New Strategies for Learning Knowledge Graph Embeddings: The Recommendation Case

Nicolas Hubert[1,2](✉), Pierre Monnin[3], Armelle Brun[1], and Davy Monticolo[2]

[1] Université de Lorraine, CNRS, LORIA, Nancy, France
{nicolas.hubert,armelle.brun}@loria.fr
[2] Université de Lorraine, ERPI, Nancy, France
{nicolas.hubert,davy.monticolo}@univ-lorraine.fr
[3] Orange, Belfort, France
pierre.monnin@orange.com

Abstract. Knowledge graph embedding models encode elements of a graph into a low-dimensional space that supports several downstream tasks. This work is concerned with the recommendation task, which we approach as a link prediction task on a single target relation performed in the embedding space. Training an embedding model requires negative sampling, which consists in corrupting the head or the tail of positive triples to generate negative ones. Although knowledge graph embedding models and negative sampling have extensively been investigated for link prediction, their combined use for performing recommendations over knowledge graphs remains largely unexplored in the literature. In this work, we propose two specialization strategies for training embedding models and performing knowledge graph-based recommendations. Both strategies first train an embedding model on the whole knowledge graph. Then, during a specialization phase, a dedicated negative sampling scheme is applied to refine the pre-trained model. Experimental results on two public datasets demonstrate that a simple strategy which refines a pre-trained model by sampling random negative tails for the target relation proves to be very effective. This strategy significantly improves performance with respect to traditional rank-based evaluation metrics as well as a newly introduced metric that reflects the semantic validity of the top-ranked candidate entities.

Keywords: Negative Sampling · Knowledge Graph Embedding · Recommendation · Link Prediction · Ontology

1 Introduction

A knowledge graph (KG) is a collection of facts (h, r, t) where h (head) and t (tail) are two entities of the graph, and r is the semantic relation that links them. KGs are used for several tasks including entity matching, question answering

O. Corcho et al. (Eds.): EKAW 2022, LNAI 13514, pp. 66–80, 2022.
https://doi.org/10.1007/978-3-031-17105-5_5

and link prediction [24]. The latter is the focus of this paper. Link prediction consists in assessing the probability of existence of a given triple, for example in a knowledge graph completion perspective. Several approaches address the link prediction task, especially Knowledge Graph Embedding (KGE) methods [18,22]. They encode entities and relations of the KG into a low-dimensional vector space that preserves the structure of the original graph [22].

Training such KGE models requires both positive and negative triples. As KGs are usually made up of only positive triples, negative sampling (NS) is used to generate non-existent triples by corrupting the head or the tail of positive triples with any other entity from the KG [18]. Resulting triples are called negative samples and they constitute the basis on which embedding learning is performed: embedding models iteratively learn to assign higher ranks to true triples than to negative ones. Hence, the way these models learn is significantly influenced by negative sampling methods, which therefore received much attention recently [7,8,12].

These negative sampling methods usually intervene in link prediction tasks that consider all relations in the KG. However, in some application domains there is a target relation, i.e. a certain type of link that is of interest for prediction. In this work, we address the recommendation task and we approach it as a link prediction task on a single relation. More specifically, the target relation represents the link between users and items to recommend [4,17]. Recommendation consists in predicting one or a few relevant items of interest for a current user. In other words, for a given triple $(u, r, ?)$, the goal is to recommend the more relevant items for user u, with r denoting the nature of the recommendation based on the application context. For instance, a use case could be recommending university curricula to high school students. Although KGE models have successfully been used for recommendation [3,17], literature has primarily focused on KGE and NS for the more generic link prediction task. Predicting on a unique relation arguably responds to a different learning objective. Thus, we claim that new approaches are needed and we formulate the following research question:

RQ1. Does refining a KGE model pre-trained on a generic link prediction task by specializing training on the target relation improve recommendation performance?

Once a KGE model is trained, its quality with regard to the link prediction task is traditionally assessed using rank-based metrics. These metrics face some limitations as they only focus on the presence (or not) of the ground-truth in the top-K list. From our view, the ability of a model at predicting links that are semantically close to the ground-truth is a supplementary dimension that needs to be evaluated in order to have a more comprehensive view of a model quality. This way, the ability of a model to retain the semantic profile (i.e. range type) of the target relation can be better assessed. Please note that in the rest of the paper, we only consider the range type of the relation when referring to its semantic profile. This raises our second research question:

RQ2. What is the impact of different training strategies on the ability of KGE models to capture the semantic profile of the target relation?

Regarding **RQ1**, our motivation is to study one-hop KGE models that proved successful in downstream recommendation tasks [3,17], and determine whether their performance can be enhanced by specializing the training procedure. **RQ2** goes a step further, and we ask whether a more informed negative sampling method has a positive impact on the ability of an embedding model to retain the semantic profile of the target relation. The main contributions of our work are summarized as follows.

