9]. For instance, they are not well-suited for drawing comparisons across datasets [7]. More importantly, they only provide a partial picture of KGEM performance [7]. 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, 10]. KGEMs with such issues may nevertheless reach a satisfying performance in terms of rank-based 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, *e.g.*, domain and range constraints, hierarchy of classes [11–13]. According to Berrendorf *et al.* [7], this would give a more complete picture of the performance of a KGEM. This is why we advocate for additional qualitative and semantic-oriented metrics to supplement traditional rank-based metrics and propose Sem@ $K$  to address this need. The relevance of using semantic-oriented metrics – more specifically Sem@ $K$  – is clearly visible in Fig. 2: Sem@ $K$  provides a supplementary dimension to the evaluation procedure and allows to confidently choose between two models. They are equally good in terms of rank-based metrics but Model A predicts entities that are semantically valid w.r.t. the range constraint.

More specifically, 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 LP task. To do so, we build on Sem@ $K$ , the semantic-oriented metric that we introduced in [14] for a recommendation task. This metric was originally used to evaluate the ability of KGEMs to recommend items that are of the expected type. We extended Sem@ $K$  analysis in [15], where the metric was used in the more generic LP framework. In the present work, we deepen the study of the semantic awareness of KGEMs by proposing different versions of Sem@ $K$  that take into account the different characteristics of KGs (*e.g.*, hierarchy of types). Moreover, the semantic awareness of a wider range of KGEMs is analyzed – especially convolutional models. Likewise, a broader array of KGs is used, in order to benchmark the semantic capabilities of KGEMs on mainstream LP datasets. Thus, the following research questions are addressed:

- RQ1: how semantic-aware agnostic KGEMs are?
- RQ2: how the evaluation of KGEM semantic awareness should adapt to the typology of KGs?
- RQ3: does the evaluation of KGEM semantic awareness offer different conclusions on the relative superiority of some KGEMs compared to rank-based metrics?

Accordingly, the main contributions of this work are:

- to evaluate KGEM semantic awareness on any kind of KG, we extend a previously defined semantic-oriented metric and tailor it to support a broader range of KGs.
- we perform an extensive study of the semantic awareness of state-of-the-art KGEMs on mainstream KGs. We show that most of the observed trends apply at the level of families of models.
- we perform a dynamic study of the evolution of KGEM semantic awareness vs. their performance in terms of rank-based metrics along training epochs. We show that a trade-off may exist for most KGEMs.
- our study supports the view that agnostic KGEMs are quickly able to infer the semantics of KG entities and relations.

The remainder of the paper is structured as follows. Related work is presented in Section 2. Section 3 outlines the main motivations for assessing the semantic awareness of KGEMs. Section 4 subsequently presents Sem@K, the semantic-oriented metric that fulfills this purpose. Sem@K comes in different flavors based on the typology of the datasets at hand and the intended use cases. In Section 5, we detail the datasets and KGEMs used in this work, before presenting the experimental findings in Section 6. A thorough discussion is provided in Section 7. Lastly, Section 8 outlines future research directions.

## 2. Related work

### 2.1. Link prediction using knowledge graph embedding models

Several LP approaches have been proposed to complete KGs. Symbolic approaches relying on rule-based [16–20] or path-based reasoning [21–23] are somewhat popular but are not considered in the present work. Instead, KGEMs are the focus of this paper.

In particular, this work is concerned with assessing the semantic awareness of agnostic KGEMs. The semantic awareness of such models is defined as their ability to score higher entities whose types belong to the domain and range of relations. Agnostic KGEMs are defined as KGEMs that rely solely on the structure of the KG to learn entity and relation representations. In this respect, these models differ according to several criteria, such as the nature of the embedding space or the type of scoring function [2, 6, 24, 25]. In this work, KGEMs are considered with respect to three main families of models that are traditionally distinguished in the literature.

**Geometric models** are additive models that consider relations as geometric operations in the latent embedding space. A head entity  $h$  is spatially transformed using an operation  $\tau$  that depends on the relation embedding  $r$ . Then the distance between the resulting vector and the tail entity  $t$  is used as a measure for assessing the plausibility of a fact  $(h, r, t)$ . A distance-based function  $\delta$  is used to define the scoring function  $f$  of such KGEMs:  $f(h, r, t) = \delta(\tau(h, r), t)$ . A large array of geometric models is purely translational and is based on TransE [3] and its extensions. TransE was the earliest introduced geometric model for link prediction. It enforces the sum of the head and relation embeddings to lie in a close neighborhood of the tail embedding. The distance-based function  $\delta$  is usually the  $L1$  or  $L2$  norm. TransE does not properly handle 1-to-N, N-to-1, nor N-to-N relations [2] and yet has been found to be very efficient in multi-relational settings [26]. A myriad of translational models have been proposed since then, such as TransH [27], TransR [28], TransD [29], TransA [30] and TransG [31]. In addition to these purely translational models, some recent geometric models either replace or combine the translation operation with rotation-wise transformations to make their models even more expressive and deal with difficult relational patterns such as symmetric or anti-symmetric relations. A case in point is RotatE [32], which considers relations as rotations in a complex latent space.  $h$ ,  $r$  and  $t$  are all embedded in  $\mathbb{C}^d$  where  $d$  denotes the embeddings' dimension. The use of the rotation operation allows RotatE to properly address many relational patterns. In particular, RotatE is able to take account for symmetry – which is not modeled correctly by TransE. In the wake of RotatE, a handful of other roto-translational models have subsequently been proposed in the literature, *e.g.* QuatE [33], DualE [34] and HAKE [35].

**Semantic matching models** are named this way as they usually use a similarity-based function to define their own scoring function. Semantic matching models are also referred to as matrix factorization models or tensor decomposition models, in the sense that a KG can be seen as a 3D adjacency matrix, in other words a three-way tensor.

This tensor can be decomposed into a set of low-dimensional vectors that actually represent the entity and relation embeddings. Semantic matching models are subdivided into bilinear and non-bilinear models. Bilinear models share the characteristic of capturing interactions between two entity vectors using multiplicative terms. RESCAL [36] is the earliest introduced bilinear model for LP. It models entities as vectors and relations as matrices. The components  $w_{i,j}$  of the relation matrix  $W_r \in \mathbb{R}^{d \times d}$  account for the interaction intensity between the  $i$ -th embedding component of  $e_h \in \mathbb{R}^d$  and the  $j$ -th embedding component of  $e_t \in \mathbb{R}^d$  under the relation  $r$ . As pointed out by the original author of RESCAL [37], the parameter complexity of this model can drastically increase when the embedding dimension is large. DistMult [38] alleviates this scalability issue by enforcing all relation-matrices  $W_r$  to be diagonal. Doing so, DistMult trades computational advantages for less expressiveness. Indeed, DistMult is not able to model anti-symmetric relations. However, DistMult still achieves state-of-the-art performance in most cases [39]. ComplEx [40] relies on complex-valued representations for both entities and relations. Because the Hadamard product is used in the scoring function, ComplEx is not commutative in the complex space and can properly model anti-symmetric relations. Other bilinear models with distinctive features were later proposed, such as Analogy [41], Simple [42] and Dihedral [43] that are supposedly more expressive models. Another category of semantic matching models leverage interactions between entities and relations using non-bilinear operations. For instance, HolE [44] uses the circular correlation operation, while TuckER [45] relies on the Tucker decomposition.

**Neural network-based models** rely on neural networks to perform LP. In neural networks, the parameters (*e.g.* weights and biases) are organized into different layers, with usually non-linear activation functions between each of these layers. The first introduced neural-network based model for LP is Neural Tensor Network (NTN) [46]. It can be seen as a combination of multi-layer perceptrons (MLPs) and bilinear models [25]. NTN defines a distinct neural network for each relation. This choice of parameterization makes NTN similar to RESCAL in the sense that both models achieve great expressiveness at the expense of computational concerns. The most recent neural network-based models rely on more sophisticated layers to perform a broader set of operations. Convolutional models are by far the most representative family of such models. They use convolutional layers to learn deep and expressive features from the input data, which pass through such layers to undergo convolution with low-dimensional filters  $\omega$ . The resulting feature maps subsequently go through dense layers to obtain a final plausibility score. Compared with fully connected neural networks, convolution-based models are able to capture complex relationships with fewer parameters by learning non-linear features. While ConvE [47] reshapes head entity and relation embeddings before concatenating them into a unique input matrix to pass through convolutional layers, ConvKB [48] does not perform any reshaping and also puts the tail embedding into the concatenated input matrix. Other models were later proposed, such as ConvR [49] and InteractE [50] that both process triples independently. Another branch of convolutional models also considers the local neighborhood around each central entity. These are based on Graph Neural Networks (GNNs). The most representative GNN-based model for LP is R-GCN [51]. The key idea is to accumulate messages from the local neighborhood of the central node over multi-hop relations. By doing so, R-GCN is better able to model a long range of relational dependencies. However, R-GCN does not outperform baselines for the LP task [52]. Subsequent models have claimed superiority over R-GCN: SACN [53] introduces a weighted GCN to adjust the amount of aggregated information from the local neighborhood and KBGAT [54] relies on attention mechanism to generate more accurate embeddings. More recently, CompGCN [55] and DisenKGAT [56] showcased impressive performance with regard to the LP task.

## 2.2. Combining embeddings and semantics

The possibility of using additional semantic information has been extensively studied in recent works [4, 5, 57–61]. In general, the semantic information stems directly from an ontology, originally defined by Gruber as an "explicit specification of a conceptualization" [62]. Ontologies formally describe a specific application domain of interest (*e.g.* education, pharmacology, etc.) in which several classes (or concepts) and relations are identified and formally specified. Ontologies support KG construction by providing a schema that specifies the nature of entities, the semantic profile of relations, and other constraints that give the KG a semantic coherence.

A significant part of the literature incorporates such semantic information to constrain the negative sampling procedure and generate meaningful negative triples [4, 5]. 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 (`rdf: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. In fact, some KGEMs leverage ontological information such as entity types and hierarchy. For instance, AutoETER [57] is an automated type representation learning mechanism that can be used with any KGEM and that learns the latent type embedding of each entity. In TaRP [58], type information and instance-level information are simultaneously considered and encoded as prior probabilities and likelihoods of relations, respectively. TKRL [59] bridges type information with hierarchical information: while type information is utilized as relation-specific constraints, hierarchical types are encoded as projection matrices for entities. TKRL allows entities to have different representations in different types. Similarly, in [61], the proposed KGEM allows entities to have different vector representations depending on their respective types. TransC [60] encodes each concept of a KG as a sphere and each instance as a vector in the same semantic space. The relations between concepts and instances (`rdf:type`), and the relations between concepts and sub-concepts (`rdfs:subClassOf`) are based on the relative distance within this shared semantic space.

Embedding models project entities and relations of a KG into a vector space. Thus, the semantics of the original KG may not be fully preserved [5, 11]. As stated by Paulheim [11], 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 [11] advocates for *semantic embeddings*. Similarly, Jain *et al.* [12] perform a thorough evaluation of popular KGEMs to better assess whether embeddings can express similarities between entities of the same type. A key finding is that because of overlapping relations among entities of different types, fine-grained semantics cannot be properly reflected by embeddings. For instance, the task of finding semantically similar entities does not always provide satisfying results when working with entity embeddings [12].

### 2.3. Evaluating KGEM performance for link prediction

KGEM performance is evaluated in two stages. During the validation phase, KGEM performance is evaluated on the validation set  $\mathcal{T}_{valid}$  after regular – often uniform – intervals of epochs. This way, the best epoch is identified. During the test phase, KGEM performance is ultimately evaluated on the test set  $\mathcal{T}_{test}$  after retrieving the optimal model parameters achieved on the best epoch of validation. Whether during the validation or test phase, KGEM performance is evaluated the same way: sifting through every triple of the test (resp. validation) set, both head and tail predictions are performed. In the case of head prediction, this amounts to taking a triple  $(h, r, t)$  from the test (resp. validation) set, hiding the ground-truth head entity – resulting in  $(?, r, t)$  – and letting the KGEM assign a score to every possible entity as a candidate for the head position. These scores are finally ordered and reflect the plausibility of such facts. Tail prediction is performed analogously on  $(h, r, ?)$ .

In both cases, the rank of the ground-truth entity from the test (resp. validation) set is used to compute aggregated rank-based metrics based on the top- $K$  scored entities. The rank of the ground-truth entity can be determined in two different ways that depend on how observed facts – *i.e.* facts that already exist in the KG – are considered. In the raw setting, observed facts outranking the ground-truth are not filtered out, while this is the case in the filtered setting. For instance, assuming head prediction is performed on the given the ground-truth triple (`BarackObama, livesIn, USA`), a KGEM may assign a lower score to this triple than to the following triple: (`MichelleObama, livesIn, USA`). The latter triple actually represents an observed fact. In the raw setting, this triple would not be filtered out from top- $K$  scored triples. This can cause the evaluation procedure to not properly assess the KGEM performance. This is why in practice, the filtered setting is commonly preferred. In the present work, the filtered setting is also used.

KGEM performance is almost exclusively assessed using the following rank-based metrics: Hits@ $K$ , Mean Rank (MR), and Mean Reciprocal Rank (MRR) [8]. Disagreements exist as to how and when these metrics can be used and compared properly. In the following, we recall their definitions and discuss their limits.

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

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

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  $\mathbb{1}[\text{rank}(q) \leq K]$  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)** (Eq. (2)) corresponds to the arithmetic mean over ranks of ground-truth triples:

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

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)** (Eq. (3)) corresponds to the arithmetic mean over reciprocals of ranks of ground-truth triples:

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

Contrary to MR, MRR is a metric bounded in the  $[0, 1]$  interval. Higher results indicate better performance. 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 [7, 8].

As mentioned in Section 1, these metrics present some caveats. LP is often used to complete knowledge graphs, 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 [10].

In addition, the aforementioned rank-based metrics have intrinsic and theoretical flaws, as pointed out in several works [7–9]. 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 [8]. 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 Eq. (2)) [7]: 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.

Recent works recommend using adjusted version of the aforementioned metrics. The Adjusted Mean Rank (AMR) proposed in [63] compares the mean rank against the expected mean rank under a model with random scores. In [7], Berrendorf *et al.* transform the AMR to define the Adjusted Mean Rank Index (AMRI) bounded in the  $[0, 1]$  interval. This way, a value of 1 indicates an optimal performance of the model. A value of 0 indicates a model performance similar to a model assigning random scores. A negative value indicates that the model performs worse than random. However, all these attempts at producing a better evaluation framework still focus on the quantitative assessment of KGEMs, *i.e.* the improvement of already existing rank-based metrics.

In Section 2.2, several approaches incorporating the semantics of entities and relations into the embeddings were mentioned. However, in such cases, the use of semantic information such as entity types and the hierarchy of classes is intended to improve KGEM performance in terms of the aforementioned rank-based metrics only. The underlying semantics of KGs is considered as an additional source of information during training but the ability of KGEMs to generate predictions in accordance with these semantic constrains is never directly addressed. This encourages further assessment of the semantic capabilities of KGEMs, as firstly suggested in [15]. 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. When such information is not available – *i.e.* the KG does not rely on a schema containing `rdfs:domain` and `rdfs:range` properties – extensional constrains can still be used to evaluate the semantic capabilities of KGEMs, as detailed in Section 4.

### 3. Motivations and problem formulation

#### 3.1. Motivating example

To motivate the use of Sem@K to evaluate KGEM quality, this section builds upon a minimalist example which is representative of the issue encountered while benchmarking the performance of several KGEM on the same dataset. As depicted in Fig. 2, two KGEMs that have been trained on the whole training set are tested on a batch of test triples. These KGEMs are referred to as Model A and Model B. Without loss of generality, it is assumed the test set only comprises the three triples shown in Fig. 2. For the sake of clarity, only the tail prediction pass and the top-5 ranked candidate entities are depicted in Fig. 2. It should be noted that the performance of both models are strictly equal in terms of MR, MRR and Hits@K with  $K \leq 5$ : MR=8/3, Hits@1=1/3, Hits@3=2/3 and Hits@5=1. MRR can only be computed knowing the total number of entities in the KG but two models having the same MR on the same dataset have *de facto* the same MRR as well. Distinguishing between these two models by relying solely on traditional rank-based metrics is not possible. One might even draw the misleading conclusion that these models are equally good. But they are not: Model A gives high scores to semantically valid entities with regard to the range of relations, while Model B semantic awareness is very low. In other words, Model B does not capture the semantic profile of relations well. This case in point illustrates the need for using semantic-oriented metrics alongside common rank-based metrics, so as to better assess the overall quality of KGEMs.

#### 3.2. Problem formulation

The traditional evaluation of KGEMs solely based on rank-based metrics can be flawed for several reasons. First, KGEMs benchmarked on the same test sets can exhibit very similar results. Using only rank-based metrics, the final choice only depends on the best achieved MRR and/or Hits@K. This raises the questions whether the chosen model is actually the best one, or whether its slight superiority over other KGEMs can be due to other factors such as better hyperparameter tuning or better modeling of a relational pattern highly present in the test set. Moreover, using only rank-based metrics does not provide the full picture of KGEM quality for the downstream LP task, as some dimensions of KGEMs are left unassessed (see Section 2). In this work, contrary to the mainstream approach consisting in comparing KGEM performance exclusively in terms of rank-based metrics, the trained KGEMs are also evaluated in terms of Sem@K which measures the ability of KGEMs to predict semantically valid triples with respect to the domain and range of relations. In particular, the following research questions are addressed:

- RQ1: how semantic-aware agnostic KGEMs are?
- RQ2: how the evaluation of KGEM semantic awareness should adapt to the typology of KGs?
- RQ3: does the evaluation of KGEM semantic awareness offer different conclusions on the relative superiority of some KGEMs compared to rank-based metrics?

