MDNCaching: A Strategy to Generate Quality Negatives for Knowledge Graph Embedding

Tiroshan Madushanka[1,2,3](✉) and Ryutaro Ichise[1,2]

[1] SOKENDAI (The Graduate University for Advanced Studies), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan

[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{tiroshan,ichise}@nii.ac.jp

[3] University of Kelaniya, Kelaniya, Sri Lanka
tiroshanm@kln.ac.lk

Abstract. Knowledge graph embedding (KGE) has become an integral part of AI as it enables knowledge construction and exploring missing information. KGE encodes the *entities* and *relations* (elements) in a knowledge graph into a low-dimensional vector space. The conventional KGE models are trained using positive and negative examples by discriminating the positives from the negatives. However, the existing knowledge graphs contain only the positives. Hence, it is required to generate negatives to train KGE models. This remains a key challenge in KGE due to various reasons. Among them, the quality of the negatives is a critical factor for KGE models to produce accurate embeddings of the observed facts. Therefore, researchers have introduced various strategies such as Bernoulli negative sampling to generate quality negatives that are hard to distinguish from positives. However, fixed negative sampling strategies are suffering from vanishing gradients and false negatives. Later, the dynamic negative sampling techniques were introduced to overcome the vanishing gradient, but the false negatives still remain as a challenge to the research community. The present research introduces a new strategy called MDNCaching (Matrix Decomposed Negative Caching), which generates negatives considering the dynamics of the embedding space while exploring the quality negatives with large similarity scores. Matrix decomposition is used to eliminate false negatives, and hence, the MDNCaching ensures the quality of the generated negatives. The performance of MDNCaching was compared with the existing state-of-art negative sampling strategies, and the results reflect that the proposed negative sampling strategy can produce a notable improvement in existing KGE models.

Keywords: Negative sampling · Knowledge graph embedding · Matrix decomposition

H. Fujita et al. (Eds.): IEA/AIE 2022, LNAI 13343, pp. 877–888, 2022.
https://doi.org/10.1007/978-3-031-08530-7_74

1 Introduction

A Knowledge Graph (KG) is a structured representation of facts, textual data in the form of $(head, relation, tail)$ known as a triplet, e.g., (*Shakespeare, isAuthorOf, Hamlet*). Knowledge graphs are constructed using the knowledge bases such as Freebase, DBpedia, WordNet, and YAGO. KGs have been utilized in many real-world applications, such as question answering, recommendation systems, and information retrieval. Although the knowledge bases contain vast volumes of facts, the KGs are often incomplete as they are created based on the available facts or the ground truth, which are often dynamic and evolving. For example, when considering the people's birthplaces, 71% and 66% are not found in Freebase and DBpedia, respectively. Therefore, it is worth having methods to complete the KGs automatically by adding the missing knowledge or the facts.

Recent research have shown that Machine Learning (ML) methods can be effectively used to complete knowledge graphs. However, applying ML methods is still a challenging task due to various facts such as high dimensionality. As a solution, knowledge graph embedding methods have been introduced by past research. Knowledge graph embedding (KGE) which maps *entities* and *relations* into a low dimensional vector space while preserving their semantic meaning. Moreover, KGE overcomes the difficulties in manipulating textual data in knowledge graphs, such as sparseness and computational cost [9]. Modern KGE strategies have shown promising results in knowledge acquisition tasks such as link prediction, triplet classification, and knowledge graph completion. Typically, KGE models accelerate training ML algorithms by extending the motivation of ranking the observed instances (positives) higher than the unobserved instances (negatives). However, the knowledge bases contain only positive examples. Hence, it is necessary to explore strategies to generate quality negatives that are hard to distinguish from positives as they have high similarity but are negatives. For instance, considering the positive (*Shakespeare, isAuthorOf, Hamlet*), we say that the generated negative (*Shakespeare, isAuthorOf, TheWidow'sTears*) is a quality negative as it is hard to distinguish from positives instead the negative (*Shakespeare, isAuthorOf, London*). Generation of quality negatives enhances the KG embeddings, which is always challenging. Therefore, negative sampling becomes indispensable in knowledge representation learning as the KGE model's performance heavily relies on negative selection.