- We introduce two novel strategies for training knowledge graph embeddings for a downstream recommendation task.
- We introduce a new metric that measures the semantic validity of the top-ranked candidate entities.

The remainder of the paper is structured as follows. Related work about KGE models for recommendation and negative sampling is presented in Sect. 2. In Sect. 3, we detail the models and the proposed strategies for making KGE-based recommendations. Dataset descriptions, experimental settings and key findings are provided in Sect. 4. A discussion is provided in Sect. 5. Lastly, Sect. 6 summarizes the key findings and outlines directions for future research.

2 Related Work

2.1 Knowledge Graph Embeddings for Recommendation

Several methods have been proposed for making recommendations over KGs. Most recent approaches based on Graph Neural Networks (GNNs) showcase impressive performance [27]. Their success in recommendation tasks derives from their ability to model higher-order connectivity and encode sparse semi-supervised signals [5]. However, GNN-based recommender systems have some limitations: due to their large memory usage and significant training time, GNNs are not always applicable in real-world scenarios [27].

On the contrary, one-hop embedding models are simpler models that proved to be successful in downstream recommendation tasks [15]. Grad-Gyenge *et al.* [6] compare traditional collaborative filtering and embedding models for making recommendations over KGs. They clearly show that the latter significantly increase recommendation performance without suffering from an increasing amount of user interactions, contrary to traditional collaborative filtering algorithms. However, their experiments do not consider mainstream KGE models that are most commonly used. By contrast, [17] analyses the experimental results obtained with popular KGE models. Their work is close to our line of research, as it is concerned with the recommendation task. Although the authors clearly emphasize the superiority of KGE models over traditional baselines in a recommendation framework, they only focus on translational models which include TransE [1], TransH [25] and TransR [14], and do not consider other popular and effective KGE models [18]. In addition, they do not study the influence of negative sampling on recommendation performance.

2.2 Negative Sampling for Link Prediction in Knowledge Graphs

While KGs usually comprise positive triples only, training KGE models requires negative samples [12]. An early introduced method is Random Negative Sampling (RNS). It consists in replacing either the head h or the tail t of a triple with any other entity sampled uniformly from the set of observed entities in the KG [1]. However, sampling random entities uniformly can generate positive triples. For example, replacing the tail of (`CristianoRonaldo`, `playedFor`, `RealMadrid`) with `ManchesterUnited` would generate the triple (`CristianoRonaldo`, `playedFor`, `ManchesterUnited`) which actually represents a true fact. The approach presented in [25] reduces the risk of creating such false-negative triples by setting different probabilities for replacing the head and the tail based on the nature of the relation that links them: if the relation r is 1-to-N (e.g. `parentOf`), the head h has a higher probability of being replaced. If the relation r is N-to-1 (e.g. `bornIn`), the tail t is more likely to be replaced. More specifically, [25] uses a Bernoulli distribution to sample heads or tails with distinct probabilities.

However, the two aforementioned approaches cannot prevent the sampling procedure from producing semantically incorrect triples such as (`CristianoRonaldo`, `playedFor`, `SoccerShoes`). Such nonsensical triples do not provide the model with sufficient signal to learn from, which causes the notorious zero loss problem [23]. Therefore, more sophisticated methods have been proposed to generate high-quality negative samples and consequently give more hints to the model training [30]. In particular, ontological constraints and domain knowledge can be leveraged to create meaningful negative samples [10,13,26]. Intuitively, generating more realistic and robust negative samples helps the embedding model learn a better vector representation of the graph components. For instance, type-constrained negative sampling (TCNS) [13] replaces the head or the tail with a random entity belonging to the same type as the ground-truth entity. By doing so, only semantically valid triples are generated during negative sampling. TCNS has been found to work better than pure RNS in several scenarios [12,13]. However, it should be noted that entities are rarely typed [2,13]. To the best of our knowledge, RNS and TCNS have not been studied in the specific frame of single-relation link prediction, especially recommendation. It would be interesting to investigate whether they demonstrate greater efficiency in such a framework.

3 Training Embedding Models for Recommendation

Figure 1 outlines the whole approach that we further explicit in the following. In Sect. 3.1, we summarize the KGE models used in this work. Then, in Sect. 3.2 we elaborate on the specialization training and negative sampling strategies that we propose. These strategies are designed to fit the recommendation task. Finally, Sect. 3.3 details a newly introduced semantic-oriented metric that – combined with traditional rank-based metrics such as Hits@K – provides a more comprehensive view of the quality of a KGE model.

