On Evaluating Embedding Models for Knowledge Base Completion

Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, Samuel Broscheit, Christian Meilicke
Data and Web Science Group
University of Mannheim, Germany
{ywang,rgemulla}@uni-mannheim.de,
{daniel,broscheit,christian}@informatik.uni-mannheim.de,

Abstract

Knowledge graph embedding models have recently received significant attention in the literature. These models learn latent semantic representations for the entities and relations in a given knowledge base; the representations can be used to infer missing knowledge. In this paper, we study the question of how well recent embedding models perform for the task of knowledge base completion, i.e., the task of inferring new facts from an incomplete knowledge base. We argue that the entity ranking protocol, which is currently used to evaluate knowledge graph embedding models, is not suitable to answer this question since only a subset of the model predictions are evaluated. We propose an alternative entity-pair ranking protocol that considers all model predictions as a whole and is thus more suitable to the task. We conducted an experimental study on standard datasets and found that the performance of popular embeddings models was unsatisfactory under the new protocol, even on datasets that are generally considered to be too easy. Moreover, we found that a simple rule-based model often provided superior performance. Our findings suggest that there is a need for more research into embedding models as well as their training strategies for the task of knowledge base completion.

1 Introduction

A knowledge base (KB) is a collection of relational facts, often represented as *(subject, relation, object)*-triples. KBs provide rich information for NLP tasks such as question answering (Abujabal et al., 2017) or entity linking (Shen et al., 2015). Since knowledge bases are inherently incomplete (West et al., 2014), there has been considerable interest into methods that infer missing knowledge.

In particular, a large number of *knowledge graph embedding (KGE) models* have been proposed in the recent literature (Nickel et al., 2016a). These models embed the entities and relations of a given knowledge base into a low-dimensional latent space such that the structure of the knowledge base is captured. The embeddings are subsequently used to assess whether unobserved triples constitute missing facts or are likely to be false.

To evaluate the performance of a KGE model, the most commonly adopted protocol is the *entity ranking* (ER) protocol.[1] The protocol takes as input a set of previously unobserved *test triples*, such as *(Einstein, bornIn, Ulm)*, and uses the embedding model to rank all possible answers to the questions *(?, bornIn, Ulm)* and *(Einstein, bornIn, ?)*. Model performance is then assessed based on the rank of the answer present in the test triple (*Einstein* and *Ulm*, resp.). Since each question is constructed from a test triple, the protocol ensures that questions are meaningful and always have a correct answer. Throughout this paper, we refer to the task of answering such questions as *question answering* (QA). The ER protocol is, in principle, well-suited to evaluate performance of KGE models for QA, although concerns about the benchmark datasets (Toutanova and Chen, 2015), the considered models (Kadlec et al., 2017) and the evaluation (Joulin et al., 2017) have been raised.

In this paper, we aim to study the performance of popular embedding models for the task of knowledge base completion (KBC): given a relation of a knowledge base (*bornIn*), infer true missing facts (*(Einstein, bornIn, Ulm)*). This task is different from QA (as defined above) since no information about potential missing triples is provided upfront. We argue that the ER protocol is not well-suited to assess model performance for KBC. To see this, observe that models that assign high confidence scores to incorrect triples such as

[1] We discuss other less adopted evaluation methods in Sec. 3.2.

Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), pages 104–112
Florence, Italy, August 2, 2019.

(Ulm, bornIn, Einstein) are not penalized by the ER protocol because the corresponding questions (e.g., *(Ulm, bornIn, ?)*) are never asked. Thus a model that performs well on ER may still not perform well for KBC. In fact, we argue here that some commonly used KGE models are inherently not well-suited to KBC.

We propose a simple *entity-pair ranking* (PR) protocol (PR), which is more suitable to assess model performance for KBC. Given a relation such as *bornIn*, PR uses the KGE model to rank all possible answers—i.e., all entity pairs—to the question $(?, \textit{bornIn}, ?)$, and subsequently assesses model performance based on the rank of the test triples for relation *bornIn* in the answer. The protocol ensures that a model's performance is negatively affected if the model assigns high scores to false triples such as *(Ulm, bornIn, Einstein)*.

We conducted an experimental study on commonly used benchmark datasets under the ER and the PR protocols. We found that the performance of popular embeddings models was often good under the ER but unsatisfactory under the PR protocol, even on "simple" datasets that are generally considered to be too easy. Moreover, we found that a simple rule-based model often provided superior performance for PR. Our findings suggests that there is a need for more research into embedding models as well as their training strategies for the task of knowledge base completion.