Most of the state-of-the-art strategies in generating negatives consider corrupting positives randomly (e.g., [2,14]), based on closed world assumption, or exploiting the KG structure when generating quality negatives (e.g., [1,17]). However, these strategies suffer from false negatives as they do not guarantee that the generated ones are always relevant, i.e., generating latent positives as negatives. As KGE models are sensitive to inputs, false negatives usually fool the model, losing the semantics of *entities* and *relations*. Furthermore, strategies that randomly corrupt positives suffer from vanishing gradients as they tend to generate triplets with zero gradients during the training phase.

To overcome the stated challenges, the present work proposes a negative sampling strategy that explores negatives, considering the dynamic distribution of embedding space and reducing false negatives by adopting matrix decomposition. We first trained the latent relation model that uses positives to utilize the matrix decomposition. Then we predict the latent relations and refer them with negative candidates generation. We utilize a caching technique to manage negative triplets with large similarity scores. To overcome the vanishing gradients problem, we up-date negative candidates concerning the changes to the embedding space with KGE model training. Furthermore, we propose a selection criterion that ensures "exploration and exploitation" that balances exploring all possible quality negative candidates and sampling a fixed number of negatives close to the positives.

The remainder of this paper is organized as follows. Section 2 discusses related work with knowledge graph embedding and negative sampling. In Sect. 3, we propose a new strategy in generating quality negatives with large similarity scores considering the dynamic distribution of embedding space while eliminating false negatives adopting matrix decomposition. In Sect. 4, we present an experimental study in which we compare our proposed negative sampling strategy with baseline results of benchmark datasets and analyze results with state-of-the-art. In Sect. 5, we conclude this paper.

2 Related Work

Various research work has been conducted in Negative Sampling and Knowledge Graph Embedding. KGE maps knowledge graph elements, i.e., *entities* and *relations*, into low dimensional continuous vector space to use numerical representation when carrying out knowledge acquisition tasks. Commonly, three mainstream KGE models are found: translational distance-based models, semantic matching-based models, and neural network approaches. Translational distance-based models represent the distance of projected KG elements (e.g., [2,6,14]). Using matrix decomposition, semantic matching-based models represent latent semantics organized in vectorized *entities* and *relations* (e.g., [8,11,16]). In addition, Neural network approaches have also gained attention in recent research work that utilizes the potential of neural networks and variants (e.g., [4,5]). Typically, KGE models learn knowledge representation by discriminating positives from negatives made by corrupting positives. However, the quality of negatives affects training and performances of knowledge representation downstream tasks. In abstract, knowledge graph embedding work has focused on providing a better representation for connection between *entities* and *relations* of the knowledge graph, while negative sampling strategies have focused on boosting the underlying embedding model.

2.1 Negative Sampling

Among the existing negative sampling strategies, Uniform negative sampling is a widely used strategy due to its simplicity and efficiency. For example, TransE [2], ComplEx [11] and DistMult [16] use Uniform negative sampling to generate negatives. Uniform negative sampling randomly corrupts positives by replacing *head* or *tail* entities, and it reflects that generated negatives do not contribute to knowledge representation learning in most cases and generate false negatives. The Bernoulli negative sampling strategy was proposed to overcome this limitation by considering relation cardinality (i.e., 1-N, N-N, and N-1) to reduce false negatives [14]. However, both Uniform and Bernoulli sampling strategies are fixed sampling schemes. They both suffer from vanishing gradients as they generate triplets with zero gradients during the training phase [17]. Hence, Generative adversarial networks (GAN) based negative sampling strategies IGAN [12] and KBGAN [3] were introduced to generate negatives with large similarity scores considering the dynamic distribution of embeddings. The GAN-based strategies adversarially train the discriminator to produce quality negatives concerning a pre-trained KGE model as the generator. In KBGAN, the generator generates a candidate set of uniformly sampled negatives, i.e., $Neg = (\bar{h}, r, \bar{t})$, selects one with the highest probability from set Neg, and then feeds to the discriminator that minimizes marginal loss between positives and negatives to improve the final embedding. However, GAN-based strategies suffer from high variance in REINFORCE gradient [15], and the generator introduces additional parameters. Both KBGAN and IGAN require pre-trained, which adds extra costs. Recently, Structure Aware Negative Sampling (SANS) [1] strategy was introduced with a different perspective that utilizes available graph structure by selecting negatives considering the neighborhood. Since SANS explores potential negatives within a k-hop neighborhood, SANS also increases the possibility of generating false negatives. NSCaching [17] was proposed to overcome the challenges in generating quality negatives by introducing a cache that maintains negative triplets with large similarity scores and updating the cache using importance sampling. Despite this, NSCaching may produce false negatives with a high possibility as the latent positives also reflect large similarity scores.