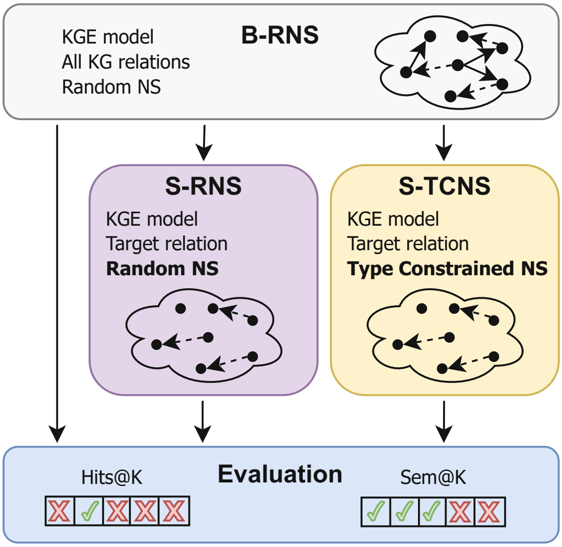

Fig. 1. Approach outline. S-RNS and S-TCNS are the two specialization strategies that we propose in this work. Both only include the target relation in the training procedure (dashed arrows), but differ regarding the negative sampling method. These two strategies are benchmarked against B-RNS in terms of Hits@K and our newly introduced semantic-oriented metric Sem@K.

3.1 Knowledge Graph Embedding Models

In line with [12], we use TransE [1], TransH [25], DistMult [28], and ComplEx [21], which all are very popular models that have proven to work well in a wide range of link prediction tasks. These models are detailed below.

TransE is the earliest translational model. It learns representations of entities and relations such that for a triple (h, r, t), $\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{e}_t$ where $\mathbf{e}_h$, $\mathbf{e}_r$ and $\mathbf{e}_t$ are the head, relation and tail embeddings, respectively. The scoring function is $f(h, r, t) = -d(\mathbf{e}_h + \mathbf{e}_r - \mathbf{e}_t)$ with d a distance function, usually the $L1$ or $L2$ norm. TransE does not properly handle 1-to-N, N-to-1 nor N-to-N relations [24] and yet has been found to be very efficient in multi-relational settings [3].

TransH 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 $f(h, r, t) = -d(\mathbf{e}_{h_\perp} + \mathbf{d}_r - \mathbf{e}_{t_\perp})$. TransH often showcases better performance than TransE with only slightly more parameters [25].

DistMult is a semantic matching model. It is characterized as such because it uses a similarity-based scoring function and matches the latent semantics of entities and relations by leveraging their vector space representations. More specifically, DistMult is a bilinear diagonal model that uses a trilinear dot product as its scoring function: $f(h,r,t) = \langle \mathbf{e}_h, \mathbf{W}_r, \mathbf{e}_t \rangle$. It is similar to RESCAL [16] – the very first semantic matching model – but restricts relation matrices $\mathbf{W}_r \in \mathbb{R}^{d \times d}$ to be diagonal. As the scoring function of DistMult is commutative, all relations are considered symmetric. This assumption does not hold in general. However, DistMult still achieves state-of-the-art performance in most cases [11].

ComplEx is also a semantic matching model. It extends DistMult by using complex-valued vectors to represent entities and relations: $\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t \in \mathbb{C}^d$. As a result, ComplEx is better able to model antisymmetric relations than DistMult [19]. Its scoring function uses the Hadamard product: $f(h,r,t) = \mathrm{Re}\left(\mathbf{e}_h \odot \mathbf{e}_r \odot \overline{\mathbf{e}}_t\right)$ where $\overline{\mathbf{e}}_t$ denotes the conjugate of $\mathbf{e}_t$.

3.2 New Specialization Strategies for Training KGE

To train the aforementioned KGE models, we consider the strategies depicted in Fig. 1. Contrary to the generic link prediction task, we assume that for a downstream recommendation task, the training phase should focus more on the target relation. However, the KGE model still needs to be trained considering all entities and relations before specializing in order to take into account all the available information in the KG. This motivates the pre-training step included in the two proposed strategies: both of them reuse a generic KGE model that was trained on the whole graph. It means that the training phase is first performed on all training triples, regardless of their relation. In the experiments, the generic KGE model trained on the whole graph with traditional RNS [1] (top of Fig. 1) serves as our baseline and is referred to as **B-RNS** (standing for baseline random negative sampling).

Once the KGE model is pre-trained, triples whose relation differs from the target relation are filtered out from the train set. For both strategies presented below, training resumes by specializing the training of the KGE model on the resulting filtered train set. The way the specialization phase is achieved depends on the negative sampling scheme used:

S-RNS stands for specialized training with random negative sampling. In this strategy, the embedding model is refined by applying uniform RNS when corrupting the tails of the remaining train triples.