### 4. Measuring KGEMs semantic awareness with Sem@K

The standard LP evaluation protocol consists in reporting aggregated results, considering the rank-based metrics presented in Section 2.3. As aforementioned, these metrics only provide a partial picture of KGEM performance [7]. To give a more comprehensive assessment of KGEMs, we aim at assessing their semantic awareness using our proposed metric called Sem@K [14, 15]. In [14], Sem@K was specifically defined for the recommendation task which was seen as predicting tails for a unique target relation. Sem@K was then extended in [15] to the more generic LP task, where not only tails but also heads are corrupted and all relations are considered. In this work, this original

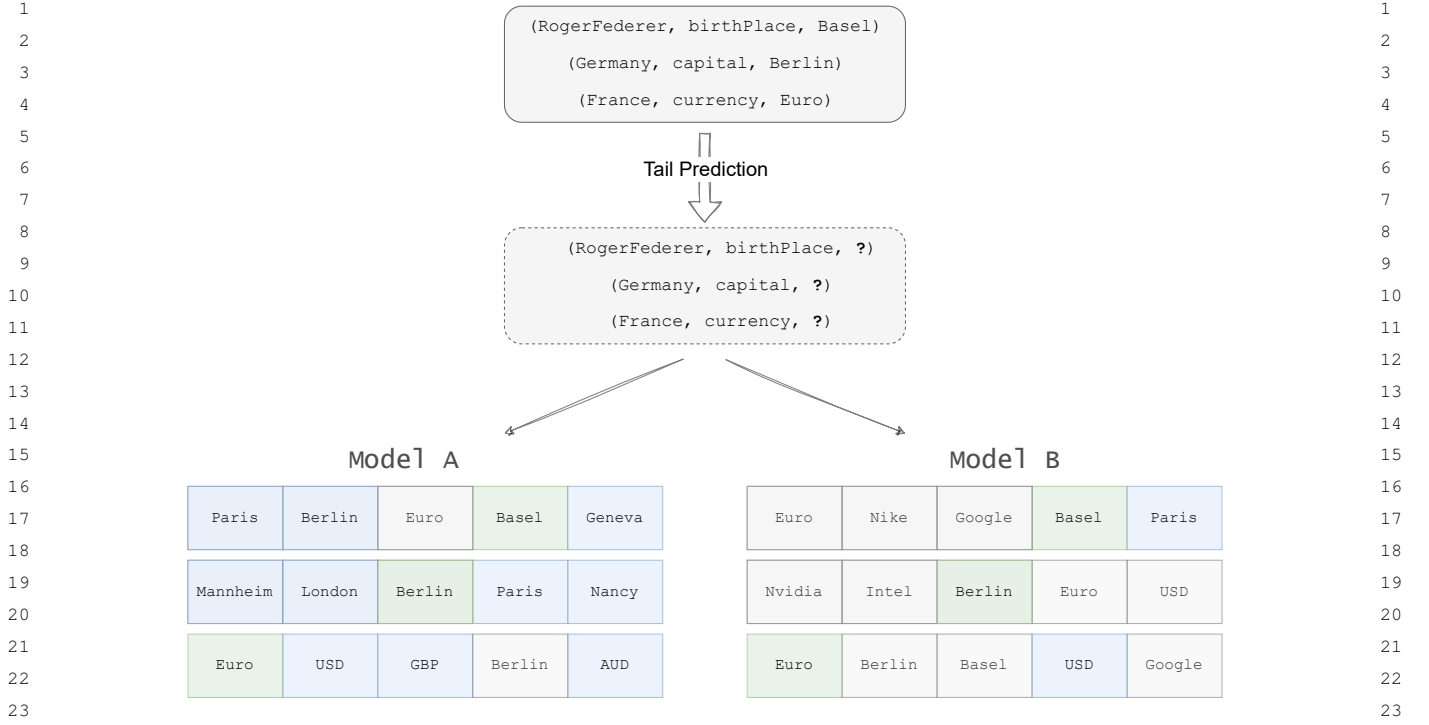


Fig. 2. Motivating example. Tail prediction is performed for the three test triples contained in the upper insert. Model A and Model B output scores for each possible entity and only the top-five ranked tail candidates are depicted here. Model A and Model B have the same Hits@1, Hits@3 and Hits@5 values. But Model A has better semantic capabilities. Green, blue and white cells respectively denote the ground-truth entity, entities other than the ground-truth and semantically valid, and entities other than the ground-truth and semantically invalid.

Table 1  
Typology of KGs and their respective adequacy for the presented Sem@K versions.

KG type	Sem@K[ext]	Sem@K[base]	Sem@K[wup]
Schemaless	×		
Schema-defined, w/o class hierarchy	×	×	
Schema-defined, w/ class hierarchy	×	×	×

formalization for LP is presented in Section 4.1, and enriched to take into account schemaless KGs (Section 4.2), or KGs with a class hierarchy (Section 4.3). As a consequence, Sem@K comes in 3 different versions (respectively denoted Sem@K[base], Sem@K[ext], and Sem@K[wup]) so as to adapt to KG typology. These distinct versions and their adequacy regarding the KG at hand are summarized in Table 1 and further detailed below. In the following, when no suffix is provided, it is assumed that we are concerned with Sem@K in general, regardless of the actual version.

#### 4.1. Sem@K[base] for schema-defined KGs

##### 4.1.1. Definition of Sem@K[base]

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



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

where, given a ground-truth triple  $q = (h, r, t)$ ,  $\mathcal{S}_q^K$  is the list of 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 function  $\text{compatibility}(q, q')$  (Eq. (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:

$$\text{compatibility}(q, q') = \begin{cases} 1, & \text{if } \text{types}(q'_h) \cap \text{domain}(q_r) \neq \emptyset \wedge \text{types}(q'_t) \cap \text{range}(q_r) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where  $\text{types}(e)$  returns the types of entity  $e$  and  $\text{domain}(r)$  (*resp.*  $\text{range}(r)$ ) 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. It is noteworthy that with this formula, we allow entities to instantiate multiple types, and domains and ranges to be defined with multiple types.  $\text{Sem@}K$  is bounded in the  $[0, 1]$  interval. Compared to  $\text{Hits@}K$  (Eq. (1)),  $\text{Sem@}K$  is non-monotonic: increasing  $K$  can lead to either lower or higher  $\text{Sem@}K$  values.

#### 4.1.2. A note on Sem@K, untyped entities, and the Open World Assumption

Traditional KGEM evaluation can be performed in all situations, regardless of whether entities are typed or whether the KG comes with a proper schema. When measuring the semantic capability of KGEMs – *e.g.* with  $\text{Sem@}K$  – some concerns arise. For instance, a fair question to ask is the following: how should untyped entities be considered? Indeed, in some KGs, some entities are left untyped. For example, in DBpedia the entity `dbr:1._FC_Union_Berlin_players` does not belong to any other class than `owl:Thing`. In some other cases as in DB15K and DB100K [64], entities have incomplete typing.

Under the OWA, an untyped entity should not count in the calculation of  $\text{Sem@}K$  since it is not possible to determine whether this entity has no types or no known types. Although this seems to be a fair option, it raises a major issue: it makes possible to score different sets of entities with rank-based metrics and  $\text{Sem@}K$ , which is not desirable. If there are  $M$  untyped entities in an ordered list of ranked entities,  $\text{Hits@}K$ , MR and MRR are calculated regardless of this fact, *i.e.* still taking into account the  $M$  untyped entities. However,  $\text{Sem@}K$  cannot be calculated for these  $M$  untyped entities. As such, MR, MRR and  $\text{Hits@}K$  would be calculated on the original entity set, whereas  $\text{Sem@}K$  would be computed on a different set of entities. This issue would be even more acute in the case of  $\text{Sem@}1$  when the first ranked entity is untyped:  $\text{Hits@}1$  and  $\text{Sem@}1$  would be calculated on two different entities, which is not acceptable. Consequently, one strategy consists in removing untyped entities from the evaluation protocol, both regarding rank-based and semantic-based metrics. By doing so, consistency is ensured in the ranked list of entities across rank-based and semantic metrics evaluation.

#### 4.2. Sem@K[ext] for schemaless KGs

Not all KGs come with a proper schema, *e.g.* relations do not appear in any `rdfs:domain` or `rdfs:range` clauses. In that particular situation, it can still be desirable to assess the semantic awareness of KGEMs. One approach is to maintain a list of all entities that have been observed as heads (*resp.* tails) of each relation  $r$ :  $\text{domain}(r) = \{e : \exists(e, r, t) \in \mathcal{T}\}$  (*resp.*  $\text{range}(r) = \{e : \exists(h, r, e) \in \mathcal{T}\}$ ). Hence, in this case,  $\text{Sem@}K[\text{ext}]$  is defined as in Eq. (4) and (5) but using these definitions of domain and range. Therefore, an entity will be considered as a semantically valid head (*resp.* tail) with respect to the relation if this entity appears as a head (*resp.* tail) in any other triple observed in the KG with the same relation.

### 4.3. Sem@K[wup]: a hierarchy-aware version of Sem@K

Sem@K as previously defined equally penalizes all entities that are not of the expected type. However, KGs may be equipped with a class hierarchy that, in turn, can support a more fine-grained penalty for entities depending on the distance or similarity between their type and the expected domain (resp. range) in this hierarchy. To illustrate, consider Figure 3 that depicts a subset of the DBpedia ontology dbo class hierarchy. Using the hierarchy-free version of Sem@K for the test triple (dbr:The\_Social\_Network, dbo:director, dbr:David\_Fincher), predicting dbr:Friends or dbr:Central\_Park as head would be penalized the same way in the compatibility function. However, it is clear that an entity of class dbo:TelevisionShow is semantically closer to dbo:Film – the domain of the relation dbo:director – than an entity of class dbo:Park, and thus should be less penalized.

To leverage such a semantic relatedness between concepts in Sem@K, the compatibility function can be adapted accordingly:

$$\text{compatibility}(q, q') = \min \left( \max_{\substack{c \in \text{type}(q_h) \\ c' \in \text{domain}(q_r)}} \sigma(c, c'), \max_{\substack{c \in \text{type}(q'_h) \\ c' \in \text{range}(q'_r)}} \sigma(c, c') \right) \quad (6)$$

where  $\sigma(c, c')$  measure the semantic similarity between the two classes  $c$  and  $c'$  based on the class hierarchy. It should be noted that in this formula  $\text{type}(e)$  only returns the most specific classes instantiated by  $e$ .

Several similarity measures  $\sigma$  have been proposed in the literature [65–69]. In this work, the Wu-Palmer similarity [66] (Eq. 7) is used:

$$\sigma(c, c') = \frac{2 \times \delta(c \wedge c', \rho)}{\delta(c, c \wedge c') + \delta(c', c \wedge c') + 2 \times \delta(c \wedge c', \rho)} \quad (7)$$

where  $\rho$  is the root of the hierarchy (e.g. owl:Thing),  $\delta(c, c')$  is the number of edges linking  $c$  to  $c'$ , and  $c \wedge c'$  represents the least common subsumer of  $c$  and  $c'$ . The Wu-Palmer similarity is well suited to a class hierarchy and provides a good indication of the semantic relatedness between the domain (resp. range) class and the classes of the chosen entity. This gives rise to the Sem@K[wup] version.

Considering the example in Fig. 3, using the Wu-Palmer score in the calculation of Sem@K, a head prediction of dbr:Friends and dbr:Central\_Park for the ground-truth triple (dbr:The\_Social\_Network, dbo:director, dbr:David\_Fincher) are now differently penalized. The instance dbr:Friends is of type dbo:TelevisionShow, so we have:  $\sigma(\text{dbo:TelevisionShow}, \text{dbo:Film}) = 1/2$ . The instance dbr:Central\_Park is of type dbo:Park, so we have:  $\sigma(\text{dbo:Park}, \text{dbo:Film}) = 0$ . This illustrates that incorporating the Wu-Palmer score into Sem@K calculation leads to more precise semantic comparisons that take into account the available class hierarchy. It should be noted that comparing the classes of a candidate entity with the expected class can result in the same penalization as the vanilla version of Sem@K. However, in most cases, two classes do not lie in a completely disjoint part of the hierarchy of classes. As such, the Wu-Palmer score between the classes of a candidate entity and the expected class is rarely 0.

## 5. Experimental setting

### 5.1. Datasets

In order to draw reliable and general conclusions, a broad range of KGs are used in our experiments. They have been chosen due to their mainstream adoption in recent research works around KGEMs for LP and the fact that they have different characteristics (e.g. entities, relations, classes, presence of a class hierarchy). In this section, the schema-defined and schemaless KGs used in the experiments are detailed. Note that in our experiments, all the schema-defined KGs come with a class hierarchy inherited from either Freebase [70], DBpedia [71], YAGO [72],

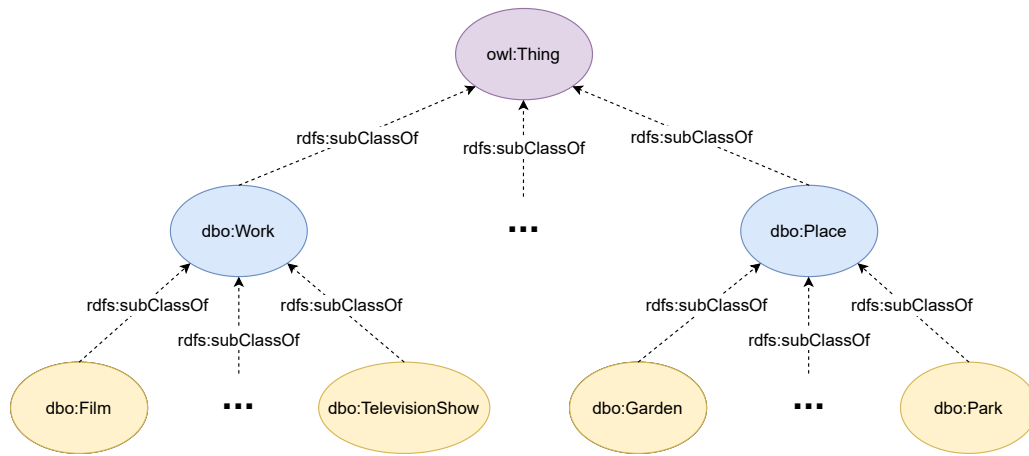


Fig. 3. Excerpt from the DBpedia class hierarchy

or `schema.org`. To meet requirements for evaluating KGEMs w.r.t.  $\text{Sem}@K$  (*i.e.* classes instantiated by entities, domain and range for relations), when necessary, subsets of schema-defined KGs are used, as explained in Section 5.1.1. Among the 4 schema-defined KGs presented hereafter, FB15K237-ET, DB93K and YAGO3-37K are derived from already existing KGs, while YAGO4-19K was specifically created and is made available on Zenodo<sup>1</sup> and GitHub<sup>2</sup>. The other KGs are made available on GitHub<sup>3</sup>.

#### 5.1.1. Schema-defined KGs

The statistics of the datasets FB15K237-ET, DB93K, YAGO3-37K and YAGO4-19K used in our experiments are provided in Table 2. Accordingly to Section 4.1.2, to create an experimental evaluation setting as unbiased and flawless as possible, the schema-defined KGs used in the experiments are filtered to keep typed entities only. This way,  $\text{Sem}@K$  is calculated under the CWA.

**FB15K237-ET** derives from FB15K [3] – a dataset extracted from the cross-domain Freebase KG [70]. In these experiments, we do not use FB15K, as it has been noted that this dataset suffers from test leakage, *i.e.*, a large number of test triples can be obtained by simply inverting the position of the head and tail in the train triples [73]. The later introduced FB15K237 dataset [73] is a subset of the original FB15K without these inverse relations. To the best of our knowledge, there is no schema-defined version of FB15K237. However, a schema-defined version of FB15K is provided in [59]. Consequently, we based ourselves on this version of FB15K and mapped the extracted entity types, relation domains and ranges to the entities and relations found in FB15K237. The resulting schema-defined version of FB15K237 is named FB15K237-ET and includes only typed entities. Besides, validation and test sets contain triples whose relation have a well-defined domain (*resp.* range), as well as more than 10 possible candidates as head (*resp.* tail). This ensures  $\text{Sem}@K$  is not unduly penalized and can be calculated on the same set of entities as Hits@K and MRR, at least until  $K = 10$ .

**DB93K** is a subset of DB100K, which was first introduced in [74]. A slightly modified version of DB100K has been proposed in [64]. Contrary to the initial version of DB100K, the latter version is schema-defined: entities are properly typed and most relations have a domain and/or a range. This second version is considered in the following experiments. However, some inconsistencies were found in the dataset. Some DBpedia entities only instantiate Wikidata<sup>4</sup> or `schema.org`<sup>5</sup> classes, while instantiation of classes from the DBpedia ontology actually exist. Moreover, some entities are only partially typed. It must also be noted that domains and ranges of relations have been extracted from DBpedia more than two years ago. DBpedia is a communaury and open-source project: DBpedia

<sup>1</sup><https://doi.org/10.5281/zenodo.7526244>

<sup>2</sup><https://github.com/pmonnin/YAGO4-LP>

<sup>3</sup><https://github.com/nicolas-hbt/benchmark-sematk>

<sup>4</sup><https://www.wikidata.org/>

<sup>5</sup><https://schema.org/>

Table 2  
 Statistics of the schema-defined, hierarchical KGs used in the experiments

Dataset	$ \mathcal{E} $	$ \mathcal{R} $	$ \mathcal{C} $	$ \mathcal{T}_{train} $	$ \mathcal{T}_{valid} $	$ \mathcal{T}_{test} $
FB15K237-ET	14,541	237	532	271,575	15,337	17,928
DB93K	92,574	277	311	237,062	18,059	36,424
YAGO3-37K	37,335	33	132	351,599	4,220	4,016
YAGO4-19K	18,960	74	1,232	27,447	485	463

classification system relies on human curation, which sometimes implies a lack of coverage for some resources and updates for others. Consequently we associated all relations of DB100K to their most up-to-date domains and ranges<sup>6</sup>. Similarly for entities, we associated all entities to their most up-to-date classes. Finally, we removed all untyped entities as well as validation and test relations having less than 10 observed entities, so as not to unfairly penalize Sem@K results in the validation and test phases.

**YAGO3-37K** derives from the schema-defined YAGO39K dataset [60] extracted from the cross-domain YAGO3 KG [72]. Compared to the original YAGO39K, in our experiments only typed entities are kept. In addition, relations having less than 10 observed heads or tails in the training set are discarded from the validation and test splits, for the same reason that keeping them would not reflect the actual Sem@K values. It should be noted that in the YAGO3 ontology, most relations have very generic domains and ranges which are very close to the root of the ontology hierarchy. To produce a more challenging evaluation setting of the models’ semantic awareness, a subset of hard relations was identified and only validation and test triples whose relation belongs to this subset are kept. The resulting dataset is named YAGO3-37K.

We built **YAGO4-19K** with several other subsets of the YAGO4 knowledge graph [75]. Similarly to other datasets, we focused on relations with a defined domain and range, and more than 10 triples to constitute the validation and test sets. We purposely favored difficult relations to feature in the validation and test sets. To enrich the training set, additional relations were added based on a manual selection. Selected relations in validation and test sets as well as additional relations in the training set can be found on the GitHub repository of the dataset. It should be noted that the class hierarchy associated with entities in YAGO 4 is `schema.org`.

### 5.1.2. Extensional KGs

Another range of datasets used in these experiments do not come with an ontological schema. In particular, this means relations do not have a clearly-defined domain or range. Although Codex-S and Codex-M are based on the Wikidata schema which does possess property constraints linking subject types to value type constraints, we limit ourselves to KGs that represent this information with `rdfs:domain` and `rdfs:range` predicates. Consequently, in this work, we only report Sem@K[ext] results for Codex-S and Codex-M. The statistics of the datasets Codex-S, Codex-M and WN18RR used in our experiments are provided in Table 3.