3 MDNCaching

Although the literature shows diverse KGE models, generating quality negatives remains as a fundamental challenge in KGE. The present research introduces a new strategy called MDNCaching, which generates negatives considering the dynamics of the embedding space while exploring the quality negatives with large similarity scores. The novel strategy addresses existing challenges; 1). reducing false negatives in the generated candidates, and 2). generation of quality negatives with large similarity scores. To this end, we introduce a novel strategy that combines the dynamic updates of the embedding space to overcome the challenge of generating quality negatives with large similarity scores, avoiding the vanishing gradients problem. More precisely, the Matrix Decomposition technique is utilized to model the latent relations when avoiding false negatives that enhance the quality of the generated negatives. Before delving into the details of the proposed strategy, it is worth knowing the idea of matrix decomposition, a critical component of the proposed strategy.

3.1 Matrix Decomposition

The matrix decomposition (MD) technique utilizes matrix multiplication to generate latent features. Collaborative filtering is the typical application of MD to identify ratings between *item* and *user* entities [7]. Referring to collaborative filtering context, let U be a set of $Users$, D be a set of *items*, and R be a *rating* matrix between U and D, i.e., $R = R_{|U| \times |D|}$, including all product ratings given by users. With matrix decomposition, the goal is to generate latent rating features, given that the input of two matrices, P that represents the association between a *user* and features, and Q that represents the association between an *item* and features, i.e., $R \approx P \times Q^\top$ [7].

Even though the matrix decomposition techniques are utilized with KGE (e.g., [8]), to the best of our knowledge, the MD technique is yet under utilized in the negative sampling strategies. Considering the benefits of MD techniques in modeling hidden semantics, we apply a matrix decomposition technique to model the latent *relations* to predict potential false negatives. In our model, let h be a set of $Heads$, t be a set of $Tails$, and R be a *relation* matrix between h and t, i.e., $R = R_{|h| \times |t|}$, includes all relations between entities. Our goal is to generate latent relations referring to the matrix decomposition model such that $R \approx H \times T^\top$ where H represents the association between a *head* and features, and T represents the association between a *tail* and the features.

3.2 The Proposed Strategy

This section describes the proposed negative sampling strategy MDNCaching. Recall the stated challenges in negative sampling 1). reduce false negatives that fool the KGE model to lose the semantics of the KG, and 2). adopt dynamics of the embedding space when generating quality negatives with large similarity scores to avoid the vanishing gradient. The proposed strategy enhances the KGE by generating quality negatives with large similarity scores while reducing the possible false negatives in the sampling space.

The proposed MDNCaching is a dynamic distribution-based negative sampling strategy that integrates a matrix decomposition technique and utilizes the dynamics of the knowledge graph embedding to address the stated challenges. We integrate the matrix decomposition technique to eliminate false negatives by predicting latent *relations*. The reduction of false negatives decreases the possible discrimination on latent positives and enhances the KGE. We consider frequent updates to the embedding space to overcome the issue of generating quality negatives with large similarity scores. Furthermore, we utilize a caching technique that maintains negatives with large similarity scores for each positive in the training set $\mathcal{S}$. Two caches are separately maintained as head-cache $\mathcal{H}$ that maintains candidates for *head* corruption and indexes negatives with *tail* and *relation* (t, r), while tail-cache $\mathcal{T}$ maintains candidates for *tail* corruption and indexes negatives with *head* and *relation* (h, r). We uniformly sample a negative from the cache efficiently without introducing any bias. The lazy update