2 Preliminaries

Given a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$, a knowledge base $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of triples (i, k, j), where $i, j \in \mathcal{E}$ and $k \in \mathcal{R}$. We refer to i, k and j as the *subject*, *relation*, and *object* to the triple, respectively.

Embedding models. A KGE model associates an *embedding* $\boldsymbol{e}_i \in \mathbb{R}^{d_e}$ and $\boldsymbol{r}_k \in \mathbb{R}^{d_r}$ with each entity i and relation k, resp. We refer to d_e and $d_r \in \mathbb{N}^+$ as the *size* of the embeddings. Each KGE model uses a *scoring function* $s : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \mathbb{R}$ to associate a score $s(i, k, j)$ to each triple $(i, k, j) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. The scores induce a ranking: triples with high scores are considered more likely to be true than triples with low scores. Roughly speaking, the models try to find embeddings that capture the structure of the entire knowledge graph well. In this work, we consider a popular family of embedding models called bilinear models.

Bilinear KGE models. Bilinear models use the relation embedding $\boldsymbol{r}_k \in \mathbb{R}^{d_r}$ to construct a *mixing matrix* $\boldsymbol{R}_k \in \mathbb{R}^{d_e \times d_e}$, and they use scoring function $s(i, k, j) = \boldsymbol{e}_i^T \boldsymbol{R}_k \boldsymbol{e}_j$. The models differ mainly in how $\boldsymbol{R}_k$ is constructed. Unless stated otherwise, the models use the same embedding sizes for entities and relations (i.e., $d_r = d_e$).

RESCAL (Nickel et al., 2011) is the most general bilinear model: it sets $d_r = d_e^2$ and stores in $\boldsymbol{r}_k$ the values of each entry of $\boldsymbol{R}_k$. Analogy (Liu et al., 2017) constrains $\boldsymbol{R}_k \in \mathbb{R}^{d_e \times d_e}$ to a block diagonal matrix in which each block is either (i) a real scalar or (ii) a 2×2 matrix of form $\begin{pmatrix} x & -y \\ y & x \end{pmatrix}$ with $x, y \in \mathbb{R}$. DistMult (Carroll and Chang, 1970; Yang et al., 2014) is a symmetric factorization model with $\boldsymbol{R}_k = \operatorname{diag}(\boldsymbol{r}_k)$ or, equivalently, considers only case (i) of Analogy. ComplEx (Trouillon et al., 2016) and HolE (Nickel et al., 2016b) are equivalent to a model that restricts $\boldsymbol{R}_k$ to case (ii). TransE (Bordes et al., 2013) is a translation-based model with scoring function $s(i, k, j) = -\|\boldsymbol{e}_i + \boldsymbol{r}_k - \boldsymbol{e}_j\|_2$ (or $\|\cdot\|_1$); it can also be written in bilinear form (Wang et al., 2018).

Rule learning. Rule learning methods derive logical rules that encode dependencies found in the KBs (Galárraga et al., 2013). We consider a simple rule-based model called RuleN (Meilicke et al., 2018) as baseline. The model learns (weighted) implication rules of form

$$\begin{aligned} r(i,j) &\leftarrow r_1(i,z_1) \wedge \cdots \wedge r_n(z_n,j) \\ r(i,c) &\leftarrow \exists z.r(i,z) \end{aligned}$$

where r_i are relations, c is a constant entity, and i, j, and z_i are variables quantified over entities. To perform KBC, rule-based models query the KB for instances of the bodies of each rule and interpret the corresponding head as (weighted) predicted fact.

3 Evaluation Protocols

We first review two widely used evaluation protocols for QA. We then argue that these protocols are not well-suited for assessing KBC performance, because they focus on a small subset of all possible facts for a given relation. We then introduce the entity-pair ranking (PR) protocol and discuss its advantages and potential shortcomings.

3.1 Current Evaluation Protocols

The triple classification (TC) or the entity ranking (ER) protocols are commonly used to assess KGE model performance, where ER is arguably the most widely adopted protocol. We assume throughout that only true but no false triples are available (as is commonly the case), and that the available true triples are divided into training, validation, and test triples.