**WN18RR** [47] originates from WN18, which is a subset of the WordNet KG [76]. As for FB15K, Toutanova *et al.* [73] reported a huge test leakage in the original WN18 dataset. More specifically, 94% of the train triples have inverse relations that are linked to test triples. Dettmers *et al.* [47] remove all inverse relations to propose WN18RR. In WN18RR, entities are nouns, verbs, and adjectives. Relations such as `hypernym` and `derivationally related from hold` between observed entities. Such relations are not linked to any `rdfs:domain` or `rdfs:range` predicates. Besides, relations such as `derivationally related from` can contain nouns, verbs or adjectives as both head or tail. It follows that it is not possible to infer any expected entity type for relations – in this case the word qualifier. This is why WN18RR is used in the extensional setting.

**Codex-S** [77] and **Codex-M** [77] are datasets extracted from Wikidata and Wikipedia. They cover a wider scope and purposely contain harder facts than most KGs [77]. Consequently, these datasets prove to be more challenging for the link prediction task. Compared to WN18RR, Codex-S and Codex-M contain entity types, relation descriptions and Wikipedia page extracts. Nonetheless, Wikidata does not contain `rdfs:domain` or `rdfs:range` predicates. The property constraints present in Wikidata are harder to manipulate than the `rdfs:domain` and

<sup>6</sup>SPARQL queries were fired against DBpedia as of November 9, 2022

Table 3  
Statistics of the schemaless KGs used in the experiments

Dataset	$ \mathcal{E} $	$ \mathcal{R} $	$ \mathcal{T}_{train} $	$ \mathcal{T}_{valid} $	$ \mathcal{T}_{test} $
Codex-S	2,034	42	32,888	1,827	1,828
Codex-M	17,050	51	185,584	10,310	10,311
WN18RR	40,943	11	86,834	3,034	3,134

Table 4  
Summary of the KGEMs used in the experiments

Model Family	Model	Scoring Function	Loss Function
Geometric	TransE	$\ \mathbf{e}_h + \mathbf{e}_r - \mathbf{e}_t\ _p$	Pairwise Hinge
	TransH	$\ \mathbf{e}_{h_\perp} + \mathbf{d}_r - \mathbf{e}_{t_\perp}\ _p$	Pairwise Hinge
Semantic Matching	DistMult	$\langle \mathbf{e}_h, \mathbf{W}_r, \mathbf{e}_t \rangle$	Pairwise Hinge
	ComplEx	$\text{Re}(\mathbf{e}_h \odot \mathbf{e}_r \odot \bar{\mathbf{e}}_t)$	Pointwise Logistic
	Simple	$\frac{1}{2} \left( \langle \mathbf{e}_h^h, \mathbf{e}_r, \mathbf{e}_t^i \rangle + \langle \mathbf{e}_h^i, \mathbf{e}_r^{-1}, \mathbf{e}_t^h \rangle \right)$	Pointwise Logistic
Convolutional	ConvE	$g(\text{vec}(g(\text{concat}(\widehat{\mathbf{e}}_h, \widehat{\mathbf{e}}_r) * \omega))) \cdot \mathbf{W} \cdot \mathbf{e}_t$	Binary Cross-Entropy
	ConvKB	$\text{concat}(g([\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t] * \omega)) \cdot \mathbf{w}$	Pointwise Logistic
	R-GCN	DistMult decoder	Binary Cross-Entropy
	CompGCN	ConvE decoder	Binary Cross-Entropy

`rdfs:range` clauses found in DBpedia. As such, in our experiments Codex-S and Codex-M are used in the extensional setting. Codex-S and Codex-M initially come with already generated hard negatives. In our experiments, we do not directly use these provided negative triples. Instead, we use the same Uniform Random Negative Sampling schema as for other datasets.

## 5.2. Baseline models

In this work, the semantic awareness of the most popular KGEMs are analyzed. More specifically, the translational models TransE [3] and TransH [27], the semantic matching models DistMult [38], ComplEx [40] and Simple [42], and the convolutional models ConvE [47], ConvKB [48], R-GCN [51] and CompGCN [55] are considered. Note that in the analysis of the results in Section 6, a distinction will be made between pure convolutional KGEMs (ConvE, ConvKB) and GNNs (R-GCN, CompGCN). Although the latter have convolutional layers, they are able to capture long-range interactions between entities due to their ability to consider k-hop neighborhoods. The characteristics of the models used in our experiments are mentioned hereinafter and summarized in Table 4.

**TransE** [3] 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

$$f(h, r, t) = d(\mathbf{e}_h + \mathbf{e}_r - \mathbf{e}_t) \quad (8)$$

with  $d$  a distance function, usually the  $L1$  or  $L2$  norm.

**TransH** [27] is an extension of TransE. It allows entities to have distinct representations when involved in different relations. Specifically,  $\mathbf{e}_h$  and  $\mathbf{e}_t$  are projected into relation-specific hyperplanes with projection matrices  $\mathbf{w}_r$ . If  $(h, r, t)$  holds, the projected entities  $\mathbf{e}_{h_\perp} = \mathbf{e}_h - \mathbf{w}_r^T \mathbf{e}_h \mathbf{w}_r$  and  $\mathbf{e}_{t_\perp} = \mathbf{e}_t - \mathbf{w}_r^T \mathbf{e}_t \mathbf{w}_r$  are expected to be linked by the relation-specific translation vector  $\mathbf{d}_r$ . Thus, the scoring function is

$$f(h, r, t) = d(\mathbf{e}_{h_\perp} + \mathbf{d}_r - \mathbf{e}_{t_\perp}) \quad (9)$$

TransH often showcases better performance than TransE with only slightly more parameters [27].

**DistMult** [38] 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:

$$f(h, r, t) = \langle \mathbf{e}_h, \mathbf{W}_r, \mathbf{e}_t \rangle \quad (10)$$

It is similar to RESCAL [36] – the very first semantic matching model – but restricts relation matrices  $\mathbf{W}_r \in \mathbb{R}^{d \times d}$  to be diagonal.

**Complex** [40] 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 [32]. Its scoring function uses the Hadamard product:

$$f(h, r, t) = \text{Re} (\mathbf{e}_h \odot \mathbf{e}_r \odot \bar{\mathbf{e}}_t) \quad (11)$$

where  $\bar{\mathbf{e}}_t$  denotes the conjugate of  $\mathbf{e}_t$ .

**Simple** [42] models each fact in both a direct and an inverse form. To do so, an entity  $e$  is simultaneously represented by two vectors  $\mathbf{e}^h, \mathbf{e}^t \in \mathbb{R}^d$ . Depending on whether  $e$  appears as head or tail in a given triple, either  $\mathbf{e}^h$  or  $\mathbf{e}^t$  is used. Consequently, the two entity representations  $\mathbf{e}^h$  and  $\mathbf{e}^t$  are learned independently. Likewise, each relation  $r$  comes with a direct and an inverse vectors  $\mathbf{e}_r$  and  $\mathbf{e}_{r^{-1}}$ . The scoring function reflects the interaction between all the aforementioned entity and relation embeddings:

$$f(h, r, t) = \frac{1}{2} (\langle \mathbf{e}_h^h, \mathbf{e}_r, \mathbf{e}_t^t \rangle + \langle \mathbf{e}_h^t, \mathbf{e}_{r^{-1}}, \mathbf{e}_t^h \rangle) \quad (12)$$

**ConvE** [47] first reshapes entity and relation embeddings and then concatenates them into a 2D matrix  $[\mathbf{h}; \mathbf{r}]$ . To model the interactions between entities and relations, ConvE subsequently uses 2D convolution over embeddings and layers of nonlinear features. The output is ultimately scored against the tail embedding  $t$  using the dot product. More precisely, the following scoring function is used:

$$f(h, r, t) = g (\text{vec} (g (\text{concat} (\hat{\mathbf{e}}_h, \hat{\mathbf{e}}_r) * \omega)) \mathbf{W}) \cdot \mathbf{e}_t \quad (13)$$

where  $g$  denotes a non-linear function,  $\text{vec}$  is the vectorization operation reshaping a tensor into a vector,  $\text{concat}$  is the concatenation operator,  $*$  and  $\cdot$  denote a convolution and a dot product, respectively,  $\hat{\mathbf{e}}$  denotes a 2D reshaping of  $e$  and  $\omega$  is the set of convolutional filters.

**ConvKB** [48] also represents entities and relations as same-sized vectors. However, ConvKB does not reshape the embeddings of entities and relations. Plus, ConvKB also considers the tail embedding in the concatenation operation, thus obtaining the 2D matrix  $[\mathbf{h}; \mathbf{r}; \mathbf{t}]$  after concatenation. Convolution by a set  $\omega$  of  $T$  filters of shape  $1 * 3$  is applied on this input. The resulting  $T * 3$  feature map then passes through a dense layer with one neuron and a weight matrix  $W$ . Finally, the following scoring function assesses the plausibility of a given triple  $(h, r, t)$ :

$$f(h, r, t) = \text{concat} (g ([\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t] * \omega)) \cdot \mathbf{w} \quad (14)$$

It has been claimed that the concatenation of a set of feature maps generated by convolution should increase the learning ability of latent features compared to ConvE [25]. However, recent works point out the evaluation procedure used in the original implementation of ConvKB [53, 78], which may result in overly optimistic results on the benchmark datasets FB15K237 and WN18RR. Nonetheless, due to the popularity of this model, we choose to include it in our experiments.

**R-GCN** [51] extends the idea of applying graph convolutional networks (GCNs) to multi-relational data. R-GCN operates on local graph neighborhoods and applies a convolution operation to the neighbors of each entity. By aggregating the messages coming from all the neighbors of an entity, the embedding of the latter is updated in

accordance. Each entity thus has a hidden representation which directly depends on the hidden representations of its neighbors. This process of accumulating messages (*i.e.* the hidden representations of neighboring entities) and aggregating them so as to update the hidden representation of the central node is performed for each layer of the R-GCN model. More formally, the hidden representation of the entity  $i$  in the layer  $(l + 1)$  is defined as:

$$h_i^{(l+1)} = \sigma \left( W_0^{(l)} h_i^{(l)} + \sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} \right) \quad (15)$$

where  $\sigma(\cdot)$  can be any element-wise activation function,  $N_i^r$  denotes the set of neighboring entities of entity  $i$  considering the relation  $r$ ,  $W_r^{(l)}$  is the relation-specific weight matrix for layer  $l$  and  $c_{i,r}$  is a normalization constant. To update entity hidden representation taking into account its previous state, a self-connection is also incorporated into the activation function  $\sigma(\cdot)$ . Such layers can be stacked multiple times in order to better learn interactions and dependencies across several relational steps. However, as applying Eq.(15) directly would dramatically increase the number of parameters for KGs having lots of different relation types, basis-decomposition and block-diagonal-decomposition are proposed in [51] to reduce model parameter size and prevent overfitting. In the case of basis decomposition which is used in the original paper, the relation-specific weight matrix  $W_r^{(l)}$  is decomposed into a linear combination of basis transformation  $V_b^{(l)}$  with coefficients  $a_{rb}^{(l)}$ :

$$W_r^{(l)} = \sum_{b=1}^B a_{rb}^{(l)} V_b^{(l)} \quad (16)$$

**CompGCN** [55] improves over R-GCN by not only learning entity representations but also relation representations. Concretely, CompGCN performs a composition operation  $\phi(\cdot)$  over each edge in the neighborhood of a central node. The composed embeddings are subsequently convolved with direction-specific weight matrices  $W_O$  and  $W_I$  – for original and inverse relations, respectively. In addition to the entity representation update, relation representations are also updated individually:

$$h_r^{(l+1)} = W_{rel}^{(l)} h_r^{(l)} \quad (17)$$

where  $W_{rel}$  is a learnable transformation matrix which projects all the relations into the same embedding space as entities.

### 5.3. Implementation details and hyperparameters

For the sake of comparisons, MRR, Hits@K and Sem@K all need to rely on the same code implementation. More specifically, for R-GCN<sup>7</sup> and CompGCN<sup>8</sup>, existing implementations were reused and Sem@K values were calculated on the trained models. Other KGEMs were implemented in PyTorch. To avoid time-consuming hyperparameter tuning, we took inspiration from the hyperparameters provided by LibKGE<sup>9</sup> for Codex-S, Codex-M, WN18RR and FB15K237. However, LibKGE does not benchmark all the datasets and models used in our experiments. For such models, we stick to the hyperparameters provided by the original authors, when available. For the models with no reported best hyperparameters, as well as for the remaining datasets used in the experiments – *i.e.* DB93K, YAGO3-37K and YAGO4-19K – different combinations of hyperparameters were manually tried. We first trained our KGEMs for 1,000 epochs, then noticed the best achieved results were found around epoch 400 or below. Consequently, we stick to a maximum of 400 epochs of training as in LibKGE (except R-GCN which is trained during 4,000 epochs due to lower convergence to the best achieved results). For each positive triple in the

<sup>7</sup><https://github.com/toooooodo/RGCN-LinkPrediction>

<sup>8</sup><https://github.com/malllabisc/CompGCN>

<sup>9</sup><https://github.com/uma-pi1/kge>

Table 5

Results on YAGO3-37K. Bold fonts and gray cells denote the best achieved results and the worst achieved results among the models reported in the table, respectively. Full results are available in Appendix B, Table 8.

Model Family	Model	Rank-based				Sem@K[base]			Sem@K[wup]			Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10	S@1	S@3	S@10	S@1	S@3	S@10
Geometric	TransE	.184	.080	.198	.408	.989	.988	.988	.993	.994	.995	.897	.904	.911
	TransH	.187	.091	.199	.415	.995	.993	.993	.995	.997	.997	.901	.911	.925
Convolutional	ConvE	<b>.493</b>	<b>.350</b>	<b>.578</b>	<b>.775</b>	.933	.923	.910	.977	.977	.975	.893	.879	.871
	R-GCN	.115	.046	.110	.254	.993	.994	.993	.995	.996	.997	.933	.932	.928
	CompGCN	.399	.269	.464	.663	.998	.997	.996	.999	.999	.998	.998	.994	.982

training set, one corresponding negative triple is generated. To ensure fair comparisons between our models, we stick with Uniform Random Negative Sampling [3]. The chosen hyperparameters leading to the best performance on the validation dataset and are provided in Appendix (see Appendix A, Table 7). Recall that the objective of this work is to perform a fair and insightful assessment of the semantic awareness of KGEMs. As such, the intended purpose was not to reach optimal performance in terms of rank-based metrics. Instead, the objective is to identify a set of hyperparameters that provides satisfying and stable performance, and then study how Sem@K behaves, both for a fixed epoch – *e.g.* for the best epoch in terms of MRR – and dynamically w.r.t. the number of epochs.

## 6. Results

In the following, we perform an extensive analysis of the results obtained using the aforescribed KGEMs and KGs. For the sake of clarity, the complete range of tables and plots are placed in Appendix B and C, respectively. When necessary to support our claim, some of them are duplicated in the body text.

### 6.1. Semantic awareness of KGEMs

This section draws on the Sem@K values (see Tables 8 and 9) achieved at the best epoch in terms of MRR to provide an analysis of the semantic awareness of state-of-the-art KGEMs. In other words, for such models we only consider a snapshot of their best epochs in terms of rank-based metrics.

A major finding is that models performing well with respect to rank-based metrics are not necessarily the most competitive when it comes to their semantic capabilities. On YAGO3-37K (see Table 5) for instance, ConvE showcases impressive MRR and Hits@K values. However, it is far from being the best KGEM in terms of Sem@K, as it is outperformed by CompGCN, R-GCN and all translational models.

From a coarse-grained viewpoint, conclusions about the relative superiority of models with the distinct consideration of rank-based metrics and semantic awareness can be generalized at the level of models families. For example, semantic matching models (DistMult, ComplEx, Simple) globally achieve better MRR and Hits@K values while their semantic capabilities are in most cases lower than the ones of translational models (TransE, TransH) – see Table 8 and Table 9 for detailed results w.r.t. rank-based and semantic-oriented metrics. The respective hierarchies of such models for the benchmarked schema-defined and schemaless KGs are depicted in Fig. 4 and Fig. 5, respectively. It is clearly visible that KGEMs are grouped by family. In particular, GNNs and translational models showcase very promising semantic capabilities. GNNs are almost always the best regarding Sem@K[ext] values – not only for schemaless KGs (Fig. 5), but also for schema-defined KGs (Table 8). This means GNNs are more capable of predicting entities that have been observed as head (resp. tail) of a given relation. Translational models are very competitive in terms of Sem@K[base]. In Fig. 4, we clearly see that they consistently rank among the best performing models regarding Sem@K[base]. Interestingly, the semantic matching models DistMult, ComplEx and Simple perform relatively poorly. This observation holds regardless of the nature of the KG, as they systematically rank among the worst performing models for schema-defined (Fig. 4) and schemaless (Fig. 5) KGs.



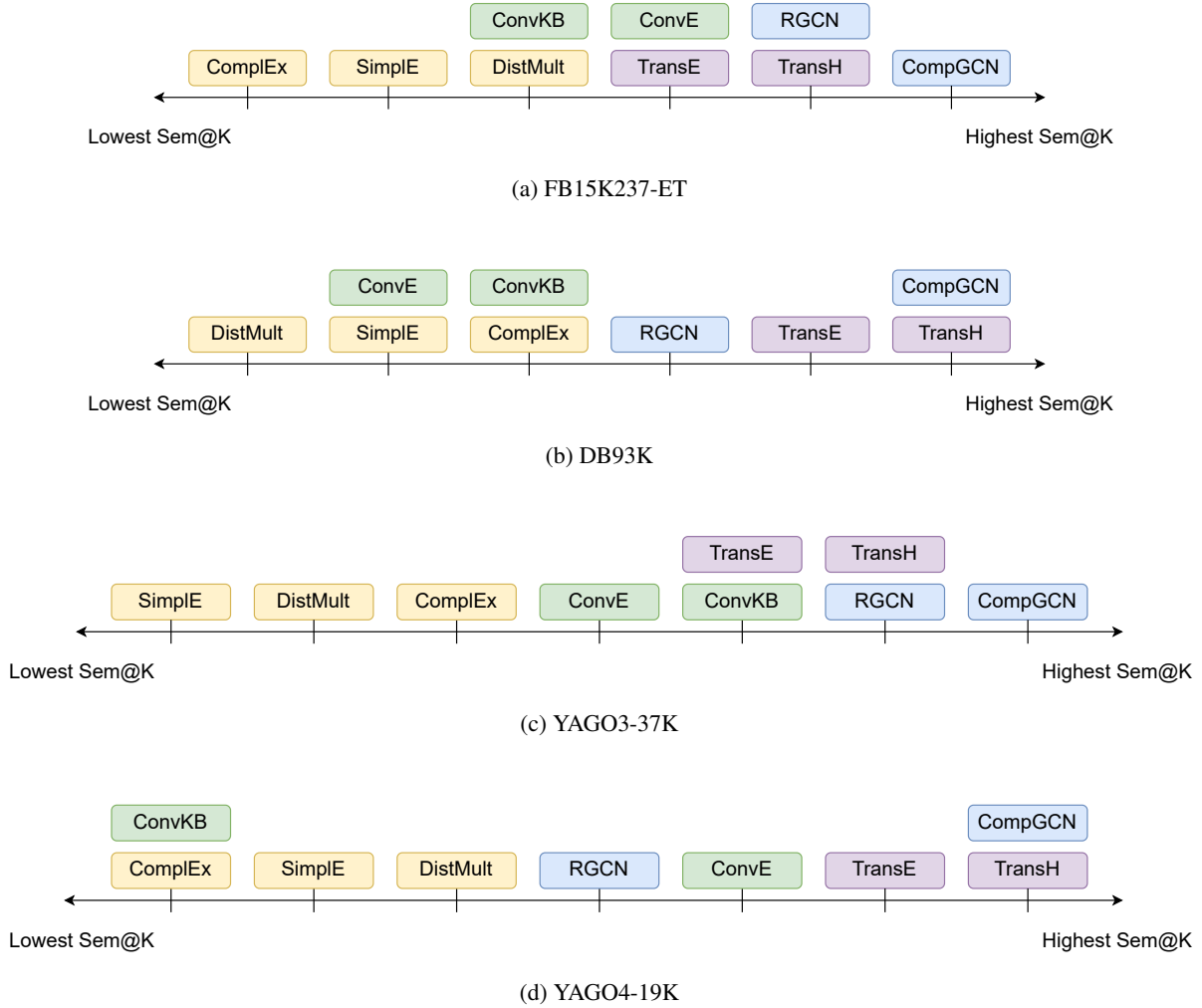


Fig. 4. Sem@K[base] comparisons between KGEMs on the 4 benchmarked schema-defined KGs. Colors indicate the family of models: blue, purple, green, and yellow cells denote GNNs, translational, convolutional, and semantic matching models, respectively. Regarding Sem@K, the relative hierarchy of models is consistent across KGs and we can clearly see that KGEMs are grouped by families of models.