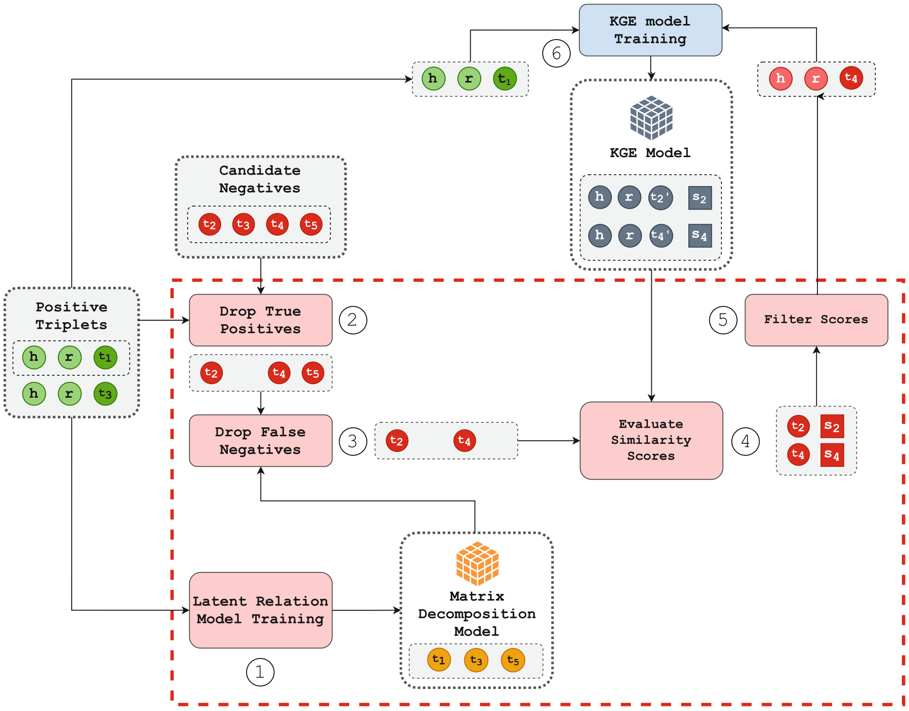

Fig. 1. The framework for the proposed negative sampling strategy MDNCaching.

technique in updating caches refreshes the cache after $\mathcal{N}$ number of epochs later rather than immediate.

Our framework for the proposed negative sampling strategy is depicted in Fig. 1, illustrating steps in generating a quality negative with *tail* corruption scenario. The proposed MDNCaching consists of six critical steps in generating quality negatives and executing the KGE task. In step 1, MDNCaching performs the **latent relation model training**. Detection and elimination of false negatives are critical tasks in the proposed negative sampling strategy. In order to predict latent positives, the matrix decomposition model is trained concerning the observed KG elements. In step 2, MDNCaching **drops true positives** from the candidate negatives. The candidate negatives are initiated as *entity* space $\mathcal{E}$ except the given positive elements. This technique ensures that the proposed strategy explores candidate negatives best. For example, given the positive (h, r, t_1), MDNCaching initializes the candidate negatives as $\{t_2, t_3, t_4, t_5\}$ where $\mathcal{E} = \{h, t_1, t_2, t_3, t_4, t_5\}$. However, the candidate negatives may comprise true positives since KG consists of 1-N, N-N and N-1 relations. Therefore, it is essential to drop true positives from the candidate negatives (e.g., given positive (h, r, t_3), candidate $\{t_3\}$ is removed from candidate negatives resulting $\{t_2, t_4, t_5\}$ as candidates). In step 3, MDNCaching **drops false negatives** from the can-

Algorithm 1: KGE model training with negatives from MDNCaching.

```
Input: Knowledge graph $\mathcal{G}$, and the matrix decomposition model $f_{md}$
Output: Knowledge graph embeddings
Initialize head-features embeddings $\forall h \in \mathcal{E}$, and tail-features embeddings
  $\forall t \in \mathcal{E}$ and train $f_{md}$ for a certain number of epochs.
Initialize caches; head-cache $\mathcal{H}$, tail-cache $\mathcal{T}$ using Algorithm 2.
Initialize knowledge graph embeddings $\forall e \in \mathcal{E}$, $\forall r \in \mathcal{R}$.
Loop
    foreach $(h, r, t) \in \mathcal{G}$ do
        Index $\mathcal{H}$ by $(t, r)$, i.e.,$\mathcal{H}_{(t,r)}$ and $\mathcal{T}$ by $(h, r)$, i.e.,$\mathcal{T}_{(h,r)}$.
        Uniformly sample $\bar{h} \in \mathcal{H}_{(t,r)}$ and $\bar{t} \in \mathcal{T}_{(h,r)}$.
        Select negative $(\bar{h}, r, \bar{t})$ either as $(\bar{h}, r, t)$ or $(h, r, \bar{t})$ considering the
          relation cardinality of $r$.
        Update knowledge graph embeddings discriminating $(h, r, t)$ against
          $(\bar{h}, r, \bar{t})$.
    end foreach
    update cache $\mathcal{H}$ and $\mathcal{T}$ using Algorithm 2;
end
```