S-TCNS stands for type-constrained negative sampling. In this strategy, the embedding model is refined by applying TCNS [13] when corrupting the tails of the remaining train triples.

3.3 Evaluating Recommendations: Hits@K and Sem@K

Hits@K is a rank-based metric extensively used for link prediction tasks. This metric accounts for the proportion of positive triples that are ranked in the top-K positions against a set of negative triples. Hits@K is also used in recommendation [29]. As this work focuses on the recommendation task, the generic definition of Hits@K [18] has to be modified to account for the fact that only tails are corrupted. The modified Hits@K is defined in Eq. (1).

$$\text{Hits@}K = \frac{1}{|\mathcal{B}|} \sum_{(h,r,t)\in\mathcal{B}} \mathbb{1}\left(\text{rank}_{(h,r)}(t) \leq K\right) \tag{1}$$

where $\mathcal{B}$ is a batch of positive triples and $\text{rank}_{(h,r)}(t)$ denotes the position of the ground-truth tail t in the sorted list of top-K entities scored by a given KGE model, for the head h, and the target relation r.

However, from our view, Hits@K remains limited when considering the recommendation task. Indeed, it only represents the ability of a model to rank the ground-truth higher in the top-K list. As such, this metric does not fully answer the following questions:

- When two KGE models have similar Hits@K, can we refine the evaluation process to safely favour one model?
- When the ground-truth does not show up in the top-K list, how do we assess the extent to which the KGE model has captured the semantic profile of the target relation?

Traditionally, the aforementioned questions have been addressed by considering additional rank-based metrics such as Mean Rank (MR) and Mean Reciprocal Rank (MRR) [18]. Compared to Hits@K, they take into account the position of the ground-truth without a threshold K. However, all rank-based metrics solely focus on the ground-truth at hand, without taking into consideration the other ranked entities. In some application domains, knowing the rank of the ground-truth is not sufficient. In the recommendation case, the catalog of items can be huge, so that recommending entities that are related to the ground-truth is also of interest. In the context of a KG, similarity between entities can be reflected through their types. As a result, it is desirable for a KGE model to also retain the semantic profile of the target relation in order to assign higher scores to triples that are semantically close to the ground-truth.

To this aim, we introduce Sem@K, a new semantic-oriented metric. Combining it with Hits@K gives a more comprehensive view of the quality of a KGE model. Sem@K reflects to what extent a K-list contains semantically valid candidates with regard to the range of the target relation. Hence the definition of Sem@K in Eq. (2):

$$\text{Sem@}K = \frac{1}{|\mathcal{B}|} \sum_{q\in\mathcal{B}} \frac{1}{K} \sum_{q'\in\mathcal{S}_q^K} \text{compatibility}(q, q') \tag{2}$$

where $\mathcal{S}_q^K$ is the top-K list of candidate triples scored by a given KGE model given a ground-truth triple q, and $\text{compatibility}(q, q')$ (Eq. (3)) assesses whether the candidate triple q' is semantically compatible with its ground-truth counterpart q. As this work focuses on the recommendation task, by semantic compatibility we refer to the fact that the predicted tail belongs to the range of the target relation:

$$\text{compatibility}(q, q') = \begin{cases} 1, & \text{if } \text{type}(q'_t) = \text{range}(q_r) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where type(e) returns the type of entity e and range(r) is the range of the relation r. q_r and q'_t denote the ground-truth relation and the tail of the candidate triple, respectively.

Compared to Hits@K that naturally increases or remains equal with higher values of K, Sem@K behaviour is non-monotonic. However, when the set of semantically valid candidates is limited, one should reasonably expect a decreasing Sem@K with higher K. Sem@K can be generalized to any other semantic context by adapting the semantic compatibility compatibility() operator with the context-dependent one. In our experiments, all strategies are evaluated using Hits@K and Sem@K, as depicted in Fig. 1.

4 Experiments

To address the relevance of the proposed specialization training strategies, experiments are conducted on real-world and public datasets and we subsequently evaluate these strategies in terms of Hits@K and the newly introduced Sem@K.

4.1 Datasets

As aforementioned, our contributions are evaluated on EduKG and KG20C, two KGs that we choose for their adequate entity typing: given a head-relation pair (h, r), the missing tail can only be of one single type. Consequently, we are able to study the influence of TCNS [13]. Both datasets naturally lend themselves to the recommendation task, whereas in most datasets used for link prediction, it is more questionable to favour one relation over the others. Below, we provide a thorough description of the KGs used in the experiments and indicate their respective target relation. Their main characteristics are summarized in Table 1.