Therefore, it seems that translational models are better able at recovering the semantics of entities and relations to properly predict entities that are in the domain (resp. range) of a given relation, while semantic matching models might sometimes be better at predicting entities already observed in the domain (resp. range) of a given relation (*e.g.* DistMult reaches very high Sem@K[ext] values on Codex-S, as evidenced in Fig. 5a). In cases when translational and semantic matching models provide similar results in terms of rank-based metrics, the nature of the dataset at hand – whether it is schema-defined or schemaless – might thus strongly influence the final choice of a KGEM.

Interestingly, CompGCN which is by far the most recent and sophisticated model used in our experiments, outperforms all the other models in terms of semantic awareness as, with very limited exceptions, it is the best in terms of Sem@K[base], Sem@K[ext] and Sem@K[wup]. In addition, it should be noted that R-GCN provides satisfying results as well. Except in a very few cases (*e.g.* outperformed by ComplEx on Codex-S and WN18RR in terms of Sem@1), R-GCN showcases better semantic awareness than translational models. Most of the time, R-GCN also provides comparable or even higher semantic capabilities than translational models. In particular, Sem@K values achieved with R-GCN are similar to the ones achieved with TransE and TransH (*e.g.* on FB15K237-ET and YAGO3-37K, see Tables 8a and 8c) while the latter models are actually outperformed by R-GCN in terms of

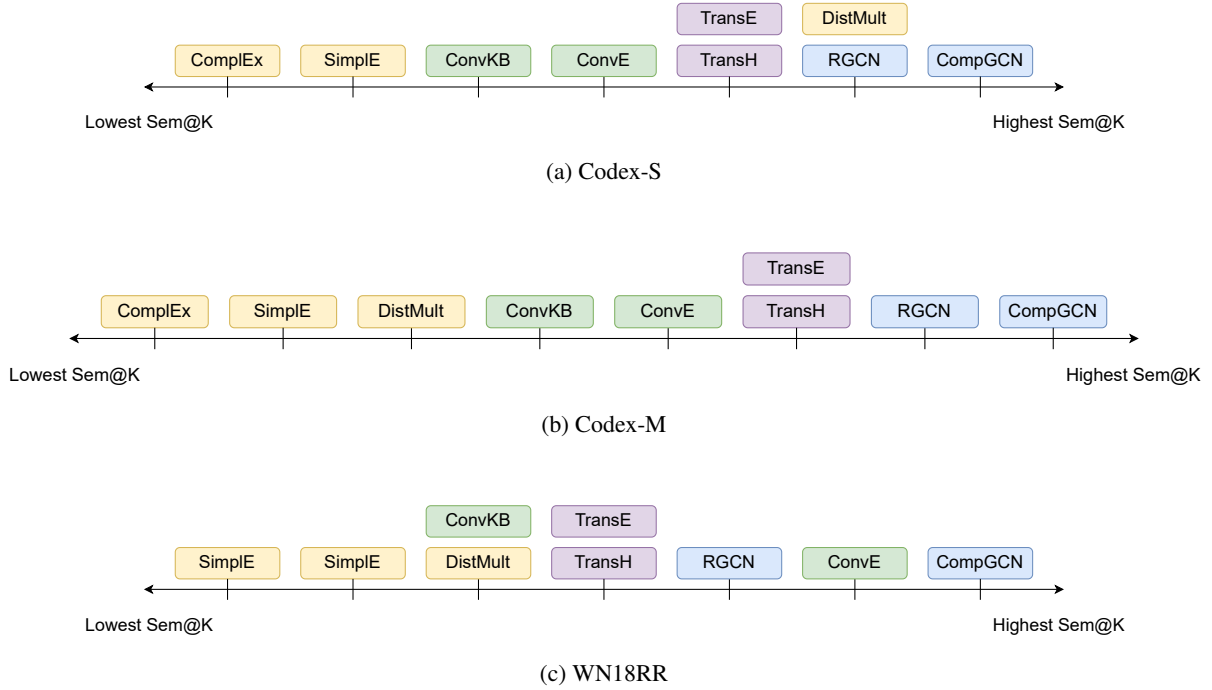


Fig. 5. Sem@K[ext] comparisons between KGEMs on the 3 benchmarked schemaless KGs. Colors indicate the family of models: blue, purple, green, and yellow cells denote GNNs, translational, convolutional, and semantic matching models, respectively. Regarding Sem@K, the relative hierarchy of models is consistent across KGs and we can clearly see that KGEMs are grouped by families of models.

Sem@K[ext]. It appears clearly on Codex-M and WN18RR (Tables 9b and 9c), although the conclusion holds for all datasets.

Our experimental results suggest that the structure of GNNs seems to be able to encode the latent semantics of the entities and relations of the graph. This ability may be attributed to the fact that, contrary to translational models which only model the local neighborhood of each triple, GNNs update entity embeddings (and relation embeddings in the case of CompGCN) based on the information found in the extended,  $h$ -hop neighborhood of the central node. Where translational and semantic matching models treat each triple independently, GNNs model interactions between entities on a large range of semantic relations. It is likely that this extended neighborhood comprises signals or patterns that help the model infer the classes of entities, thus providing very promising semantic capabilities in all experimental conditions.

## 6.2. Dynamic appraisal of KGEM semantic awareness

For certain models, rank-based metrics performance and semantic capabilities improve jointly. For others, the enhancement of their performance in terms of rank-based metrics comes at the expense of their semantic awareness. Interestingly, trends emerge relatively to families of models. First, we observe that a trade-off exist for semantic matching models. Results are particularly striking on FB15K237-ET (see Fig. 6), where it is obvious that after reaching the best Sem@K values after a few epochs, Sem@K values of DistMult and ComplEx quickly drop while MRR continues rising. Conversely, translational models are more robust to Sem@K degradation throughout the epochs. Even though the best achieved Sem@K values are also reported in the very first epochs, once these values are reached they remain stable for the remaining epochs of training. This might be due to the geometric nature of such KGEMs, which will organize the representation space so as to  $h + r$  falls in a region of the space where neighboring entities of the ground-truth tail  $t$  all are entities of the expected type. This is highly related to the block structure property, which is a common statistical pattern found in KGs [37]. It refers to the fact that entities can naturally belong to different groups (blocks), such that all the entities of a given group are linked to entities of another group

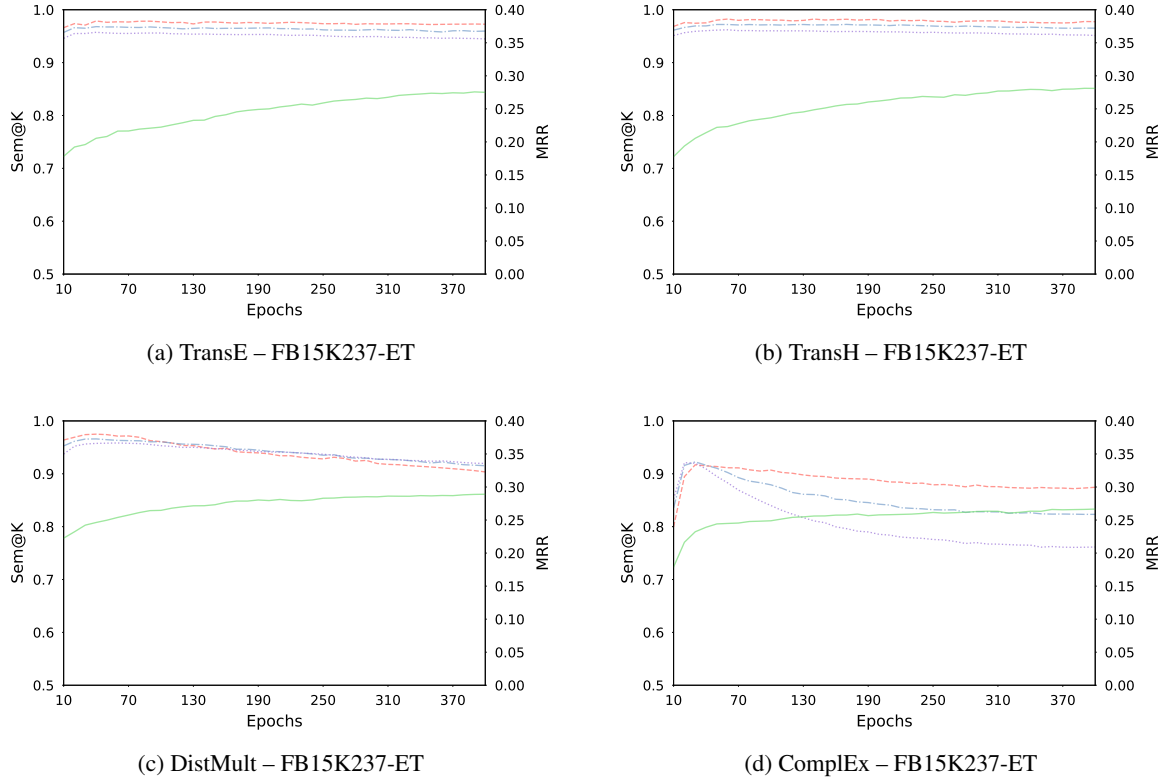


Fig. 6. Evolution of MRR (—), Sem@1 (---), Sem@3 (···), and Sem@10 (· · ·) for translational (TransE, TransH) and semantic matching models (DistMult, ComplEx) on FB15K237-ET.

through the same relationship. In this case, each group comprises entities of the same class. Translational models will naturally group entities of the same class in the same region of the representation space, as this is determined by the translation vector in that space.

Plots of the joint evolution of MRR and Sem@K values show that most of the KGEMs reaches their best Sem@K values after a few number of epochs. This means that predictions get semantically valid in the early stages of training. As previously mentioned, Sem@K then usually start to decrease, as it has been noted for semantic matching models in particular. To this respect, an excerpt of the head and tail predictions of DistMult on YAGO3-37K is depicted in Fig. 7. Even though the ground-truth entity does not show up in neither the head nor the tail top-K list, we clearly see that after only 30 epochs of training, predictions made by DistMult are more meaningful than after 400 epochs of training. This relative trade-off between making semantically valid predictions and predictions that comprise the ground-truth entity higher in the top-K list calls for finding a compromise in terms of training. The LP task is usually addressed in terms of rank-based metrics only, hence the choice of performing more and more training epochs so as to find the optimal KGEM in terms of MRR and Hits@K. However, as discussed in the present work, adding training steps may improve KGEM performance at the expense of its semantic awareness. In many cases, rank-based metrics values only slightly increase, whereas Sem@K values drastically drop. For instance, comparing MRR vs. Sem@K evolution of DistMult on FB15K237-ET (Fig. 8c), we clearly see that after a moderate number of epochs, any additional epoch of training only provides a very slight improvement in terms of MRR, while it is very detrimental to Sem@K values. Depending on the use case, such a decline in the semantic capabilities of the model is not desirable, and a compromise is to be found between training more to increase KGEM predictive performance and stopping training early enough so as not to deteriorate its semantic awareness too much.

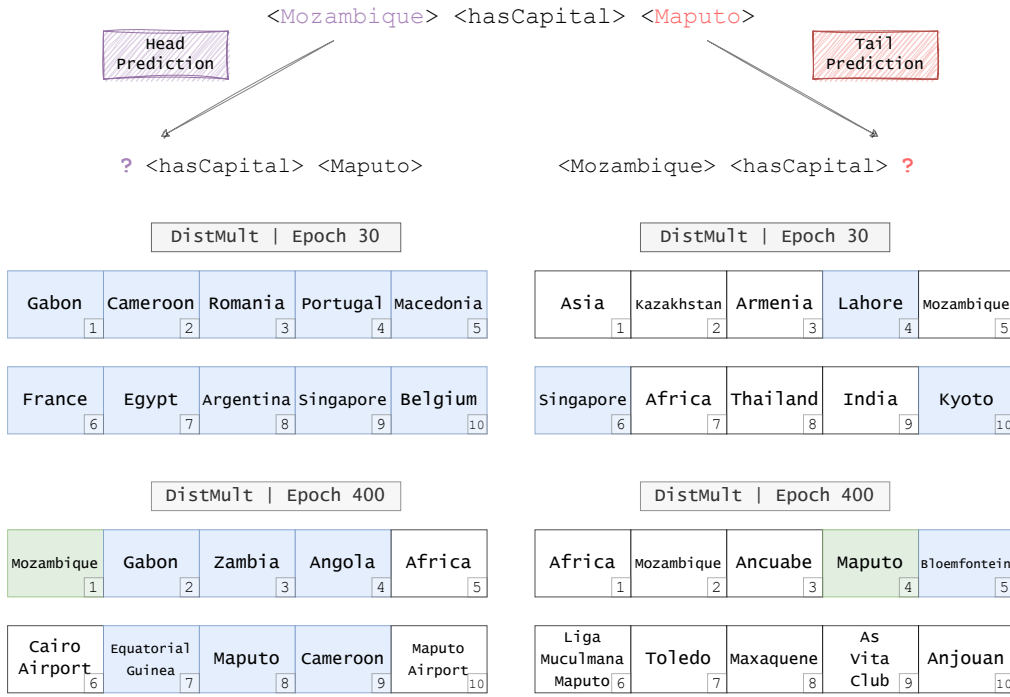


Fig. 7. Top-ten ranked entities for head and tail predictions at epochs 30 and 400 for a sample triple from YAGO3-37K. Green, blue and white cells respectively denote the ground-truth entity, entities other than the ground-truth and semantically valid, and entities other than the ground-truth and semantically invalid. In this case, semantic validity is based on the domain and range of the relation  $\langle \text{hasCapital} \rangle$ .

### 6.3. On the use of Sem@K for different kinds of KGs

As reported in Table 1, KGs based upon a schema and a class hierarchy are candidates for the computation of all the versions of Sem@K. For the schema-defined KGs used in our experiments, we choose to report values regarding all these metrics so as to enable multi-view comparisons across models. From Table 8 it can be clearly seen that the relative superiority of models is consistent throughout the different Sem@K definitions. From a higher perspective, this means that even for schema-defined KGs with a hierarchy class, Sem@K[ext] is already a good proxy. This may be a good option to only rely on the Sem@K[ext] whenever the computation of Sem@K[base] is too expensive, due to the entity type checking part. This is even more true for Sem@K[wup], which requires an additional step of semantic relatedness computation.

## 7. Discussion

Three major research questions have been formulated in Section 1 and recalled in Section 3.2. Based on the analysis presented in Section 6, we address each research question individually:

- RQ1: how semantic-aware agnostic KGEMs are?

From a coarse-grained viewpoint, we noted that KGEMs trained in an agnostic way prove capable of giving higher scores to semantically valid triples. However, disparities exist between models. Interestingly, these disparities seem to derive from the family of such models. Globally, translational models and GNNs – represented by R-GCN and CompGCN in this work – provide promising results. It appears that the two aforementioned families of KGEMs are better able than semantic matching models (DistMult, ComplEx, Simple) at recovering the semantics of entities

and relations to give higher score to semantically valid triples. In fact, semantic matching models are almost systematically the worst performing models in terms of semantic awareness. From a dynamic standpoint, it is worth noting the high semantic capabilities of KGEMs reached during the first epochs of training. In most cases, this is even during the first epochs that the optimal semantic awareness is attained.

- RQ2: how KGEM semantic awareness’ evaluation should adapt to the typology of KGs?

Drawing on the initial version of Sem@K as presented in [15] – referred to herein as Sem@K[base] – an issue is quickly encountered when it comes to schemaless KGs, which do not contain any `rdfs:domain` (resp. `rdfs:range`) clause to indicate the class that candidate heads (resp. tails) should belong to. Our work introduces Sem@K[ext] – a new version of Sem@K that overcomes the aforementioned limitation. In addition, even with schema-defined KGs, Sem@K[base] is not necessarily sufficient in itself. This metric can be further enriched whenever a KG comes with a class hierarchy. In Section 4.3, we integrate class hierarchy into Sem@K by means of a similarity measure between concepts. We subsequently provide an example using the Wu-Palmer similarity score. The resulting Sem@K[wup] is used in the experiments in Section 6 and provide a finer-grained measure of KGEMs semantic awareness.

- RQ3: does the evaluation of KGEM semantic awareness offers different conclusions on the relative superiority of some KGEMs?

A major finding is that models performing well with respect to rank-based metrics are not necessarily the most competitive regarding their semantic capabilities. We previously noted that translational models globally showcase better Sem@K values compared to semantic matching models. Considering MRR and Hits@K, the opposite conclusion is often drawn. Hence, the performance of KGEMs in terms of rank-based metrics are not indicative of their semantic capabilities. The only exception that might exist is for GNNs that perform well both in terms of rank-based metrics and semantic-oriented measures.