didate negatives utilizing the trained MD model at step 1. It is essential to identify false negatives before the score filtration since they also contain large similarity scores. Therefore, the proposed strategy predicts latent relations to exclude false negatives from the candidate negatives (e.g., given the (h, r) pair, $\{t_1, t_3, t_5\}$ are predicted, and the $\{t_5\}$ is removed from the candidate negatives, which is the latent). In step 4, the proposed strategy **evaluates similarity scores** for the candidate negatives, referring to the baseline scoring function. Since MDNCaching drops true positives and false negatives, the candidate negatives consist of potential negatives. In step 5, the quality negatives are filtered considering the similarity score, i.e., **filter scores**, and negatives with large similarity scores are selected (e.g., given s_2 and s_4 are similarity scores for (h, r, t_2), (h, r, t_4) respectively, and given $s_4 \geq \textit{filtration threshold}$, we update the candidate negatives as $\{t_4\}$). In step 6, the proposed strategy performs the **KGE model training** by discriminating provided positives (e.g., (h, r, t_1)) against generated negatives (e.g., (h, r, t_4)).

3.3 Integration of MDNCaching with KGE Framework

Figure 1 describes the general framework of negative sample generation. However, we utilize a caching technique to manage generated negatives effectively in MDNCaching. Therefore, we describe the integration of MDNCaching with the typical KGE framework and the utilization of the caching technique in Algorithm 1. First, the matrix decomposition model training is performed. Then, the caches are initialized. Generally, we generate a triplet as a candidate for negatives by replacing either *head* or *tail*. The generated negatives are stored in two separate caches , i.e., head-cache $\mathcal{H}$ (indexed by (t, r)) and tail-cache $\mathcal{T}$ (indexed

Algorithm 2: MDNCaching cache update considering the dynamic distribution of embedding space.

```
Input: Baseline scoring function $f$, knowledge graph $\mathcal{G}$, matrix decomposition
       model $f_{md}$
Output: head-cache $\mathcal{H}$ and tail-cache $\mathcal{T}$
1 foreach $(h,r,t) \in \mathcal{G}$ do
2     Initialize negative candidates for $\mathcal{H}_{(t,r)}$ and $\mathcal{T}_{(h,r)}$.
3     Remove true positives from $\mathcal{H}_{(t,r)}$ and $\mathcal{T}_{(h,r)}$.
4     Predict latent relations from $f_{md}$, $\forall \bar{h} \in \mathcal{H}_{(t,r)}$, $\forall \bar{t} \in \mathcal{T}_{(h,r)}$ and drop false
        negatives from $\mathcal{H}_{(t,r)}$ and $\mathcal{T}_{(h,r)}$.
5     Evaluate similarity scores $\forall \bar{h} \in \mathcal{H}_{(t,r)}$ and $\forall \bar{t} \in \mathcal{T}_{(h,r)}$ considering the
        baseline scoring function $f$, i.e., $f(\bar{h},r,t)$ and $f(h,r,\bar{t})$ respectively.
6     Select candidates with large similarity scores from $\mathcal{H}_{(t,r)}$ and $\mathcal{T}_{(h,r)}$.
7 end foreach
```

by (h, r)). Next, KGEs are initialized for KG elements, and KGE model training is performed iteratively for a certain number of epochs. When a positive triplet is received, the head-cache $\mathcal{H}$ and the tail-cache $\mathcal{T}$ are indexed. Then, a candidate negative triplet is generated referring to $\mathcal{H}_{(\mathrm{t,r})}$ and $\mathcal{T}_{(h,r)}$. Since the caches maintain quality negatives with large similarity scores, selecting any candidate from $\mathcal{H}_{(\mathrm{t,r})}$ or $\mathcal{T}_{(h,r)}$ avoids the vanishing gradient problem with high probability. Then, it performs the typical embedding update task referring to the baseline KGE model. Finally, caches are updated, adopting the changes to the KGE space, and strategy refers the Algorithm 2 to populate quality negatives in the head-cache $\mathcal{H}$ and the tail-cache $\mathcal{T}$. Algorithm 2 describes the process of generating quality negatives with large scores following the previously described steps 2–5 iteratively for each element in the KG.