Triple classification (TC) The goal of triple classification is to test the model's ability to discriminate between true and false triples (Socher et al., 2013). Since only true triples are available in practice, pseudo-negative triples are generated by randomly replacing either the subject or the object of each test triple by a random entity (that appears as a subject or object in the considered relation). All triples are then classified as positive or negative according to the KGE scores. In particular, triple (i, k, j) is classified as positive if its score $s(i, k, j)$ exceeds a relation-specific decision threshold τ_k (learned on validation data using the same procedure). Model performance is assessed by classification accuracy.

Entity ranking (ER) ER assesses model performance by testing its ability to perform QA (as defined before). In particular, for each test triple $t = (i, k, j)$, two questions $q_s = (?, k, j)$ and $q_o = (i, k, ?)$ are generated. For question q_s, all entities $i' \in \mathcal{E}$ are ranked based on the score $s(i', k, j)$. To avoid misleading results, entities $i' \neq i$ that correspond to observed triples in the dataset—i.e., (i', k, j) occurs in the training/validation/test triples—are discarded to obtain a *filtered ranking*. The same process is applied for question q_o. Model performance is evaluated based on the recorded positions of the test triples in the filtered ranking. Models that tend to rank test triples (known to be true) higher than unknown triples (assumed to be false) are thus considered superior. Usually, the micro-average of *filtered Hits@K*—i.e., the proportion of test triples ranking in the top-K—and *filtered MRR*—i.e., the mean reciprocal rank of the test triples—are used to assess model performance.

3.2 Discussion

Wang et al. (2018) found that most models achieve a TC accuracy of at least 93% on a benchmark dataset. This is because each test triple is compared against a single negative triple, and due to the high number of possible negative triples, it is unlikely that the chosen triple has a high predicted score, rendering most classification tasks "easy". Consequently, the accuracy of triple classification overestimates model performance. This protocol is less adopted in recent work.

We argue that ER is appropriate to evaluate QA performance, but may overestimate model performance for KBC. Since ER generates questions from true test triples, it only asks questions that are *known to have a correct answer*. The question itself thus provides useful information. This perfectly matches QA, but it does not match KBC where such information is not available.

To better illustrate why ER can lead to misleading assessment of a model's KBC performance, consider the DistMult model and the asymmetric relation *nominatedFor*. As described in Sec. 2, DistMult models all relations as symmetric in that $s(i, k, j) = s(j, k, i)$. Now consider triple $t =$ *(H. Simon, nominatedFor, Nobel Prize)*, and let us suppose that the model correctly assigns t a high score $s(t)$. Then the inverse triple $t' =$ *(Nobel Prize, nominatedFor, H. Simon)* will also obtain a high score since $s(t') = s(t)$. Thus the score produced by DistMult does not discriminate between the true triple t and the false triple t'. In ER, however, questions about t' are never asked; there is no test triple for this relation containing either *Nobel Prize* as subject or *H. Simon* as object. The symmetry of DistMult's prediction thus barely affects its performance under the ER protocol.

For another example, consider TransE and the relation $k =$ *marriedTo*, which is symmetric but not reflexive. One can show that for all (i, k, j), the TransE scores satisfy

$$\begin{aligned} &s(i, k, j) + s(j, k, i) \\ &\quad = -\|\boldsymbol{e}_i + \boldsymbol{r}_k - \boldsymbol{e}_j\| - \|\boldsymbol{e}_j + \boldsymbol{r}_k - \boldsymbol{e}_i\| \\ &\quad \leq -\|\boldsymbol{e}_i + \mathbf{0} - \boldsymbol{e}_j\| - \|\boldsymbol{e}_j + \mathbf{0} - \boldsymbol{e}_i\|. \end{aligned}$$

For symmetric relations, TransE should aim to assign high scores to both (i, k, j) and (j, k, i). To do so, TransE has the tendency to push the relation embedding $\boldsymbol{r}_k$ towards $\mathbf{0}$ as well as $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ towards each other. But when $\boldsymbol{r}_k \approx \mathbf{0}$, then $s(i, k, i)$ is high for all i, so that the relation is treated as if it were reflexive. Again, in ER, this property only slightly influences the results: there is only one "reflexive" tuple in each filtered entity list so that

the correct answer i for question $(?, k, j)$ ranks at most one position lower.

More expressive models such as RESCAL or ComplEx do not have such inherent limitations. Nevertheless, our experimental study shows that these models (at least in the way are currently trained) also tend to assign high scores to false triples.