The answers provided to the research questions also lead to consider new matters. As evidenced in Table 8 and Table 9, some KGs are more challenging with regard to Sem@K results. Due to its tailored extraction strategy that purportedly favored difficult relations to feature in the validation and test sets, YAGO4-19K is the schema-defined KG with the lowest achieved Sem@K. This observation raises a deeper question: what characteristics of a KG make it inherently challenging for KGEMs to recover the semantics of entities and relations? An extensive study of the influence of KGs characteristics on the semantic capabilities of KGEMs would require to benchmark them on a broad set of KGs with varying dimensions, so as to determine those that are the most prevalent. Such characteristics can be the total number of relations, the average number of instances per class, or a combination of different factors. We leave this experimental study for a future work.

Recall that this work is motivated by the possibility of going beyond a mere assessment of KGEM performance regarding rank-based metrics. We showed that these metrics only evaluate one aspect of such models, somehow providing a partial view on the quality of KGEMs. Our proposal for further assessing KGEM semantic capabilities aims at diving deeper into their predictive expressiveness and measuring to what extent their predictions are semantically valid. However, this second evaluation component does not shed full light on the respective KGEM peculiarities. Other evaluation components may be added, such as the storage and computational requirements of KGEMs [79, 80] and the environmental impact of their training and testing procedure [81]. Furthermore, the explainability of KGEMs is another dimension that deserves great attention [82, 83].

## 8. Conclusion

In this work, we consider the link prediction task and extend our previously introduced Sem@K metric to measure the ability of KGEMs to assign higher scores to triples that are semantically valid. In particular, to adapt to different types of KGs (*e.g.*, schemaless, class hierarchy), we introduce Sem@K[base], Sem@K[ext], or Sem@K[wup]. Compared with the traditional evaluation approach that solely relies on rank-based metrics, we show that the evaluation procedure is enhanced with the addition of semantic-oriented metrics that bring an additional perspective on

KGEM quality. Our experiments with different types of KGs highlight that there is no clear correlation between the performance of KGEMs in terms of traditional rank-based metrics versus their performance regarding semantic-oriented ones. In some cases, however, a trade-off does exist. Consequently, this calls for monitoring KGEM training under more scrutiny. Our experiments also point out that most of the conclusions that have been drawn actually hold at the level of families of models.

In future work, we will conduct experiments with additional evaluation components and propose evaluation metrics that result from the combination of such components.

## Acknowledgements

This work is supported by the AILES PIA3 project (see <https://www.projetailes.com/>). Experiments presented in this paper were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

## Appendix A. Hyperparameters

For datasets with no reported optimal hyperparameters, grid-search based on curated hyperparameters were performed. The full hyperparameter space is provided in Table 6. Chosen hyperparameters for each pair of KGEM and dataset are provided in Table 7.

Table 6  
Hyperparameter search space

Hyperparameters	Range
Batch Size	{128, 256, 512, 1024, 2048}
Embedding Dimension	{50, 100, 150, 200}
Regularizer Type	{None, $L1$ , $L2$ }
Regularizer Weight ( $\lambda$ )	{ $1e^{-1}$ , $1e^{-2}$ , $1e^{-3}$ , $1e^{-4}$ , $1e^{-5}$ }
Learning Rate ( $lr$ )	{0.005, 0.003, 0.001, 0.0005, 0.0003, 0.0001}

## Appendix B. Results achieved with the best reported hyperparameters

Results achieved with the best reported hyperparameters are presented in Tables 8 and 9.

## Appendix C. Evolution of MRR and Sem@K values with respect to the number of epochs

The evolution of MRR and Sem@K values with respect to the number of epochs is presented in Fig. 8, 9, 10, 11, 12, 13, and 14. For equity and clarity sakes, we choose to present 2 KGEMs for each family of model (translational models, semantic matching models, CNNs, and GNNs). Regarding semantic matching models, DistMult and ComplEx are chosen, as the evolution of their MRR and Sem@K values is less erratic than Simple. The evolution of MRR and Sem@K values for Simple are made available on the GitHub repository of the datasets<sup>10</sup>.

<sup>10</sup><https://github.com/nicolas-hbt/benchmark-sematk>

Table 7

Chosen hyperparameters for schema-defined and schemaless KGs used in the experiments.  $|\mathcal{B}|$ ,  $d$ ,  $lr$ , and  $\lambda$  denote the batch size, embedding dimension, learning rate, and regularization weight, respectively. We experimentally found that  $L2$  regularizer systematically worked the best. We therefore decide not to refer to it in the table. For ConvKB, the number of filters  $\mathcal{T}$  was set to 128. Parameters specific to R-GCN and CompGCN were left as defined in the original implementations.

Model	Hyperparameters	FB15K237-ET	DB93K	YAGO3-37K	YAGO4-19K	Codex-S	Codex-M	WN18RR
TransE	$ \mathcal{B} $	512	256	256	512	128	128	512
	$d$	100	200	150	100	100	100	100
	$lr$	$1e^{-3}$	$5e^{-3}$	$5e^{-3}$	$5e^{-2}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$
TransH	$ \mathcal{B} $	512	256	256	512	128	128	512
	$d$	100	200	150	100	100	100	100
	$lr$	$1e^{-3}$	$5e^{-3}$	$5e^{-3}$	$5e^{-2}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$
DistMult	$ \mathcal{B} $	1024	1024	1024	1024	1024	1024	1024
	$d$	100	200	150	100	100	100	100
	$lr$	$1e^{-3}$	$1e^{-3}$	$5e^{-3}$	$5e^{-2}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	0	0	0	0
ComplEx	$ \mathcal{B} $	1024	1024	1024	1024	1024	1024	1024
	$d$	100	200	150	100	100	100	100
	$lr$	$1e^{-3}$	$5e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-2}$	$1e^{-1}$	$1e^{-5}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$
Simple	$ \mathcal{B} $	1024	1024	1024	1024	1024	1024	1024
	$d$	100	200	150	100	100	100	100
	$lr$	$1e^{-3}$	$5e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-2}$	$1e^{-1}$	$1e^{-5}$	0	$1e^{-2}$	$1e^{-2}$	0
ConvE	$ \mathcal{B} $	512	256	256	512	512	512	512
	$d$	200	200	200	200	200	200	200
	$lr$	$1e^{-3}$	$5e^{-3}$	$5e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$
ConvKB	$ \mathcal{B} $	512	256	256	512	512	512	512
	$d$	100	100	100	100	100	100	100
	$lr$	$1e^{-3}$	$1e^{-3}$	$5e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$
R-GCN	$d$	500	500	500	500	500	500	500
	$lr$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$
	$\lambda$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$
CompGCN	$ \mathcal{B} $	1024	1024	1024	1024	1024	1024	1024
	$d$	200	200	200	200	200	200	200
	$lr$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$	$1e^{-3}$
	$\lambda$	0	0	0	0	0	0	0

Table 8

Rank-based and semantic-based results on the schema-defined knowledge graphs. Bold fonts indicate which model performs best with respect to a given metric.

(a) FB15K237-ET

Model Family	Model	Rank-based				Sem@K[base]			Sem@K[wup]			Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10	S@1	S@3	S@10	S@1	S@3	S@10
Geometric	TransE	.273	.182	.302	.453	.978	.964	.949	.983	.972	.961	.888	.870	.845
	TransH	.281	.187	.306	.458	.981	.968	.955	.986	.976	.966	.896	.873	.855
Semantic Matching	DistMult	.289	.206	.313	.456	.907	.919	.922	.919	.930	.934	.842	.853	.849
	CompLex	.267	.185	.292	.431	.874	.823	.761	.897	.856	.804	.822	.770	.7005
	SimplE	.257	.180	.275	.416	.895	.878	.848	.911	.896	.869	.839	.817	.776
Convolutional	ConvE	.310	.211	.337	.490	.972	.970	.961	.977	.973	.968	.883	.880	.870
	ConvKB	.251	.159	.261	.411	.918	.897	.872	.935	.918	.899	.842	.812	.781
	R-GCN	.238	.153	.257	.412	.980	.970	.953	.985	.976	.961	.902	.889	.861
	CompGCN	<b>.337</b>	<b>.227</b>	<b>.365</b>	<b>.530</b>	<b>.998</b>	<b>.996</b>	<b>.992</b>	<b>.999</b>	<b>.997</b>	<b>.995</b>	<b>.993</b>	<b>.986</b>	<b>.976</b>

(b) DB93K

Model Family	Model	Rank-based				Sem@K[base]			Sem@K[wup]			Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10	S@1	S@3	S@10	S@1	S@3	S@10
Geometric	TransE	.233	.145	.275	.397	.985	.975	.961	.991	.987	.979	.767	.730	.689
	TransH	.236	.147	.278	.399	<b>.988</b>	<b>.979</b>	<b>.968</b>	<b>.993</b>	<b>.990</b>	<b>.981</b>	.769	.739	.694
Semantic Matching	DistMult	.261	.202	.287	.369	.865	.831	.790	.890	.865	.833	.716	.670	.617
	CompLex	.287	.213	.325	.417	.941	.917	.877	.955	.937	.907	.815	.767	.698
	SimplE	.252	.202	.274	.339	.896	.871	.837	.915	.895	.867	.774	.719	.659
Convolutional	ConvE	.256	.183	.289	.392	.871	.862	.862	.882	.874	.874	.764	.750	.734
	ConvKB	.178	.121	.199	.283	.883	.895	.881	.902	.895	.881	.742	.702	.659
	R-GCN	.208	.144	.233	.319	.969	.959	.945	.979	.972	.961	.819	.785	.740
	CompGCN	<b>.319</b>	<b>.249</b>	<b>.351</b>	<b>.446</b>	<b>.988</b>	.976	.967	.991	.982	.975	<b>.953</b>	<b>.918</b>	<b>.878</b>

(c) YAGO3-37K

Model Family	Model	Rank-based				Sem@K[base]			Sem@K[wup]			Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10	S@1	S@3	S@10	S@1	S@3	S@10
Geometric	TransE	.184	.080	.198	.408	.989	.988	.988	.993	.994	.995	.897	.904	.911
	TransH	.187	.091	.199	.415	.995	.993	.993	.995	.997	.997	.901	.911	.925
Semantic Matching	DistMult	.225	.112	.251	.465	.885	.940	.959	.921	.962	.980	.795	.851	.886
	CompLex	.430	.250	.551	.780	.740	.820	.859	.911	.935	.950	.662	.735	.794
	SimplE	.311	.081	.468	.749	.377	.733	.866	.786	.901	.944	.310	.640	.777
Convolutional	ConvE	<b>.493</b>	<b>.350</b>	<b>.578</b>	<b>.775</b>	.933	.923	.910	.977	.977	.975	.893	.879	.871
	ConvKB	.305	.162	.352	.631	.979	.979	.978	.990	.990	.990	.899	.896	.904
	R-GCN	.115	.046	.110	.254	.993	.994	.993	.995	.996	.997	.933	.932	.928
	CompGCN	.399	.269	.464	.663	<b>.998</b>	<b>.997</b>	<b>.996</b>	<b>.999</b>	<b>.999</b>	<b>.998</b>	<b>.998</b>	<b>.994</b>	<b>.982</b>

(d) YAGO4-19K

Model Family	Model	Rank-based				Sem@K[base]			Sem@K[wup]			Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10	S@1	S@3	S@10	S@1	S@3	S@10
Geometric	TransE	.749	.651	.833	.901	.975	.890	.843	.980	.914	.870	.844	.647	.547
	TransH	.752	.656	.829	.898	.979	<b>.909</b>	<b>.868</b>	<b>.982</b>	<b>.931</b>	<b>.891</b>	.848	.656	.564
Semantic Matching	DistMult	.863	.839	.881	.898	.931	.687	.534	.933	.686	.551	.901	.556	.376
	CompLex	.890	.881	.897	.903	.923	.627	.479	.923	.635	.496	.909	.524	.349
	SimplE	.808	.757	.848	.883	.891	.646	.500	.894	.654	.516	.840	.530	.367
Convolutional	ConvE	.901	.891	<b>.908</b>	.913	.977	.858	.844	.977	.869	.854	.948	.731	.709
	ConvKB	.828	.772	.883	.908	.865	.614	.564	.870	.639	.590	.793	.460	.361
	R-GCN	.893	.885	.896	.903	.969	.832	.727	.970	.841	.741	.930	.670	.522
	CompGCN	<b>.907</b>	<b>.903</b>	<b>.908</b>	<b>.918</b>	<b>.980</b>	.906	.862	.981	.912	.870	<b>.968</b>	<b>.815</b>	<b>.725</b>



Table 9

Rank-based and semantic-based results on the schemaless knowledge graphs. Bold fonts indicate which model performs best with respect to a given metric.

(a) Codex-S

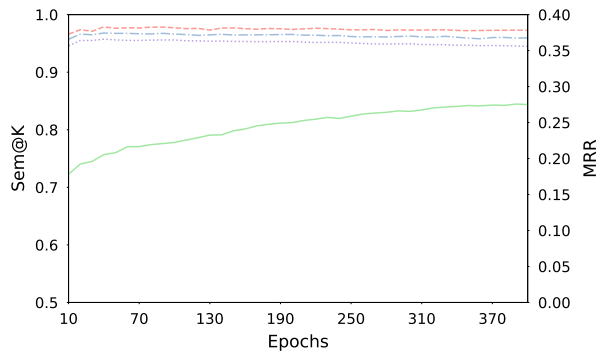
Model Family	Model	Rank-based				Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10
Geometric	TransE	.354	.223	.409	.620	.927	.900	.873
	TransH	.355	.225	.410	.621	.928	.901	.875
Semantic Matching	DistMult	.442	.336	.487	.661	.935	.919	.887
	ComplEx	<b>.511</b>	<b>.426</b>	<b>.552</b>	<b>.668</b>	.872	.786	.694
	SimpleE	.464	.368	.514	.641	.876	.789	.727
Convolutional	ConvE	.379	.259	.439	.610	.912	.913	.886
	ConvKB	.453	.361	.493	.632	.858	.806	.769
	R-GCN	.335	.225	.381	.556	.936	.921	.889
	CompGCN	.380	.268	.427	.597	<b>.993</b>	<b>.986</b>	<b>.974</b>

(b) Codex-M

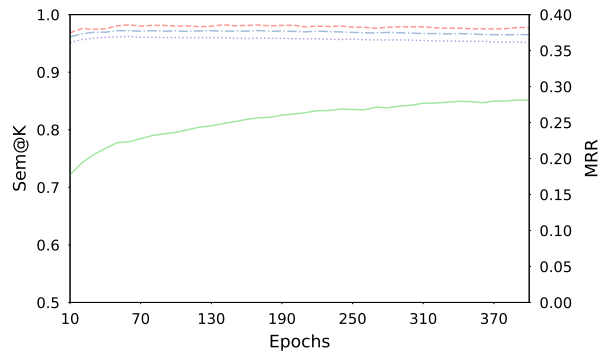
Model Family	Model	Rank-based				Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10
Geometric	TransE	.262	.188	.284	.405	.889	.855	.839
	TransH	.266	.190	.291	.415	.894	.862	.842
Semantic Matching	DistMult	.268	.199	.291	.402	.726	.749	.765
	ComplEx	.274	.213	.298	.386	.768	.687	.608
	SimpleE	.269	.212	.289	.378	.799	.713	.629
Convolutional	ConvE	.255	.184	.280	.395	.863	.839	.814
	ConvKB	.230	.156	.250	.376	.797	.771	.770
	R-GCN	.185	.110	.201	.340	.944	.920	.887
	CompGCN	<b>.314</b>	<b>.237</b>	<b>.346</b>	<b>.460</b>	<b>.996</b>	<b>.992</b>	<b>.983</b>

(c) WN18RR

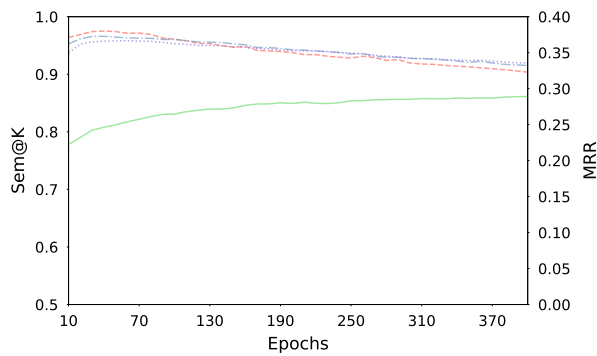
Model Family	Model	Rank-based				Sem@K[ext]		
		MRR	H@1	H@3	H@10	S@1	S@3	S@10
Geometric	TransE	.186	.032	.303	.455	.770	.756	.715
	TransH	.191	.034	.306	.458	.773	.760	.717
Semantic Matching	DistMult	.399	.372	.413	.444	.795	.732	.688
	ComplEx	.430	.400	.446	.487	.715	.669	.673
	SimpleE	.397	.375	.406	.438	.690	.598	.673
Convolutional	ConvE	.406	.375	.422	.459	.865	.854	.871
	ConvKB	.356	.302	.391	.444	.712	.713	.732
	R-GCN	.382	.345	.402	.441	.836	.797	.752
	CompGCN	<b>.471</b>	<b>.437</b>	<b>.483</b>	<b>.536</b>	<b>.970</b>	<b>.946</b>	<b>.927</b>