EduKG[1] is an educational KG that authors of this work built and that is instantiated with students and a broad spectrum of university curricula. The goal is to recommend university curricula to high school students. More formally, for each triple of the form $(h, r, ?)$, we aim at retrieving the ground-truth tail t where h is an entity of type `Student`, r accounts for the relation `likedCurriculum` and t is an entity of type `Curriculum`. An example would consist in retrieving the correct tail for the following test triple: (`Bob`, `likedCurriculum`, `?`). Hence, for the S-TCNS strategy, only entities of type `Curriculum` are used to replace tails of positive triples during negative sampling. In EduKG, both the users (students) and items (curricula) are highly connected to other entities of the KG. For instance, students are linked to a `HighSchoolMajor`, they are characterized by `PersonalityTraits`, they have a certain amount of favorite `SchoolSubjects`, and they provide `Keywords` reflecting their interests. Likewise, curricula are

[1] purl.org/edukg/doc.

linked to recommended or mandatory HighSchoolMajors, they belong to one or more FieldsOfStudy, and they are also related to Keywords. Importantly, EduKG comprises a total of 286 curricula. This means there are 286 semantically valid tail entities out of 5,452 entities when it comes to recommending university curricula to students.

KG20C [20] is a scholarly KG encompassing 5 different entity types: Authors, Papers, Domains, Conferences, and Affiliations. The goal is to recommend a conference where to publish a paper. For each triple of the form $(h, r, ?)$, we aim at retrieving the ground-truth tail t where h is an entity of type Paper, r accounts for the relation publishedIn and t is an entity of type Conference. An example would be retrieving the correct tail for the following test triple: (LearningToEfficientlyRank, publishedIn, ?). In this case, the ground-truth tail is SIGIR. Hence, for the S-TCNS strategy, only entities of type Conference are used to replace tails of positive triples during negative sampling. Compared to EduKG, the set of semantically valid tails for the relation publishedIn is much more limited: KG20C comprises 20 distinct conferences out of the 16,362 observed entities.

Table 1. Characteristics of EduKG and KG20C. $|\mathcal{E}|$, $|\mathcal{R}|$ and $|\mathcal{T}|$ stand for the number of entities, relations and triples, respectively. $|\mathcal{V}|$ denotes the number of semantically valid tails for the target relation.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{V}\rvert$
EduKG	5,452	27	36,301	286
KG20C	16,362	5	55,607	20

The number of triples in EduKG and KG20C is comparable. However, it should be noted that EduKG contains fewer entities but more relations than KG20C (see Table 1).

4.2 Experimental Setup

The experiments are conducted using a 5-fold cross-validation. In each cross-validation setting, the test fold only comprises triples whose relation r is the target relation. TransE, TransH, DistMult, and ComplEx are implemented using PyTorch. Importantly, when comparing different training strategies, models are instantiated with the same initialization seed. The choice of the number of epochs, negative samples, embedding dimensions, and learning rate are based on what was found to work best for these datasets. Regardless of the strategy used, we first perform 1,000 epochs of general training with early-stopping (B-RNS) to ensure that training is achieved in a reasonable amount of time and that the best evaluation metrics are recorded. 100 epochs of specialized training are subsequently performed with early-stopping by applying the two strategies S-RNS and S-TCNS. These two strategies are compared against the baseline

B-RNS. All models are trained using max-margin loss and Adam optimizer as in [12]. The hyperparameters are shared across all configurations to ensure a fair comparison of their respective performance: number of negative triples per positive one $C = 50$ (in accordance with [21]), embedding dimension $k = 20$, learning rate $\eta = 0.01$, margin $\gamma = 1.0$. For TransE and TransH, distance $d = L2$ was used. Due to the presence of 1-to-N, N-to-1 and N-to-N relations in EduKG and KG20C, negative sampling can still generate false-positive triples after the corruption phase. Consequently, such positive triples are filtered out before ranking. In this work, only filtered Hits@K are reported [18]. Finally, all entities observed in the KG (regardless of their type) are scored during evaluation so that Sem@K can be reported.

4.3 Results

The experimental results of the strategies outlined in Sect. 3.2 are presented in Tables 2 and 3.

Table 2. Evaluation results on EduKG. H@K and S@K stand for Hits@K and Sem@K respectively, for $K \in \{1, 5, 10\}$. Green cells indicate which strategy performs best for a given model and metric.