3.3 Entity-Pair Ranking Protocol

We propose a simple alternative protocol called entity-pair ranking (PR). The protocol is more suitable to assess a model's KBC performance (although it is not without flaws either; see below). PR proceeds as follows: for each relation k, we use the KGE model to rank all triples for a specified relation k, i.e., to rank all answers to question $(?, k, ?)$. As in ER, we filter out all triples that occur in the training and validation data to obtain a filtered ranking, i.e., to only rank triples that were not used during model training. If a model tends to assign a high score to negative triples, its performance is likely to be negatively affected because it becomes harder for true triples to rank high.

Note that the number of candidate answers considered by PR is much larger than those considered by ER. Consider a relation k and let $\mathcal{T}_k$ be the set of test triples for relation k. Then ER considers $2|\mathcal{T}_k|\,|\mathcal{E}|$ candidates in total during evaluation, while PR considers $|\mathcal{E}|^2$ candidates. Moreover, PR considers all test triples in $\mathcal{T}_k$ simultaneously instead of sequentially. For this reason, we cannot use the MRR metric commonly used in ER. Instead, we assess model performance using weighted *MAP@K*—i.e., the weighted mean average precision in the top-K filtered results—and weighted *Hits@K*—i.e., the weighted percentage of test triples in the top-K filtered results. We weight the influence of relation k proportionally to its number of test triples (capped at K), thereby closely following ER:

$$\text{MAP@}K = \sum_{k \in \mathcal{R}} \text{AP}_k@K \times \frac{\min(K, |\mathcal{T}_k|)}{\sum_{k' \in \mathcal{R}} \min(K, |\mathcal{T}_{k'}|)}$$

$$\text{Hits@}K = \sum_{k \in \mathcal{R}} \text{Hits}_k@K \times \frac{\min(K, |\mathcal{T}_k|)}{\sum_{k' \in \mathcal{R}} \min(K, |\mathcal{T}_{k'}|)}.$$

Here $\text{AP}_k@K$ is the average precision of the top-K list (w.r.t. test triples $\mathcal{T}_k$) and $\text{Hits}_k@K$ refers to the fraction of test triples in the top-K list. Note that K should be chosen much larger for PR than for ER since it roughly corresponds to the number of triples we aim to predict for relation k.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}^{train}\rvert$	$\lvert\mathcal{T}^{val}\rvert$	$\lvert\mathcal{T}^{test}\rvert$
FB15K	14 951	1 345	483 142	50 000	59 071
FB-237	14 505	237	272 115	17 535	20 466
WN18	40 943	18	141 442	5 000	5 000
WNRR	40 559	11	86 835	2 824	2 924

Table 1: Dataset statistics

The PR protocol is more suited to evaluate KBC performance because it considers all model predictions. The protocol also has some disadvantages, however. First, as ER, the PR protocol may underestimate model performance due to unobserved true triples ranked high by the model. Since a larger number of candidates is considered, PR may be more sensitive to this problem than ER. We explore the effect of underestimation in our empirical study in Sec. 4.4. Another concern with PR is its potentially high computational cost. For current benchmark datasets, we found that the PR evaluation is feasible. Generally, one may argue that an embedding model is suitable for KBC only if it is feasible to determine high-scoring triples in a sufficiently efficient way. Since PR only requires the computation of the top-K predictions, performance can potentially be improved using techniques such as maximum inner-product search Shrivastava and Li (2014).

4 Experimental Study

We conducted an experimental study to assess the performance of various bilinear embedding models for KBC.[2] All datasets, experimental results, and source code are publicly available.[3] For all models, we performed evaluation under both the ER and PR protocols in order to assess their performance for the QA and KBC tasks, respectively. We found that many KGE models provided good ER but low PR performance. We also considered a simple rule-based system called *RuleN* (Meilicke et al., 2018), which provided good performance under the ER protocol, and found that RuleN performed better in both ER and PR. Our results imply that more research into KGE models for KBC is needed.

We also investigated the extent to which PR

[2] Some other KGE models do not support KBC directly due to their architecture; e.g., ConvE (Dettmers et al., 2018).

[3] http://www.uni-mannheim.de/dws/research/resources/kge-eval/

underestimates model performance due to unobserved true triples. We found that underestimation is not the main reason for the low PR performance of many KGE models; in fact, many KGE models ranked high clearly wrong tuples (e.g., with incorrect types).