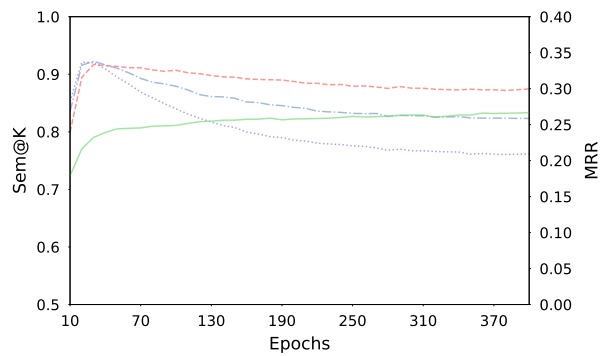
(a) TransE – FB15K237-ET



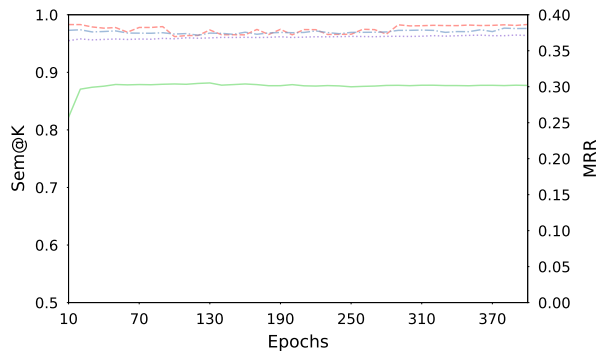
(b) TransH – FB15K237-ET



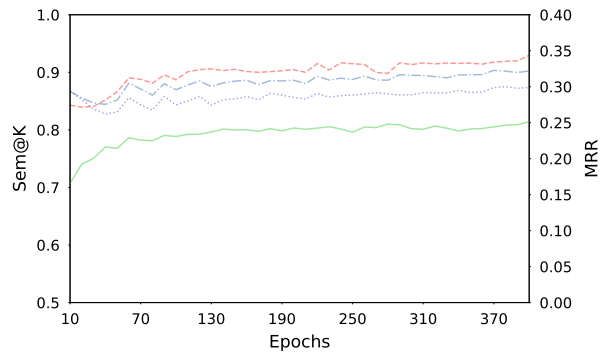
(c) DistMult – FB15K237-ET



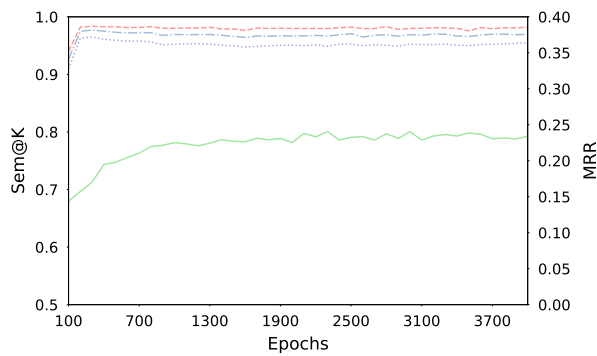
(d) Complex – FB15K237-ET



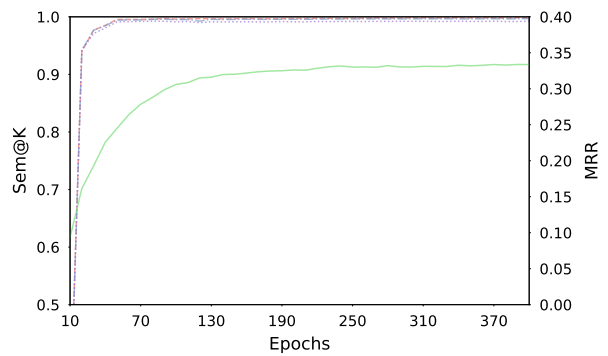
(e) ConvE – FB15K237-ET



(f) ConvKB – FB15K237-ET

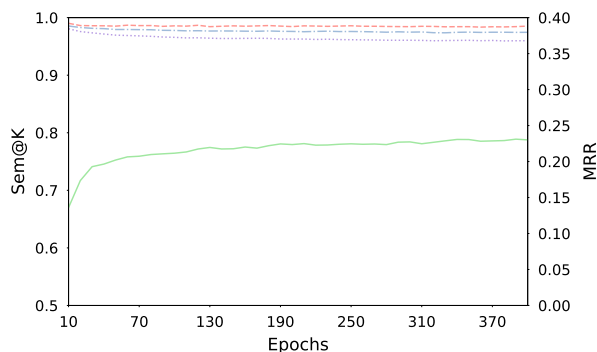


(g) R-GCN – FB15K237-ET

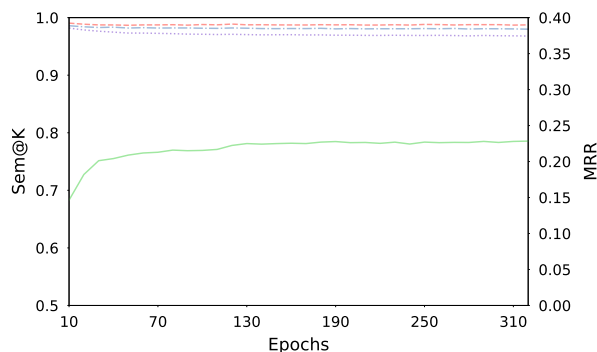


(h) CompGCN – FB15K237-ET

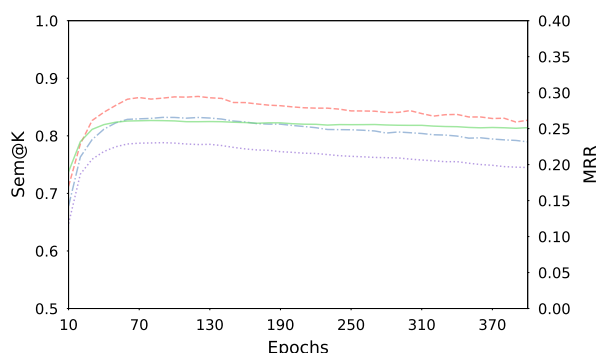
Fig. 8. Evolution of MRR (—), Sem@1 (---), Sem@3 (-.-), and Sem@10 (⋯) on FB15K237-ET



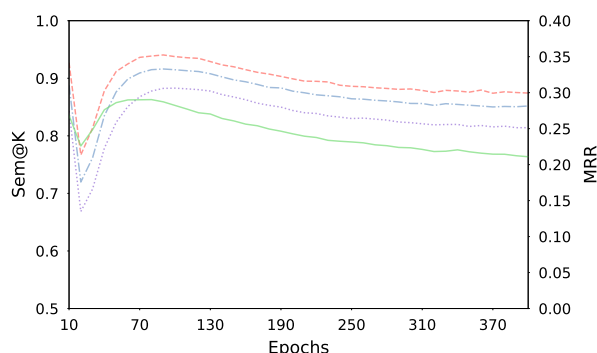
(a) TransE – DB93K



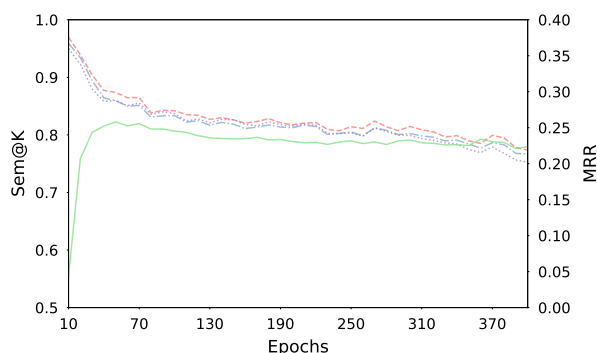
(b) TransH – DB93K



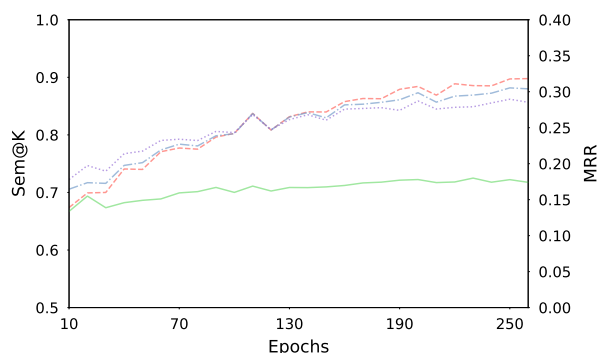
(c) DistMult – DB93K



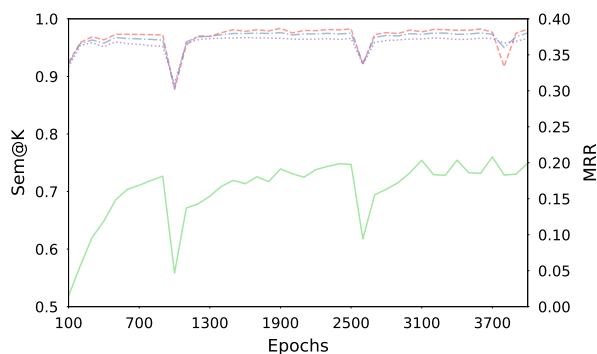
(d) ComplEx – DB93K



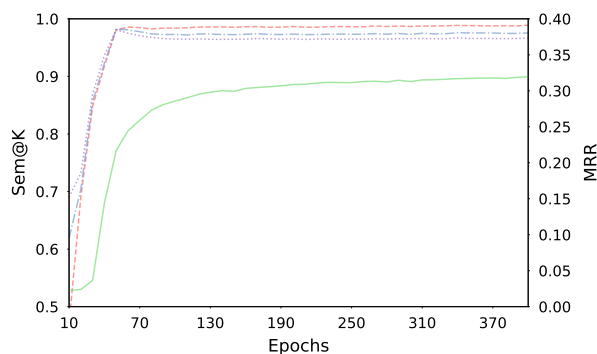
(e) ConvE – DB93K



(f) ConvKB – DB93K

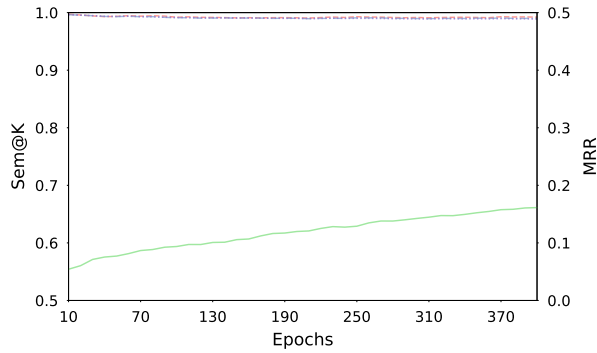


(g) R-GCN – DB93K

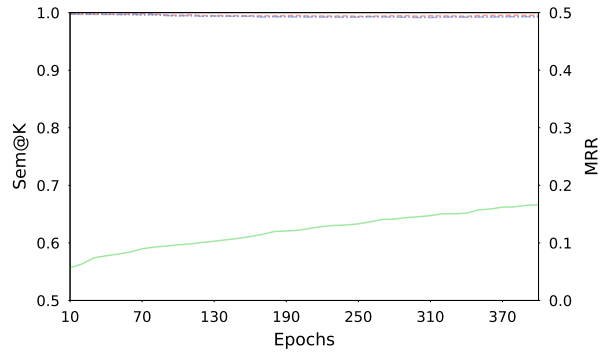


(h) CompGCN – DB93K

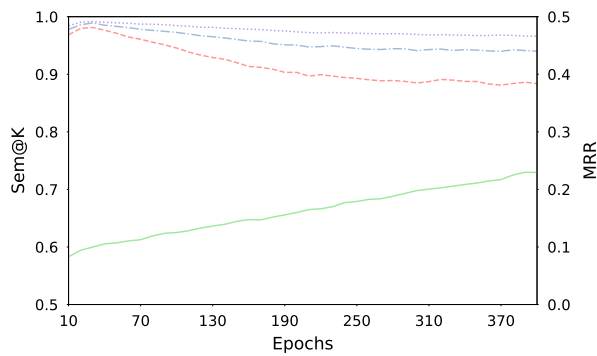
Fig. 9. Evolution of MRR (—), Sem@1 (---), Sem@3 (-.-), and Sem@10 (···) on DB93K



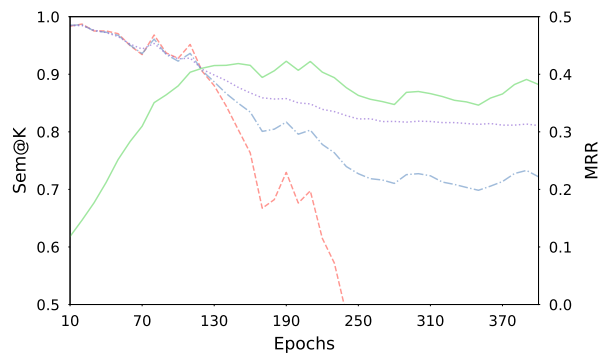
(a) TransE – YAGO3-37K



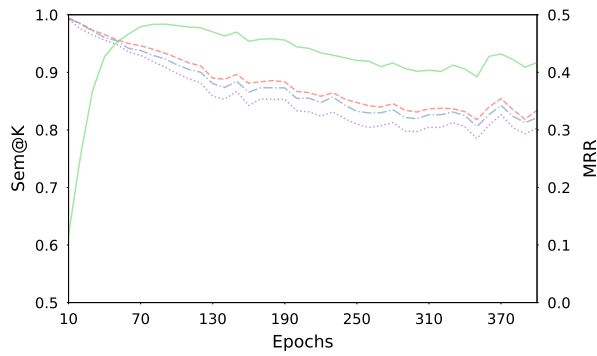
(b) TransH – YAGO3-37K



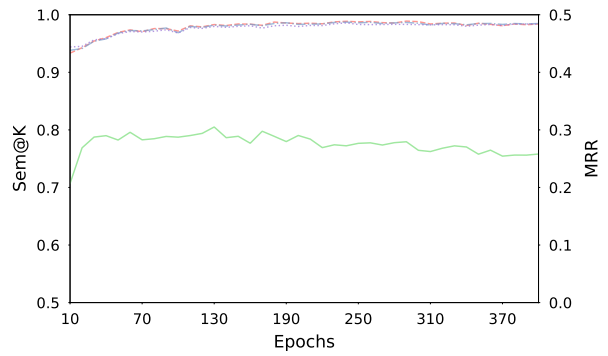
(c) DistMult – YAGO3-37K



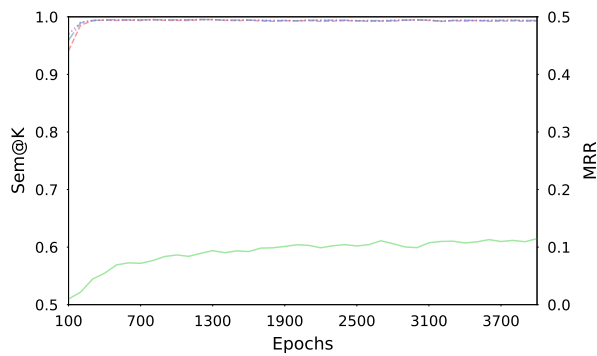
(d) ComplEx – YAGO3-37K



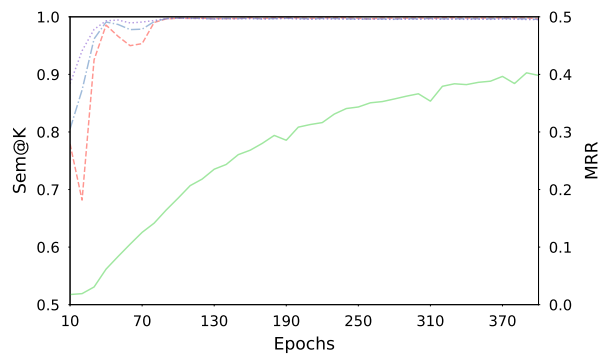
(e) ConvE – YAGO3-37K



(f) ConvKB – YAGO3-37K

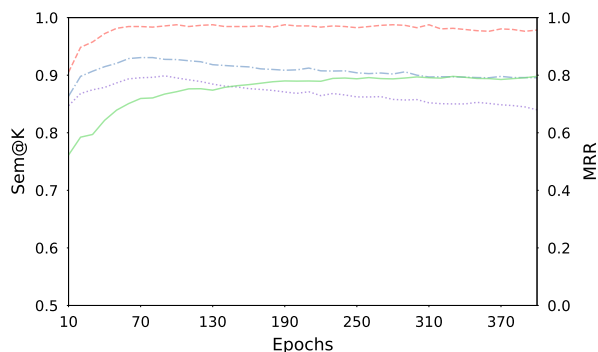


(g) R-GCN – YAGO3-37K

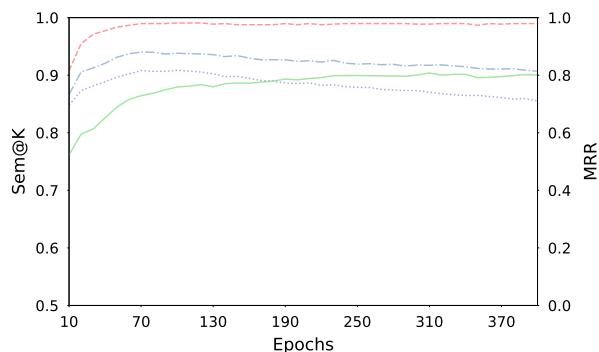


(h) CompGCN – YAGO3-37K

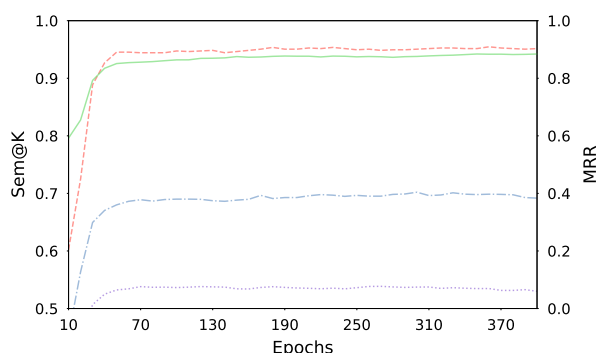
Fig. 10. Evolution of MRR (—), Sem@1 (---), Sem@3 (---), and Sem@10 (····) on YAGO3-37K