	B-RNS			S-RNS			S-TCNS		
	H@1	H@5	H@10	H@1	H@5	H@10	H@1	H@5	H@10
TransE	4.66	16.12	25.49	6.27	22.66	33.64	8.19	26.75	37.82
TransH	7.19	22.96	35.51	9.67	30.02	43.49	12.16	34.38	47.80
DisMult	5.40	18.35	29.11	5.75	19.91	31.55	6.06	21.70	34.03
ComplEx	5.14	17.91	26.62	7.28	21.27	34.12	7.10	25.14	38.08
	S@1	S@5	S@10	S@1	S@5	S@10	S@1	S@5	S@10
TransE	97.39	94.79	92.80	99.69	98.94	98.36	96.21	91.62	88.19
TransH	99.13	98.35	97.39	99.61	99.39	98.98	99.17	97.90	96.17
DisMult	95.77	95.23	95.02	97.17	96.75	96.45	97.04	96.44	96.01
ComplEx	82.31	79.47	77.80	98.69	97.94	97.61	98.48	96.99	95.39

Baseline Strategy Comparison Across Datasets. In this section, we focus on the performance of the baseline model (B-RNS) regarding Hits@K. Results achieved under B-RNS are presented in the top left parts of Tables 2 and 3.

Overall, Hits@K are higher on KG20C than on EduKG. At first glance, this may seem counter-intuitive as the number of distinct entities is three times higher in KG20C ($\sim$16K) than in EduKG ($\sim$5K) (see Table 1). In this case, the lower Hits@K values achieved on EduKG compared to KG20C may reasonably be explained by the number of relations in EduKG, which exceeds the number of relations in KG20C.

Table 3. Evaluation results on KG20C. H@K and S@K stand for Hits@K and Sem@K respectively, for $K \in \{1, 5, 10\}$. Green cells indicate which strategy performs best for a given model and metric.

	B-RNS			S-RNS			S-TCNS		
	H@1	H@5	H@10	H@1	H@5	H@10	H@1	H@5	H@10
TransE	16.21	33.48	41.58	27.33	62.97	76.08	16.37	33.84	41.92
TransH	44.56	78.99	87.29	46.50	84.14	94.51	47.55	80.89	87.41
DisMult	23.05	36.57	40.89	28.54	46.20	51.97	27.67	43.03	47.37
ComplEx	24.98	39.54	44.40	32.21	57.28	63.03	27.45	49.16	58.87
	S@1	S@5	S@10	S@1	S@5	S@10	S@1	S@5	S@10
TransE	43.90	27.15	20.59	93.68	83.59	72.62	36.91	21.95	16.26
TransH	97.42	86.46	69.87	99.82	99.26	93.54	96.15	75.36	54.81
DisMult	51.46	30.30	20.72	73.48	49.20	34.74	67.77	38.59	24.22
ComplEx	54.91	31.16	20.48	86.72	64.04	44.06	74.53	59.84	47.36

On both datasets, the best performing model under B-RNS is TransH and the worst performing one is TransE. The superiority of TransH over TransE is in accordance with [25]. In terms of Hits@K, DistMult is slightly better than ComplEx on EduKG and slightly worse than ComplEx on KG20C. Recall that ComplEx has been proposed to better model antisymmetric relations than DistMult [21]. EduKG does not contain any antisymmetric relation, which may explain the relatively similar Hits@K between DistMult and ComplEx. However, in KG20C, there is one antisymmetric relation (`citedBy`), hence the slightly better Hits@K provided by ComplEx compared to DistMult.

RQ1: Impact of the Training Strategies on Hits@*K*. In order to answer the first research question, we evaluate the impact of the specialization strategies on the recommendation performance, as measured by Hits@K. From a more general point of view, we note that S-RNS and S-TCNS consistently achieve better Hits@K compared to B-RNS, for all $K \in \{1, 5, 10\}$ and regardless of the dataset and KGE model used (see Tables 2 and 3).

Differences exist between S-RNS and S-TCNS. On EduKG, S-TCNS provides better Hits@K than S-RNS (on average +35.7% for S-TCNS and +22.8% for S-RNS on Hits@10, compared to the results achieved with the baseline B-RNS). The reason may come from the fact that S-TCNS puts even more emphasis than S-RNS on discriminating between valid candidates for replacement, as the negative sampling is type-constrained. On KG20C, the conclusion differs: S-RNS performs systematically better than S-TCNS (on average +62.0% for S-RNS and +12.3% for S-TCNS on Hits@10, compared to the results achieved with the baseline B-RNS). This could be attributed to the low number of semantically valid candidates in KG20C: as there are only 20 entities of type `Conference`, S-TCNS is limited to a narrow set of semantically valid entities when performing negative sampling. As such, training may quickly reach a stage where performance cannot be improved anymore because most negative triples stop providing any further guidance to the model on improving the embeddings. This echoes the zero loss problem [23] mentioned in Sect. 2.2. In our experiments, we indeed noted that under S-TCNS, early-stopping was usually triggered before S-RNS, especially for TransE and TransH. For TransE, early-stopping was triggered after 7 epochs in average under S-TCNS and after 70 epochs in average under S-RNS. For TransH, early-stopping was triggered after 4 epochs in average under S-TCNS and after 22 epochs in average under S-RNS.