4.1 Experimental Setup

Datasets. We used the four common KBC benchmark datasets: FB15K, WN18, FB-237, and WNRR. The latter two datasets were created since the former two datasets were considered too easy for embedding models (based on ER). Key dataset statistics are summarized in Table 1.

Negative sampling. We trained the embedding models using negative sampling to obtain pseudo-negative triples. We consider three sampling strategies in our experiments:
Perturb 1: For each training triple $t = (i, k, j)$, sample pseudo-negative triples by randomly replacing either i or j with a random entity (but such that the resulting triple is unobserved). This strategy matches ER, which is based on questions $(?, k, j)$ and $(i, k, ?)$.
Perturb 1-R: For each training triple $t = (i, k, j)$, sample pseudo-negative triples by randomly replacing either i, k or j. The generated negative samples are not compared with the training set (Liu et al., 2017).
Perturb 2: For each training triple $t = (i, k, j)$, obtain pseudo-negative triples by randomly sampling unobserved tuples for relation k. This method appears more suited to PR.

Training and implementation. We trained DistMult, ComplEx, Analogy and RESCAL with AdaGrad (Duchi et al., 2011) using binary cross-entropy loss. We used pair-wise ranking loss for TransE (as it always produces negative scores). All embeddding models are implemented on top of the code of Liu et al. (2017)[4] in C++ using OpenMP. For RuleN, we use the original implementation provided by the authors. The evaluation protocols were written in Python, with Bottleneck[5] used for efficiently obtaining the top-K entries for PR. We found PR (which took ≈30–90 minutes) was about 3–4 times slower than ER .

Hyperparameters. The best hyperparameters were selected based on MRR (for ER) and MAP@100 (for PR) on the validation data. For both protocols, we performed an exhaustive grid search over the following hyperparameter settings: $d_e \in \{100, 150, 200\}$, weight of l_2-regularization $\lambda \in \{0.1, 0.01, 0.001\}$, learning rate $\eta \in \{0.01, 0.1\}$, negative sampling strategies *Perturb 1*, *Perturb 2* and *Perturb 1-R*,[6] and margin hyperparameter $\gamma \in \{0.5, 1, 2, 3, 4\}$ for TransE. For each training triple, we sampled 6 pseudo-negative triples. To keep effort tractable, we only used the most frequent relations from each dataset for hyperparameter tuning (top-5, top-5, top-15, and top-30 most frequent relations for WN18, WNRR, FB-237 and FB-15k, respectively). We trained each model for up to 500 epochs during grid search. In all cases, we evaluated model performance every 50 epochs and used the overall best-performing model. For RuleN, we used the best settings reported by the authors for ER (Meilicke et al., 2018). For PR, we learned path rules of length 2 using a sampling size of 500 for FB15K and FB-237. For WN18 and WNRR, we learned path rules of length 3 and sampling size of 500.

4.2 Performance Results with ER

Table 2 summarizes the ER results. Embedding models perform competitively with respect to RuleN on all datasets, except for their MRR performance on FB15K. Notice that this generally holds even for the more restricted models (TransE and DistMult) on the more challenging datasets, which were created after criticizing FB15K and WN18 as too easy (Toutanova and Chen, 2015; Dettmers et al., 2018). In particular, although DistMult can only model symmetric relations, and although most relations in these datasets are asymmetric, DistMult has good ER performance. Likewise, TransE achieved great performance in Hits@10 on all datasets, including WN18 which contains a large number of symmetric relations, which are not easily modeled by TransE.

4.3 Performance Results with PR

The evaluation results of PR with $K = 100$ are summarized in Table 3. Note that Tables 2 and 3 are not directly comparable: they measure different tasks. Also note that we use a different value of K, which in PR corresponds to the number of predicted facts per relation. We discuss the effect of the choice of K later.

[4] https://github.com/quark0/ANALOGY
[5] https://pypi.org/project/Bottleneck/
[6] We found that *Perturb-2* can be useful in both protocols.