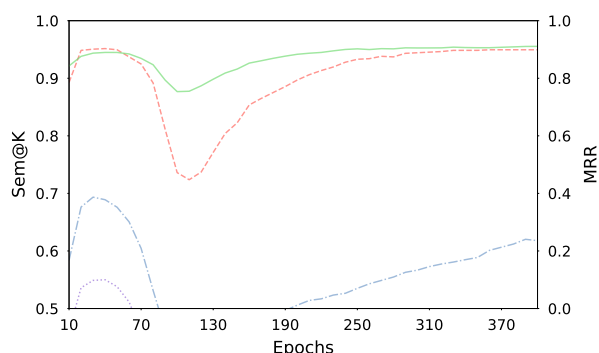
(a) TransE – YAGO4-19K



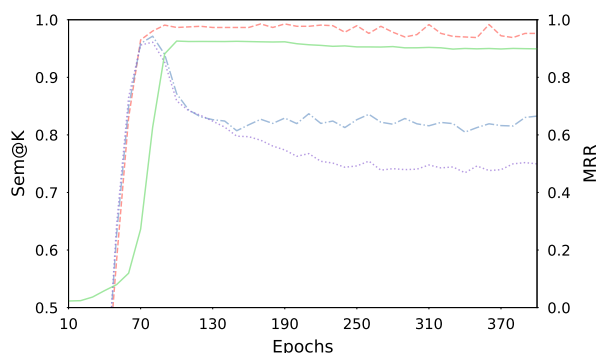
(b) TransH – YAGO4-19K



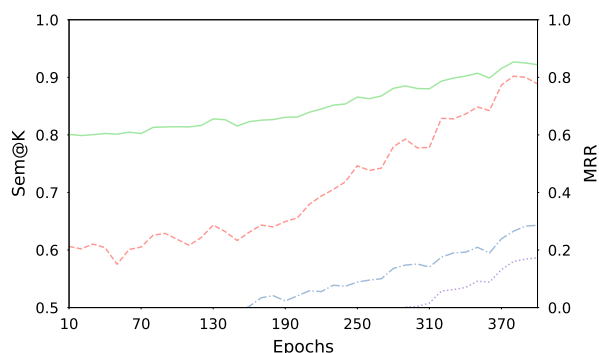
(c) DistMult – YAGO4-19K



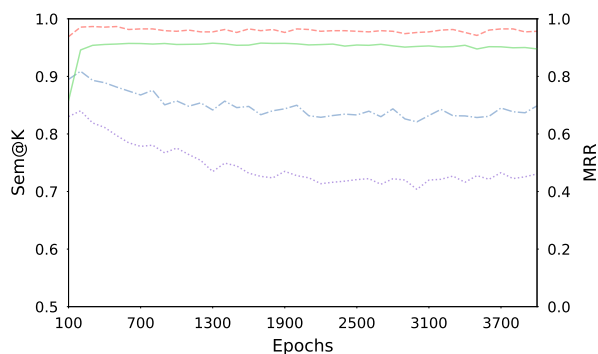
(d) ComplEx – YAGO4-19K



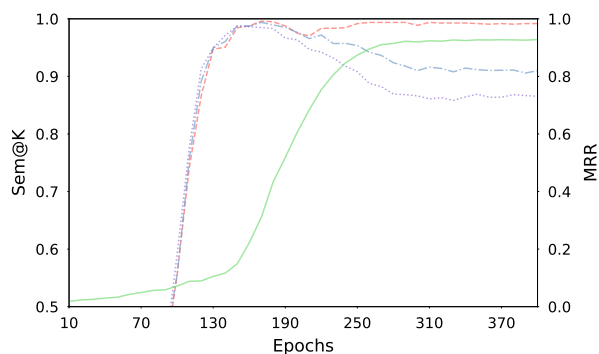
(e) ConvE – YAGO4-19K



(f) ConvKB – YAGO4-19K

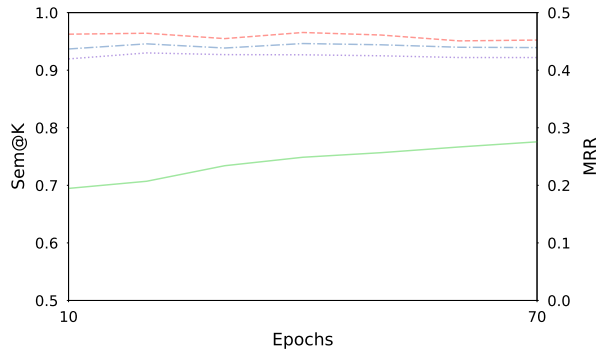


(g) R-GCN – YAGO4-19K

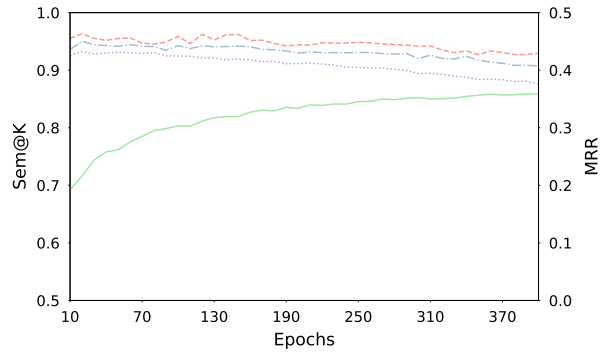


(h) CompGCN – YAGO4-19K

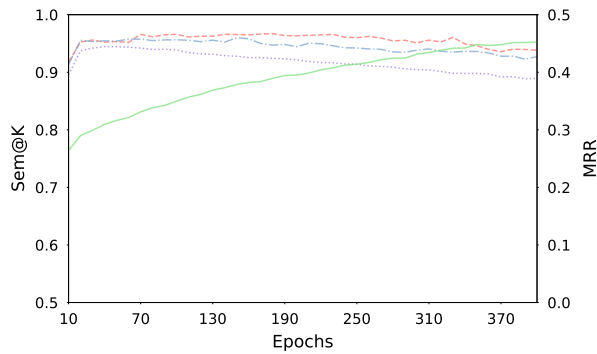
Fig. 11. Evolution of MRR (—), Sem@1 (---), Sem@3 (— · —), and Sem@10 (· · ·) on YAGO4-19K



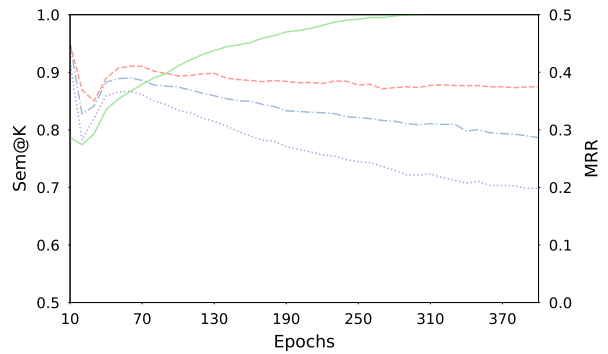
(a) TransE – Codex-S



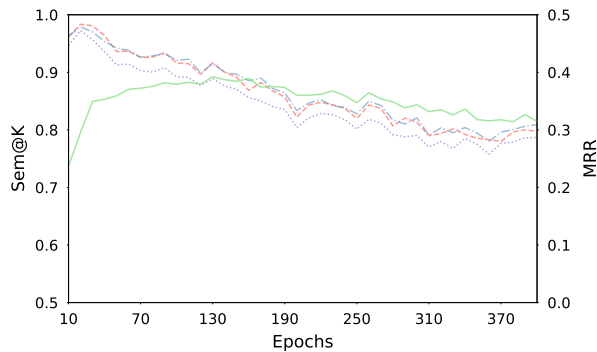
(b) TransH – Codex-S



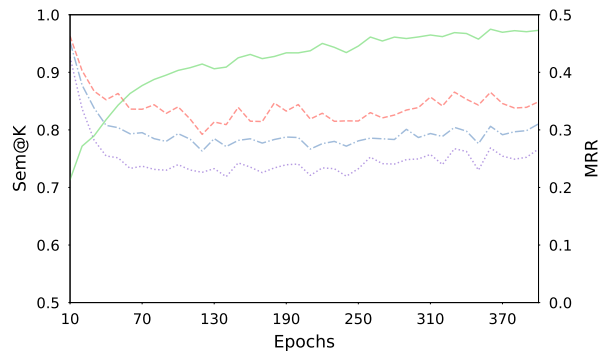
(c) DistMult – Codex-S



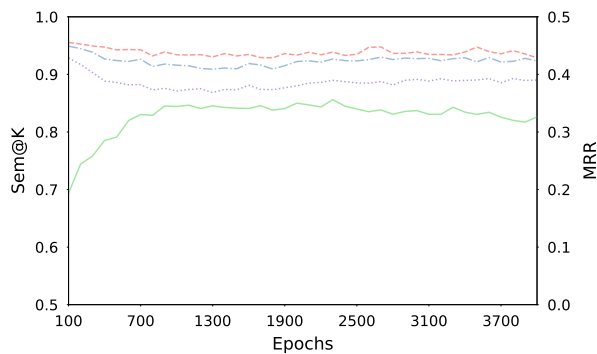
(d) ComplEx – Codex-S



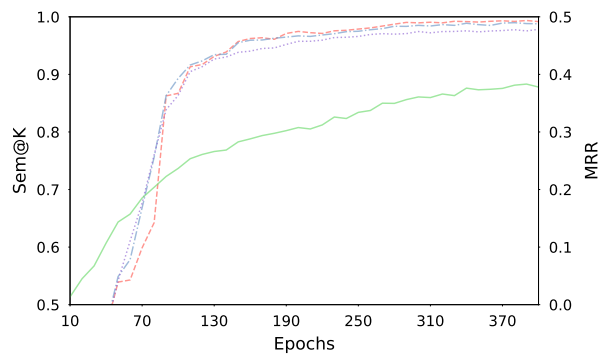
(e) ConvE – Codex-S



(f) ConvKB – Codex-S

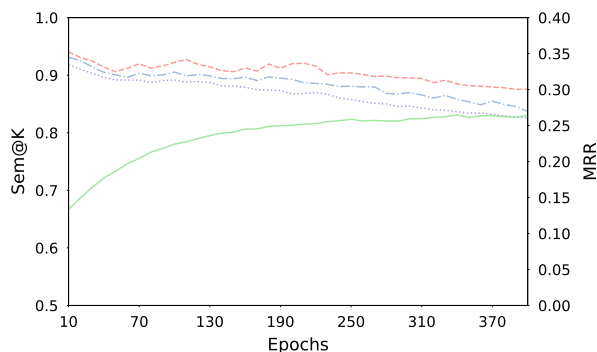


(g) R-GCN – Codex-S

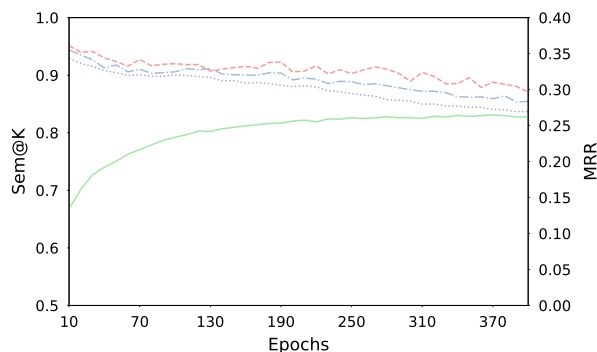


(h) CompGCN – Codex-S

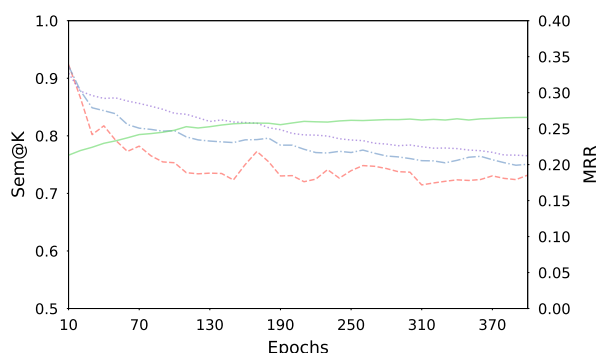
Fig. 12. Evolution of MRR (—), Sem@1 (---), Sem@3 (-.-), and Sem@10 (⋯) on Codex-S



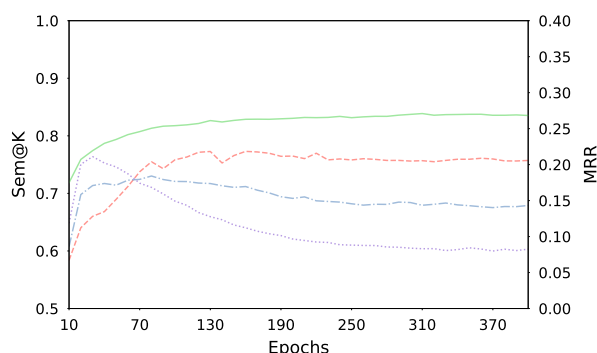
(a) TransE – Codex-M



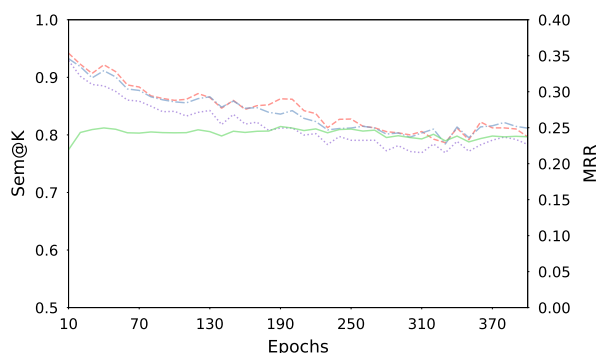
(b) TransH – Codex-M



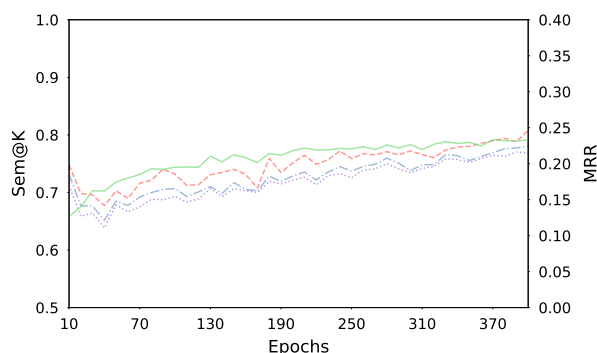
(c) DistMult – Codex-M



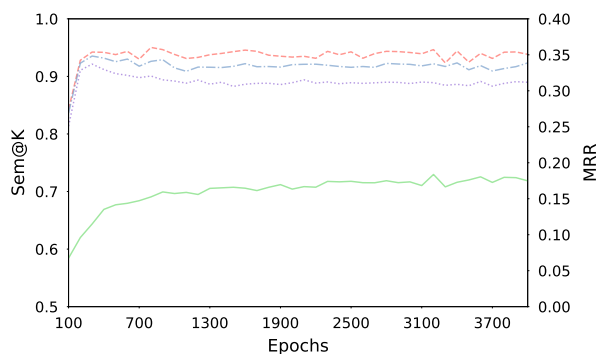
(d) ComplEx – Codex-M



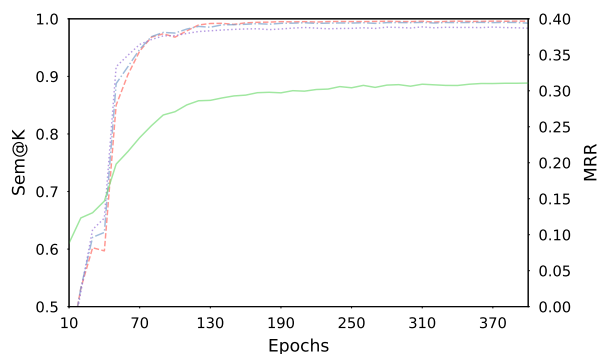
(e) ConvE – Codex-M



(f) ConvKB – Codex-M

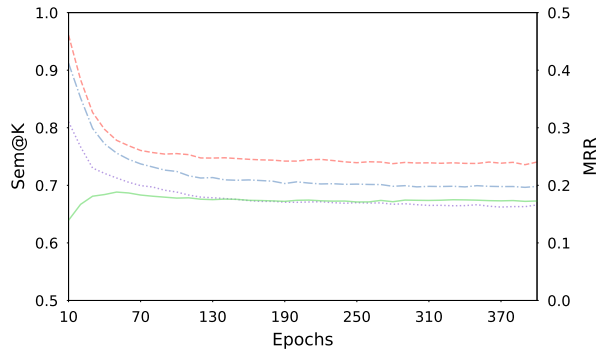


(g) R-GCN – Codex-M

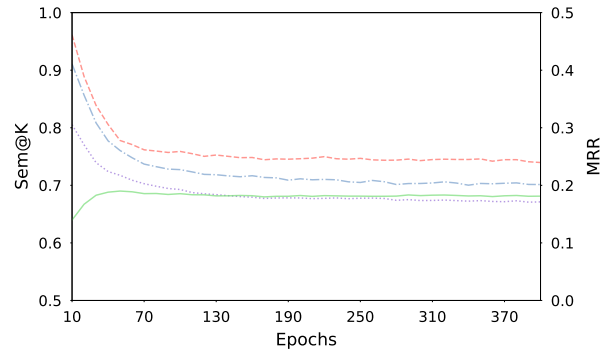


(h) CompGCN – Codex-M

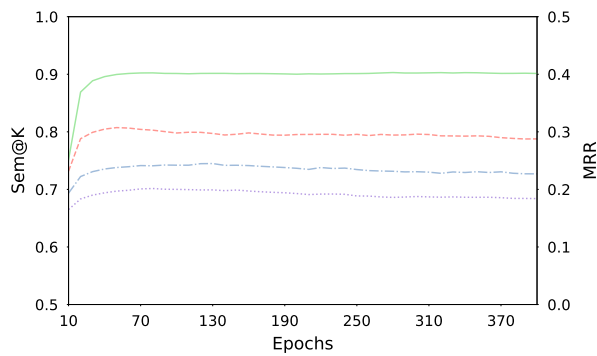
Fig. 13. Evolution of MRR (—), Sem@1 (---), Sem@3 (-.-), and Sem@10 (....) on Codex-M



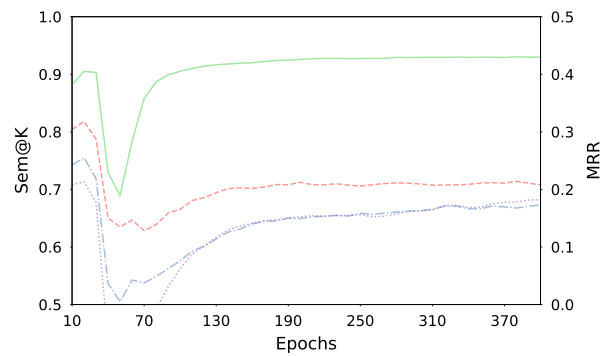
(a) TransE – WN18RR



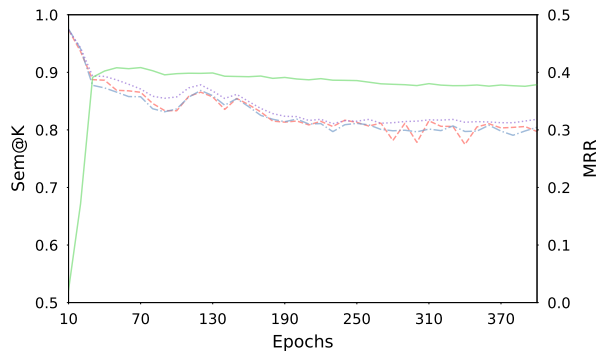
(b) TransH – WN18RR



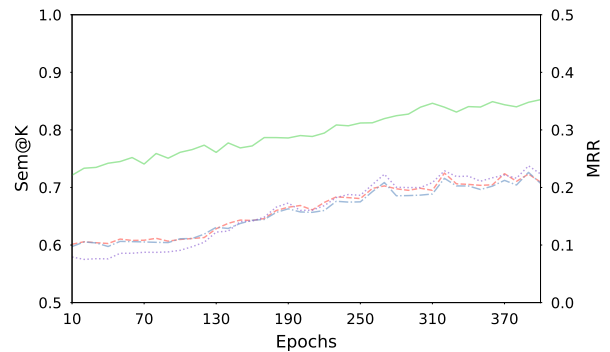
(c) DistMult – WN18RR



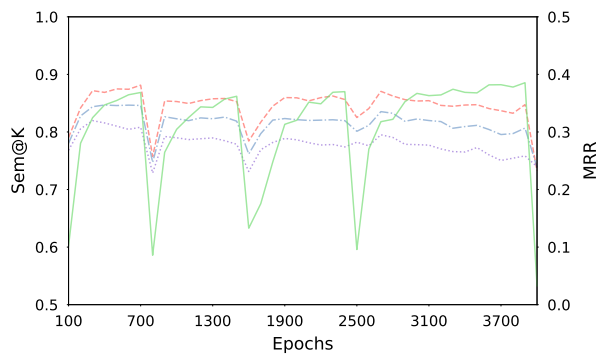
(d) ComplEx – WN18RR



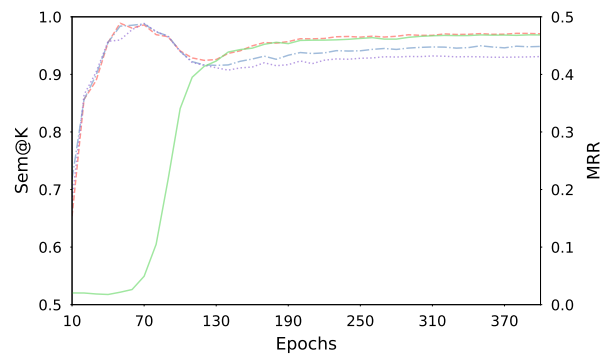
(e) ConvE – WN18RR



(f) ConvKB – WN18RR



(g) R-GCN – WN18RR



(h) CompGCN – WN18RR

Fig. 14. Evolution of MRR (—), Sem@1 (---), Sem@3 (-.-), and Sem@10 (....) on WN18RR