Consequently, when the set of semantically valid candidates is relatively large, S-TCNS seems to be an appropriate strategy as it helps the KGE model focus on semantically valid triples only, which are harder to differentiate. When the set of semantically valid candidates is limited, S-RNS is expected to perform better by providing more diverse and numerous samples to learn from.

To sum up, although S-RNS and S-TCNS do not provide comparable performance gains, they both improve results in terms of Hits@K compared to the baseline strategy B-RNS. This directly answers **RQ1**: specializing training on a single relation actually enhances the performance of a pre-trained KGE model in a downstream recommendation task.

RQ2: Impact of Training Strategies on Sem@*K*. To answer the second research question, we evaluate Sem@K on all models with the three strategies B-RNS, S-RNS and S-TCNS. First, the impact of S-RNS is significant: this strategy achieves the best results across all configurations. Interestingly, the use of S-RNS may even allow weaker models to compete with the other ones in terms of Sem@K. For example, ComplEx achieves disappointing Sem@K on EduKG under the baseline B-RNS. But when using S-RNS, Sem@K values obtained with ComplEx are comparable to the other models.

Focusing on the impact of S-TCNS on Sem@K, we note that on both datasets, the combined use of this strategy with the two translational models TransE and TransH deteriorates the model ability to capture the semantic profile of the target relation, compared to the results achieved under B-RNS. On the contrary, using S-TCNS enhances Sem@K of DistMult and ComplEx, which are two semantic-matching models. As evidenced by the green cells in Tables 2 and 3 for Sem@K, $K \in \{1, 5, 10\}$, we clearly see that S-RNS consistently performs better under all configurations. This answers **RQ2**: even though integrating the entity type into the negative sampling (S-TCNS) can improve the ability of a model to capture the semantic profile of the target relation, it is always a less interesting option compared to randomly sampling tails among all entities (S-RNS).

It appears that evaluating the quality of a KGE model by only using rank-based metrics provides a limited view of its performance. In some cases S-TCNS gives better Hits@K than S-RNS. By analyzing the influence of type-constrained sampling on the understanding of the semantic profile of the target relation, we see that S-TCNS performs poorly – sometimes even worse than the baseline B-RNS. Therefore, Sem@K is complementary to traditional rank-based metrics and special attention should be paid to this metric whenever there is a need for predicting semantically valid entities.

5 Discussion

Although the literature often presents RNS as an ineffective sampling strategy for the generic link prediction task [8,23], we have shown it can be successfully used in a recommendation framework when combined with a specialization training procedure. In particular, S-TCNS sometimes improves Hits@K at the expense of the model understanding of the semantic profile of the target relation. In this case, there seems to be a trade-off. On the contrary, S-RNS appears to pursue both objectives at the same time: improving the ability of a KGE model to assign a high score to the ground-truth, and better retain the semantic profile of the target relation. By generating both nonsensical and hard negatives, S-RNS reports substantial improvement over the baseline B-RNS. This leads us to think that when it comes to recommendation, focusing on triples whose relation is the target relation (S-RNS) may actually lead to better results than both focusing and using an informed negative sampling method (S-TCNS). Although S-RNS is a simple strategy that does not require any filtering or additional information – contrary to S-TCNS that requires knowing the type of each entity – we clearly demonstrate it could significantly enhance the quality of a KGE model in a downstream recommendation task.

Jain *et al.* [9] investigate whether embeddings are actually able to capture KG semantics. Although our research questions differ, we show that in the recommendation task, specific training procedures may improve the ability of an embedding model to capture the semantic profile of the target relation. Our experimental results even highlight that models that initially perform poorly on Sem@K could achieve competitive results after proper specialized training.

Compared to the baseline, S-RNS improves both Hits@K and Sem@K on the target relation. However, our work did not study whether the two proposed specialization strategies decrease the overall quality of a KGE model, i.e. its performance in terms of Hits@K and Sem@K for other relations. In addition, we choose RNS as our baseline. This may have an impact on the performance gains of the two specialization strategies. One may wonder whether choosing a stronger baseline would impact the outcomes of this work. Finally, it should be noted that in both EduKG and KG20C, the target relation cardinality is N-to-1. It would be interesting studying whether our proposed specialization strategies perform similarly on target relations of other cardinality, especially N-to-N relations that are frequently observed in recommendation.

6 Conclusion and Future Directions

In this work, we approach the recommendation task as a link prediction task on a single target relation of a knowledge graph, i.e. the relation that represents the links between users and items to recommend. To this aim, we consider knowledge graph embedding models and evaluate their performance with three training strategies: one baseline strategy that trains a model on the whole graph, and two specialization strategies that refine this model by focusing on the target relation. These two specialization strategies differ in their negative sampling method. Beside evaluating our models with the usual Hits@K metric, we introduce Sem@K, a new semantic-oriented metric that reflects the semantic validity of the top-ranked candidate entities. In light of these two metrics, we clearly show that the specialization strategy which refines a pre-trained embedding model by randomly sampling negative tails for the target relation consistently enhances both Hits@K and Sem@K compared to the baseline. In future works, we will extend our analysis using a broader range of models (e.g. GNNs). We will also address multi-typed KGs, different application domains, and propose an adjusted version of Sem@K that would benefit from further theoretical guarantees.