Dataset	FB15K		FB-237		WN18		WNRR	
Model	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
DistMult	0.660	0.845	0.270	0.432	0.790	0.937	0.432	0.474
TransE	0.500	0.777	0.290	0.466	0.720	0.908	0.220	0.491
ComplEx	0.700	0.835	0.280	0.435	0.940	0.948	0.440	0.481
Analogy	0.700	0.836	0.270	0.433	0.941	0.942	0.440	0.486
RESCAL	0.464	0.699	0.270	0.427	0.920	0.939	0.420	0.447
RuleN	0.805	0.870	0.260	0.420	0.950	0.958	—	0.536

Table 2: Results with the entity ranking protocol (ER), which assesses QA performance

Dataset	FB15K		FB-237		WN18		WNRR	
Model	MAP@100	Hits@100	MAP@100	Hits@100	MAP@100	Hits@100	MAP@100	Hits@100
DistMult	0.013	0.104	0.030	0.042	0.079	0.097	0.141	0.178
TransE	0.211	0.363	0.079	0.176	0.223	0.315	0.020	0.013
ComplEx	0.311	0.486	0.071	0.166	0.825	0.904	0.168	0.200
Analogy	0.188	0.348	0.049	0.143	0.776	0.874	0.154	0.198
RESCAL	0.150	0.303	0.067	0.150	0.482	0.609	0.131	0.138
RuleN	0.774	0.837	0.076	0.158	0.948	0.968	0.215	0.251

Table 3: Results with the entity-pair ranking protocol (PR), which assesses KBC performance

For the embeddings, observe that with the exception of Analogy and ComplEx on WN18, the performance of all models is unsatisfactory on all datasets, especially when compared with RuleN on FB15K and WN18, which were previously considered to be too easy for embedding models. Specifically, DistMult's Hits@100 is slightly less than 10% on WN18, meaning that if we add the top 100 ranked triples to the KB, over 90% of what is added is likely false. Even when using ComplEx, the best model on FB15K, we would potentially add more than 50% false triples. This implies that embedding models cannot capture simple rules successfully. The notable exceptions are ComplEx and Analogy on WN18, although both are still behind RuleN. TransE and DistMult did not achieve competitive results on WN18. In addition, DistMult did not achieve competitive results on FB15K and FB-237 and TransE did not achieve competitive results in WNRR. In general, ComplEx and Analogy performed consistently better than other models across different datasets. When compared with the RuleN baseline, however, the performance of these models was often not satisfactory. This suggests that better KGE models and/or training strategies are needed for KBC.

RuleN did not perform well on FB-237 and WNRR, likely because the way these datasets were constructed makes them intrinsically difficult for rule-based methods (Meilicke et al., 2018). This is reflected in both ER and PR results.

To better understand the change in performance of TransE and DistMult, we investigated their predictions for the top-5 most frequent relations on WN18. Table 4 shows the number of test triples appearing in the top-100 for each relation (after filtering triples from the training and validation sets). The numbers in parentheses are discussed in Section 4.4.

We found that DistMult worked well on the symmetric relation *derivationally related form*, where its symmetry assumption clearly helps. Here 93% of the training data consists of symmetric pairs (i.e., (i, k, j) and (j, k, i)), and 88% of the test triples have its symmetric counterpart in the training set. In contrast, TransE contained no test triples for *derivationally related form* in the top-100 list. We found that the norm of the embedding vector of this relation was 0.1, which was considerably smaller than for the other relations (avg. 1.4). This supports our argument that TransE tends to push symmetric relation embeddings to $\mathbf{0}$.

Note that while *hyponymy*, *hypernymy*, *member meronym* and *member holonym* are semantically transitive, the dataset contains almost exclusively their transitive core, i.e., the dataset (both train and test) does not contain many of the transitive links of the relations. As a result, models that cannot handle transitivity well may still produce good results. This might explain why TransE performed better for these relations than for *derivationally related form*. DistMult did not perform well on

Relation	Model											
	DistMult		TransE		ComplEx		Analogy		RESCAL		RuleN	
hyponymy	1	(1)	18	(32)	99	(99)	99	(99)	92	(93)	100	(100)
hypernymy	0	(0)	5	(33)	99	(99)	99	(99)	96	(98)	100	(100)
derivationally related form	100	(100)	0	(0)	100	(100)	100	(100)	6	(68)	100	(100)
member meronym	0	(0)	18	(41)	74	(84)	83	(85)	44	(63)	100	(100)
member holonym	0	(0)	16	(47)	74	(83)	83	(85)	37	(54)	100	(100)

Table 4: Number of test triples in the top-100 filtered predictions on WN18. An estimate of the number of true triples in the top-100 list is given in parentheses.

these relations (they are asymmetric). ComplEx and Analogy showed superior performance across all relations. RESCAL is in between, most likely due to difficulties in finding a good parameterization. However, it is unclear to us why TransE performed well on FB15K and FB-237.