## References

- [1] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic Web* 8(3) (2017), 489–508. doi:10.3233/SW-160218.
- [2] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Trans. Knowl. Data Eng.* 29(12) (2017), 2724–2743.
- [3] A. Bordes, N. Usunier, A. García-Durán, J. Weston and 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 and V. Tresp, Type-Constrained Representation Learning in Knowledge Graphs, in: *The Semantic Web - 14th International Semantic Web Conf. (ISWC)*, Vol. 9366, Springer, 2015, pp. 640–655.
- [5] N. Jain, T. Tran, M.H. Gad-Elrab and D. Stepanova, Improving Knowledge Graph Embeddings with Ontological Reasoning, in: *The Semantic Web - International Semantic Web Conf. ISWC*, Vol. 12922, 2021, pp. 410–426.
- [6] A. Rossi, D. Barbosa, D. Firmani, A. Matinata and P. Merialdo, Knowledge Graph Embedding for Link Prediction: A Comparative Analysis, *ACM Transactions on Knowledge Discovery from Data* 15(2) (2021), 14:1–14:49.
- [7] M. Berrendorf, E. Faerman, L. Vermue and V. Tresp, On the ambiguity of rank-based evaluation of entity alignment or link prediction methods, *arXiv preprint arXiv:2002.06914* (2020).
- [8] C.T. Hoyt, M. Berrendorf, M. Gaklin, V. Tresp and B.M. Gyori, A Unified Framework for Rank-based Evaluation Metrics for Link Prediction in Knowledge Graphs, *arXiv preprint arXiv:2203.07544* (2022).
- [9] S. Tiwari, I. Bansal and 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.
- [10] Y. Wang, D. Ruffinelli, R. Gemulla, S. Broscheit and C. Meilicke, On Evaluating Embedding Models for Knowledge Base Completion, in: *Proc. of the 4th Workshop on Representation Learning for NLP, RepLANLP@ACL*, 2019, pp. 104–112.
- [11] H. Paulheim, Make Embeddings Semantic Again!, in: *Proc. of the ISWC Posters & Demonstrations, Industry and Blue Sky Ideas Tracks*, CEUR Workshop Proceedings, Vol. 2180, 2018.
- [12] N. Jain, J. Kalo, W. Balke and R. Krestel, Do Embeddings Actually Capture Knowledge Graph Semantics?, in: *The Semantic Web - 18th International Conf., ESWC*, LNCS, Vol. 12731, Springer, 2021, pp. 143–159.
- [13] P. Monnin, C. Raïssi, A. Napoli and A. Coulet, Discovering alignment relations with Graph Convolutional Networks: A biomedical case study, *Semantic Web* 13(3) (2022), 379–398.
- [14] N. Hubert, P. Monnin, A. Brun and 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.
- [15] N. Hubert, P. Monnin, A. Brun and D. Monticolo, Knowledge Graph Embeddings for Link Prediction: Beware of Semantics!, in: *DLAKG@ISWC 2022: Workshop on Deep Learning for Knowledge Graphs, held as part of ISWC 2022: the 21st International Semantic Web Conference*, Virtual, China, 2022.
- [16] L.A. Galárraga, C. Teflioudi, K. Hose and F.M. Suchanek, AMIE: association rule mining under incomplete evidence in ontological knowledge bases, in: *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 413–422. doi:10.1145/2488388.2488425.
- [17] L. Galárraga, C. Teflioudi, K. Hose and F.M. Suchanek, Fast rule mining in ontological knowledge bases with AMIE+, *VLDB J.* 24(6) (2015), 707–730. doi:10.1007/s00778-015-0394-1.
- [18] J. Lajus, L. Galárraga and F.M. Suchanek, Fast and Exact Rule Mining with AMIE 3, in: *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, Lecture Notes in Computer Science, Vol. 12123, Springer, 2020, pp. 36–52. doi:10.1007/978-3-030-49461-2\_3.
- [19] C. Meilicke, M.W. Chekol, D. Ruffinelli and H. Stuckenschmidt, Anytime Bottom-Up Rule Learning for Knowledge Graph Completion, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, ijcai.org, 2019, pp. 3137–3143. doi:10.24963/ijcai.2019/435.
- [20] S. Ott, C. Meilicke and M. Samwald, SAFRAN: An interpretable, rule-based link prediction method outperforming embedding models, in: *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021*, 2021. doi:10.24432/C5MK57.
- [21] N. Lao, T.M. Mitchell and W.W. Cohen, Random Walk Inference and Learning in A Large Scale Knowledge Base, in: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, ACL, 2011, pp. 529–539. <https://aclanthology.org/D11-1049/>.
- [22] W. Xiong, T. Hoang and W.Y. Wang, DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, Association for Computational Linguistics, 2017, pp. 564–573. doi:10.18653/v1/d17-1060.
- [23] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola and A. McCallum, Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning, in: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018. <https://openreview.net/forum?id=Syg-YfWCW>.

- [24] M. Wang, L. Qiu and X. Wang, A Survey on Knowledge Graph Embeddings for Link Prediction, *Symmetry* **13**(3) (2021), 485.
- [25] S. Ji, S. Pan, E. Cambria, P. Marttinen and P.S. Yu, A Survey on Knowledge Graphs: Representation, Acquisition, and Applications, *IEEE Trans. Neural Networks Learn. Syst.* **33**(2) (2022), 494–514. doi:10.1109/TNNLS.2021.3070843.
- [26] G. Chowdhury, M. Srilakshmi, M. Chain and S. Sarkar, Neural Factorization for Offer Recommendation using Knowledge Graph Embeddings, in: *Proc. of the SIGIR Workshop on eCommerce*, Vol. 2410, 2019.
- [27] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, in: *Proc. of the Twenty-Eighth AAAI Conf. on Artificial Intelligence*, 2014, pp. 1112–1119.
- [28] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning Entity and Relation Embeddings for Knowledge Graph Completion, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, AAAI Press, 2015, pp. 2181–2187. <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571>.
- [29] G. Ji, S. He, L. Xu, K. Liu and 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, The Association for Computer Linguistics, 2015, pp. 687–696. doi:10.3115/v1/p15-1067.
- [30] Y. Jia, Y. Wang, H. Lin, X. Jin and X. Cheng, Locally Adaptive Translation for Knowledge Graph Embedding, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, AAAI Press, 2016, pp. 992–998. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12018>.
- [31] H. Xiao, M. Huang and X. Zhu, TransG : A Generative Model for Knowledge Graph Embedding, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, The Association for Computer Linguistics, 2016. doi:10.18653/v1/p16-1219.
- [32] Z. Sun, Z. Deng, J. Nie and J. Tang, RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, in: *7th International Conf. on Learning Representations, ICLR*, 2019.
- [33] S. Zhang, Y. Cai, Y. Yao and Q. Liu, Quaternion Knowledge Graph Embeddings, in: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 2731–2741. <https://proceedings.neurips.cc/paper/2019/hash/d961e9f236177d65d21100592edb0769-Abstract.html>.
- [34] Z. Cao, Q. Xu, Z. Yang, X. Cao and Q. Huang, Dual Quaternion Knowledge Graph Embeddings, in: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 6894–6902. <https://ojs.aaai.org/index.php/AAAI/article/view/16850>.
- [35] Z. Zhang, J. Cai, Y. Zhang and J. Wang, Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction, in: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 3065–3072. <https://ojs.aaai.org/index.php/AAAI/article/view/5701>.
- [36] M. Nickel, V. Tresp and 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.
- [37] M. Nickel, K. Murphy, V. Tresp and E. Gabrilovich, A Review of Relational Machine Learning for Knowledge Graphs, *Proc. IEEE* **104**(1) (2016), 11–33. doi:10.1109/JPROC.2015.2483592.
- [38] B. Yang, W. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: *3rd International Conf. on Learning Representations, ICLR*, 2015.
- [39] R. Kadlec, O. Bajgar and 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.
- [40] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex Embeddings for Simple Link Prediction, in: *Proc. of the 33rd International Conf. on Machine Learning, ICML*, Vol. 48, 2016, pp. 2071–2080.
- [41] H. Liu, Y. Wu and Y. Yang, Analogical Inference for Multi-relational Embeddings, in: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, Vol. 70, PMLR, 2017, pp. 2168–2178. <http://proceedings.mlr.press/v70/liu17d.html>.
- [42] S.M. Kazemi and D. Poole, Simple Embedding for Link Prediction in Knowledge Graphs, in: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 4289–4300.
- [43] C. Xu and R. Li, Relation Embedding with Dihedral Group in Knowledge Graph, in: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Association for Computational Linguistics, 2019, pp. 263–272. doi:10.18653/v1/p19-1026.
- [44] M. Nickel, L. Rosasco and T.A. Poggio, Holographic Embeddings of Knowledge Graphs, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, AAAI Press, 2016, pp. 1955–1961. <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484>.
- [45] I. Balazevic, C. Allen and T.M. 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 2019, Hong Kong, China, November 3-7, 2019*, Association for Computational Linguistics, 2019, pp. 5184–5193. doi:10.18653/v1/D19-1522.

- [46] R. Socher, D. Chen, C.D. Manning and A.Y. Ng, Reasoning With Neural Tensor Networks for Knowledge Base Completion, in: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013*, pp. 926–934. <https://proceedings.neurips.cc/paper/2013/hash/b337e84de8752b27eda3a12363109e80-Abstract.html>.
- [47] T. Dettmers, P. Minervini, P. Stenetorp and S. Riedel, Convolutional 2D Knowledge Graph Embeddings, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, AAAI Press, 2018, pp. 1811–1818. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366>.
- [48] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen and D.Q. Phung, A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, Association for Computational Linguistics, 2018, pp. 327–333. doi:10.18653/v1/n18-2053.
- [49] X. Jiang, Q. Wang and 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019, pp. 978–987. doi:10.18653/v1/n19-1103.
- [50] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal and P.P. Talukdar, InteractE: Improving Convolution-Based Knowledge Graph Embeddings by Increasing Feature Interactions, in: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, AAAI Press, 2020, pp. 3009–3016. <https://ojs.aaai.org/index.php/AAAI/article/view/5694>.
- [51] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov and M. Welling, Modeling Relational Data with Graph Convolutional Networks, in: *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, Lecture Notes in Computer Science, Vol. 10843, Springer, 2018, pp. 593–607. doi:10.1007/978-3-319-93417-4\_38.
- [52] I. Ferrari, G. Frisoni, P. Italiani, G. Moro and C. Sartori, Comprehensive Analysis of Knowledge Graph Embedding Techniques Benchmarked on Link Prediction, *Electronics* **11**(23) (2022).
- [53] C. Shang, Y. Tang, J. Huang, J. Bi, X. He and B. Zhou, End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion, in: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, AAAI Press, 2019, pp. 3060–3067. doi:10.1609/aaai.v33i01.33013060.
- [54] D. Nathani, J. Chauhan, C. Sharma and M. Kaul, Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs, in: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Association for Computational Linguistics, 2019, pp. 4710–4723. doi:10.18653/v1/p19-1466.
- [55] S. Vashishth, S. Sanyal, V. Nitin and P.P. Talukdar, Composition-based Multi-Relational Graph Convolutional Networks, in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020. [https://openreview.net/forum?id=BylA\\_C4tPr](https://openreview.net/forum?id=BylA_C4tPr).
- [56] J. Wu, W. Shi, X. Cao, J. Chen, W. Lei, F. Zhang, W. Wu and X. He, DisenKGAT: Knowledge Graph Embedding with Disentangled Graph Attention Network, in: *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, ACM, 2021, pp. 2140–2149. doi:10.1145/3459637.3482424.
- [57] G. Niu, B. Li, Y. Zhang, S. Pu and J. Li, AutoETER: Automated Entity Type Representation with Relation-Aware Attention for Knowledge Graph Embedding, in: *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, Findings of ACL, Vol. EMNLP 2020, Association for Computational Linguistics, 2020, pp. 1172–1181. doi:10.18653/v1/2020.findings-emnlp.105.
- [58] Z. Cui, P. Kapanipathi, K. Talamadupula, T. Gao and Q. Ji, Type-augmented Relation Prediction in Knowledge Graphs, in: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 7151–7159. <https://ojs.aaai.org/index.php/AAAI/article/view/16879>.
- [59] R. Xie, Z. Liu and M. Sun, Representation Learning of Knowledge Graphs with Hierarchical Types, in: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, IJCAI/AAAI Press, 2016, pp. 2965–2971. <http://www.ijcai.org/Abstract/16/421>.
- [60] X. Lv, L. Hou, J. Li and Z. Liu, Differentiating Concepts and Instances for Knowledge Graph Embedding, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Association for Computational Linguistics, 2018, pp. 1971–1979. doi:10.18653/v1/d18-1222.
- [61] P. Wang, J. Zhou, Y. Liu and X. Zhou, Transet: Knowledge graph embedding with entity types, *Electronics* **10**(12) (2021), 1407.
- [62] T.R. Gruber, A Translation Approach to Portable Ontology Specifications, *Knowl. Acquis.* **5**(2) (1993), 199–220. doi:10.1006/knac.1993.1008.
- [63] M. Ali, M. Berrendorf, C.T. Hoyt, L. Vermue, M. Galkin, S. Sharifzadeh, A. Fischer, V. Tresp and J. Lehmann, Bringing Light Into the Dark: A Large-Scale Evaluation of Knowledge Graph Embedding Models Under a Unified Framework, *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12) (2022), 8825–8845. doi:10.1109/TPAMI.2021.3124805.

- [64] C. d'Amato, N.F. Quatraro and N. Fanizzi, Injecting Background Knowledge into Embedding Models for Predictive Tasks on Knowledge Graphs, in: *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, Lecture Notes in Computer Science, Vol. 12731, Springer, 2021, pp. 441–457. doi:10.1007/978-3-030-77385-4\_26.
- [65] R. Rada, H. Mili, E. Bicknell and M. Blettner, Development and application of a metric on semantic nets, *IEEE Trans. Syst. Man Cybern.* **19**(1) (1989), 17–30. doi:10.1109/21.24528.
- [66] Z. Wu and M. Palmer, Verbs Semantics and Lexical Selection, in: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ACL '94, Association for Computational Linguistics, USA, 1994, pp. 133–138-. doi:10.3115/981732.981751.
- [67] C. Leacock and M. Chodorow, *Combining Local Context and WordNet Similarity for Word Sense Identification*, 1998, p. 265-.
- [68] P. Resnik, Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, *J. Artif. Intell. Res.* **11** (1999), 95–130. doi:10.1613/jair.514.
- [69] Y. Li, Z. Bandar and D. McLean, An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources, *IEEE Trans. Knowl. Data Eng.* **15**(4) (2003), 871–882. doi:10.1109/TKDE.2003.1209005.
- [70] K.D. Bollacker, C. Evans, P.K. Paritosh, T. Sturge and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proc. of the ACM SIGMOD International Conf. on Management of Data*, ACM, 2008, pp. 1247–1250.
- [71] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z.G. Ives, DBpedia: A Nucleus for a Web of Open Data, in: *The Semantic Web, 6th International Semantic Web Conf., 2nd Asian Semantic Web Conf., ISWC + ASWC*, Lecture Notes in Computer Science, Vol. 4825, Springer, 2007, pp. 722–735.
- [72] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: a core of semantic knowledge, in: *Proc. of the 16th International Conf. on World Wide Web*, WWW, ACM, 2007, pp. 697–706.
- [73] K. Toutanova and 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.
- [74] B. Ding, Q. Wang, B. Wang and L. Guo, Improving Knowledge Graph Embedding Using Simple Constraints, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, Association for Computational Linguistics, 2018, pp. 110–121. doi:10.18653/v1/P18-1011.
- [75] T.P. Tanon, G. Weikum and F.M. Suchanek, YAGO 4: A Reason-able Knowledge Base, in: *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, Lecture Notes in Computer Science, Vol. 12123, Springer, 2020, pp. 583–596. doi:10.1007/978-3-030-49461-2\_34.
- [76] G.A. Miller, WordNet: A Lexical Database for English, *Commun. ACM* **38**(11) (1995), 39–41.
- [77] T. Safavi and D. Koutra, CoDEX: A Comprehensive Knowledge Graph Completion Benchmark, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Association for Computational Linguistics, 2020, pp. 8328–8350. doi:10.18653/v1/2020.emnlp-main.669.
- [78] Z. Sun, S. Vashishth, S. Sanyal, P.P. Talukdar and Y. Yang, A Re-evaluation of Knowledge Graph Completion Methods, in: *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics ACL*, Association for Computational Linguistics, 2020, pp. 5516–5522.
- [79] H. Wang, Y. Wang, D. Lian and J. Gao, A Lightweight Knowledge Graph Embedding Framework for Efficient Inference and Storage, in: *Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM '21*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1909–1918-. ISBN 9781450384469. doi:10.1145/3459637.3482224.
- [80] J. Portisch, M. Hladik and H. Paulheim, RDF2Vec Light - A Lightweight Approach for Knowledge Graph Embeddings, in: *Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 19th International Semantic Web Conference (ISWC 2020), Globally online, November 1-6, 2020 (UTC)*, CEUR Workshop Proceedings, Vol. 2721, CEUR-WS.org, 2020, pp. 79–84. <http://ceur-ws.org/Vol-2721/paper520.pdf>.
- [81] X. Peng, G. Chen, C. Lin and M. Stevenson, Highly Efficient Knowledge Graph Embedding Learning with Orthogonal Procrustes Analysis, in: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, 2021, pp. 2364–2375. doi:10.18653/v1/2021.naacl-main.187. <https://aclanthology.org/2021.naacl-main.187>.
- [82] W. Zhang, S. Deng, H. Wang, Q. Chen, W. Zhang and H. Chen, *XTransE: Explainable Knowledge Graph Embedding for Link Prediction with Lifestyles in e-Commerce*, 2020, pp. 78–87. ISBN 978-981-15-3411-9. doi:10.1007/978-981-15-3412-6\_8.
- [83] A. Rossi, D. Firmani, P. Merialdo and T. Teofili, Explaining Link Prediction Systems Based on Knowledge Graph Embeddings, in: *Proceedings of the 2022 International Conference on Management of Data, SIGMOD '22*, Association for Computing Machinery, New York, NY, USA, 2022, pp. 2062–2075-. ISBN 9781450392495. doi:10.1145/3514221.3517887.