References

1. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Conference on Neural Information Processing Systems (NeurIPS) 2013, pp. 2787–2795 (2013)
2. Cai, L., Wang, W.Y.: KBGAN: adversarial learning for knowledge graph embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 1470–1480 (2018)
3. Chowdhury, G., Srilakshmi, M., Chain, M., Sarkar, S.: Neural factorization for offer recommendation using knowledge graph embeddings. In: Proceedings of the SIGIR Workshop on eCommerce, vol. 2410 (2019)
4. Edwards, G., Nilsson, S., Rozemberczki, B., Papa, E.: Explainable biomedical recommendations via reinforcement learning reasoning on knowledge graphs. arXiv preprint arXiv:2111.10625 (2021)
5. Gao, C., Wang, X., He, X., Li, Y.: Graph neural networks for recommender system. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 1623–1625 (2022)
6. Grad-Gyenge, L., Kiss, A., Filzmoser, P.: Graph embedding based recommendation techniques on the knowledge graph. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP, pp. 354–359 (2017)
7. Hajimoradlou, A., Kazemi, M.: Stay positive: knowledge graph embedding without negative sampling. arXiv preprint arXiv:2201.02661 (2022)
8. Islam, M.K., Aridhi, S., Smail-Tabbone, M.: Negative sampling and rule mining for explainable link prediction in knowledge graphs. Knowl. Based Syst. **250** (2022)
9. Jain, N., Kalo, J.-C., Balke, W.-T., Krestel, R.: Do embeddings actually capture knowledge graph semantics? In: Verborgh, R., et al. (eds.) ESWC 2021. LNCS, vol. 12731, pp. 143–159. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77385-4_9
10. Jain, N., Tran, T.-K., Gad-Elrab, M.H., Stepanova, D.: Improving knowledge graph embeddings with ontological reasoning. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 410–426. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_24
11. Kadlec, R., Bajgar, O., Kleindienst, J.: Knowledge base completion: baselines strike back. In: Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL, pp. 69–74 (2017)
12. Kotnis, B., Nastase, V.: Analysis of the impact of negative sampling on link prediction in knowledge graphs. arXiv preprint arXiv:1708.06816 (2017)
13. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 640–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_37

14. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2181–2187. AAAI Press (2015)
15. Liu, C., Li, L., Yao, X., Tang, L.: A survey of recommendation algorithms based on knowledge graph embedding. In: 2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI), pp. 168–171 (2019)
16. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML, pp. 809–816 (2011)
17. Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., Ferro, E.: Translational models for item recommendation. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 11155, pp. 478–490. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98192-5_61
18. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans. Knowl. Discov. Data **15**(2), 141–1449 (2021)
19. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations, ICLR (2019)
20. Tran, H.N., Takasu, A.: Exploring scholarly data by semantic query on Knowledge Graph Embedding Space. In: Doucet, A., Isaac, A., Golub, K., Aalberg, T., Jatowt, A. (eds.) TPDL 2019. LNCS, vol. 11799, pp. 154–162. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30760-8_14
21. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on Machine Learning, ICML, vol. 48, pp. 2071–2080 (2016)
22. Wang, M., Qiu, L., Wang, X.: A survey on knowledge graph embeddings for link prediction. Symmetry **13**(3), 485 (2021)
23. Wang, P., Li, S., Pan, R.: Incorporating GAN for negative sampling in knowledge representation learning. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 2005–2012 (2018)
24. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
25. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1112–1119 (2014)
26. Weyns, M., Bonte, P., Steenwinckel, B., Turck, F.D., Ongenae, F.: Conditional constraints for knowledge graph embeddings. In: Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG@ISWC), vol. 2635 (2020)
27. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: a survey. ACM Comput. Surv. (2020)
28. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: 3rd International Conference on Learning Representations, ICLR (2015)
29. Yang, Z., Ding, M., Zhou, C., Yang, H., Zhou, J., Tang, J.: Understanding negative sampling in graph representation learning. In: KDD '20: The 26th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, pp. 1666–1676. ACM (2020)
30. Zhang, Y., Yao, Q., Shao, Y., Chen, L.: Nscaching: Simple and efficient negative sampling for knowledge graph embedding. In: 35th IEEE International Conference on Data Engineering, ICDE, pp. 614–625 (2019)