To investigate model performance in PR for different values of K, we give the curves of Hits@K as a function of K for all datasets in Fig. 1. ComplEx and Analogy, which are universal models, performed best for large K w.r.t. other embedding models. Similarly, TransE works the best for small values of K on FB15K and FB-237. Notice that RuleN performs considerably better on FB15K, WN18 and WNRR, while it still performs competitively on FB-237.

4.4 Influence of Unobserved True Triples

Since all datasets are based on incomplete knowledge bases, all evaluation protocols may systematically underestimate model performance. In particular, any true triple t that is neither in the training, nor validation, nor test data is treated as negative during ranking-based evaluations. A model which correctly ranks t high is thus penalized. PR might be particularly sensitive to this due to the large number of candidates considered.

It is generally unclear how to design an automatic evaluation strategy that avoids this problem. Manual labeling can be used to address this, but it may sometimes be infeasible given the large number of relations, entities, and models for KBC.

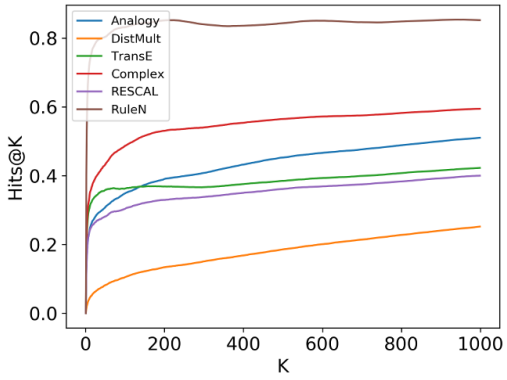
To explore such underestimation effect in PR, we decoded the unobserved triples in the top-100 predictions of the 5 most frequent relations of WN18. We then checked whether those triples are implied by the symmetry and transitivity properties of each relation. In Table 4, we give the resulting number of triples in parentheses (i.e., number of test triples + implied triples). We observed that underestimation indeed happened. TransE was mostly affected, but still did not lead to competitive results when compared to ComplEx and Analogy. RuleN achieves the best possible results in all 5 relations. These results suggest that (1) underestimation is indeed a concern, and (2) the results in PR can nevertheless give an indication of relative model performance.

4.5 Type Filtering

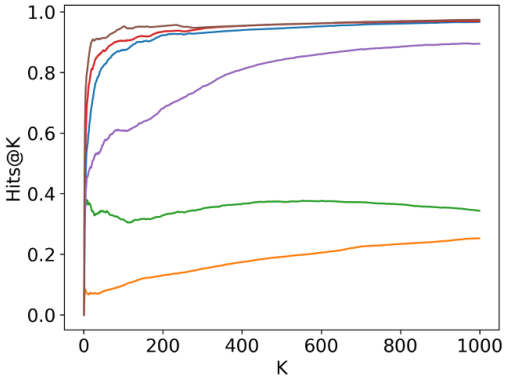
When background knowledge (BK) is available, embedding models only need to score triples consistent with the BK. We explored whether their performance can be improved by filtering out type-inconsistent triples from each model's predictions. Notice that this is inherently what rule-based approaches do, since all predicted candidates will be type-consistent. In particular, we investigated how model performance is affected when we filter out predictions that violate type constraints (domain and range of each relation). If a model's performance improves with such type filtering, it must have ranked tuples with incorrect types high in the first place. We can thus assess to what extent models capture entity types as well as the domain and range of the relations.

We extracted from Freebase type definitions for entities and domain and range constraints for relations. We also added the domain (or range) of a relation k to the type set of each subject (or object) entity which appeared in k. We obtained types for all entities in both FB datasets, and domain/range specifications for roughly 93% of relations in FB15K and 97% of relations in FB-237. The remaining relations were evaluated as before.

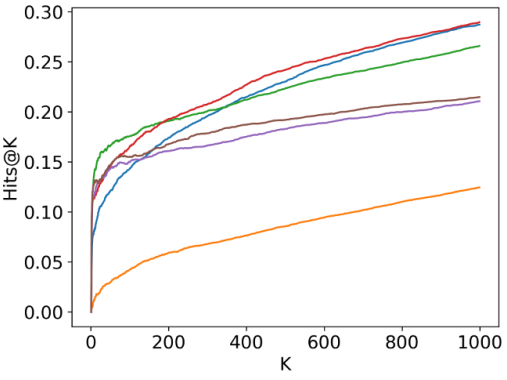
We report in Table 5 the Hits@100 and MAP@100 as well as their absolute improvement (in parentheses) w.r.t. Table 3. We also include the results of RuleN from Table 3, which are already type-consistent. The results show that all KGE models improve by type filtering; thus all models do predict triples with incorrect types. In particular, DistMult shows considerable improvement on both datasets. Indeed, about 90% of the



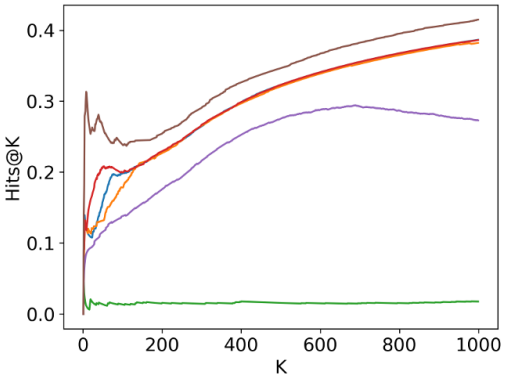
(a) FB15K



(b) WN18



(c) FB-237



(d) WN18RR

Figure 1: Hits@K with PR as a function of K

Data	Model	MAP@K (%)		Hits@K (%)	
FB15K	DistMult	18.8	(+17.5)	36.4	(+26.0)
	TransE	25.7	(+4.5)	41.7	(+5.4)
	ComplEx	53.1	(+22.0)	69.6	(+21.0)
	Analogy	41.3	(+22.5)	61.5	(+26.7)
	RESCAL	16.7	(+1.7)	32.8	(+2.5)
	RuleN	77.4	(0.0)	83.7	(0.0)
FB-237	DistMult	9.5	(+9.2)	18.1	(+13.9)
	TransE	11.3	(+3.4)	21.2	(+3.6)
	ComplEx	11.3	(+4.2)	21.8	(+5.2)
	Analogy	10.5	(+5.6)	20.9	(+6.6)
	RESCAL	10.2	(+3.5)	19.0	(+4.0)
	RuleN	7.6	(0.0)	15.8	(0.0)

Table 5: Results with PR using type filtering (K = 100).

relations in FB15K (about 85% for FB-237) have a different type for their domain and range. As DistMult treats all relations as symmetric, it introduces a wrong triple for each true triple into the top-K list on these relations; type filtering allows us to ignore these wrong tuples. This is also consistent with DistMult's improved performance under ER, where type constraints are implicitly used since only questions with correct types are considered. Interestingly, ComplEx and Analogy improved considerably on FB15K, suggesting that the best performing embedding models on this dataset are still making a considerable number of type-inconsistent predictions. On FB15K, the relative ranking of the models with type filtering is roughly equal to the one without type filtering. On the harder FB-237 dataset, all models now perform similarly. Notice that when compared with RuleN, embedding models are still behind on FB15K, but are no longer behind on FB-237.

5 Conclusion

We investigated whether current embedding models provide good results for knowledge base completion, i.e., the task or inferring new facts from an incomplete knowledge base. We argued that the commonly-used ER evaluation protocol is not suited to answer this question, and proposed the PR evaluation protocol as an alternative. We evaluated a number of popular KGE models under the ER and PR protocols and found that most KGE models obtained good results under the ER but not the PR protocol. Therefore, more research into embedding models and their training is needed to assess whether, when, and how KGE models can be exploited for knowledge base completion.

References

Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NIPS)*.

J. Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3).

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web (WWW)*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Maximilian Nickel, and Tomas Mikolov. 2017. Fast linear model for knowledge graph embeddings. *CoRR*, abs/1710.10881.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. In *Workshop on Representation Learning for NLP (RepL4NLP@ACL)*.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *International Conference on Machine Learning (ICML)*.

Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. 2018. Fine-grained evaluation of rule- and embedding-based systems for knowledge graph completion. In *International Semantic Web Conference (ISWC)*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016a. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1).

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016b. Holographic embeddings of knowledge graphs. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning (ICML)*.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 27(2).

Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems (NIPS)*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems (NIPS)*.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*.

Yanjie Wang, Rainer Gemulla, and Hui Li. 2018. On multi-relational link prediction with bilinear models. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *International World Wide Web Conference (WWW)*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. International Conference on Learning Representations (ICLR).