In summary, the proposed MDNCaching strategy introduces an additional step to train a matrix decomposition model before KGE model training, and it introduces a caching technique to manage generated candidate negatives effectively. With flexibility in integrating any translational distance-based or semantic matching-based model, MDNCaching enables robustness in training models from scratch with fewer parameters than previous dynamic negative sampling work IGAN [12], and KBGAN [3]. The generator in GAN approaches tends to generate correct facts that are considered as *positives* instead of *negatives*, and in contrast to GAN approaches, the proposed MDNCaching strategy considers latent relations to eliminate plausible positive facts from the negative candidates by utilizing the matrix decomposition technique. Besides, MDNCaching extends the idea of caching candidate negative that proposed in NSCaching [17]. The proposed strategy *explores* the candidate space to the best at step 2 and *exploits* the candidates by carefully managing caches at step 5. In addition to that, the exploration of negatives with large similarity scores effectively impacts embedding training.

Table 1. Scoring functions for triple (h, r, t), and parameters. $diag(r)$ constructs the diagonal matrix with r.

Model	Scoring function	Definition	Parameters
Translational distance-based	TransE [2]	$\lVert h + r - t \rVert_i$	$h, r, t \in \mathbb{R}^n$
	TransD [6]	$\left\lVert h + w_r w_h^\top h + r - (t + w_r w_t^\top t) \right\rVert_i$	$h, r, t, w_h, w_t, w_r \in \mathbb{R}^n$
Semantic matching-based	DistMult [16]	$h \cdot diag(r) \cdot t^\top$	$h, r, t \in \mathbb{R}^n$
	ComplEx [11]	$Re(h \cdot diag(r) \cdot t^\top)$	$h, r, t \in \mathbb{C}^n$

Table 2. Statistics of the datasets used with experiments

Dataset	#entity	#relation	#train	#valid	#test
WN18RR	93,003	11	86,835	3,034	3,134
FB15K237	14,541	237	272,115	17,535	20,466

4 Experiments

We evaluated the proposed negative sampling strategy, i.e., MDNCaching, on the link prediction in KGs and compared results with the state-of-the-art negative sampling strategies. In this case, the task was to predict the missing *head* (h) or *tail* (t) entity for a positive triplet (h, r, t) and evaluate the rank of the *head* and *tail* entities among all predicted entities. We evaluated the results for link prediction with TransE [2], TransD [6], DistMult [16], and ComplEx [11] baseline KGE models, and definitions are described in Table 1.

4.1 Experimental Setup

Datasets. The experiments were conducted on two popular benchmark datasets WN18RR [13] and FB15K237 [10]. These datasets were constructed by removing inverse-duplicate relations from previous WN18 and FB15K datasets respectively. The experiments were carried out in these two variants as they were more challenging and realistic than originals. The statistics of the data sets are described in Table 2.

Performance Measurement. We consider the "Filtered" setting with performance evaluation so that valid entities outscoring the target are not considered mistakes. Hence they are skipped when computing the rank. We evaluate results based on the following metrics,

1. *Mean Rank (MR)* is the average of the obtained ranks; $MR = \frac{1}{|Q|} \sum_{q \in Q} q$. The smaller value of MR tends to infer better results. However, since MR is susceptible to outliers, the Mean Reciprocal Rank is widely used.

2. *Mean Reciprocal Rank (MRR)* is the average of the inverse of the obtained ranks; $MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{q}$. The higher value of MRR tends to infer better results.
3. *Hit@K* is the ratio of predictions for which the rank is equal or lesser than a threshold k; $Hits@K = \frac{|\{q \in Q : q \leq K\}|}{|Q|}$. The higher value of Hits@K tends to infer better results.

Optimization and Implementation. A knowledge graph model was optimized by minimizing the objective function with Adam optimizer, and first, we tuned hyper-parameters referring to Bernoulli sampling strategy based on MRR. We conducted the evaluation for 1000 epochs and presented the best result for MRR. We started our experiments within the following ranges for hyper-parameters: embedding dimension $d \in \{50, 100, 250, 1000\}$, learning rate $\eta \in \{0.0005, 0.005,\ 0.05, 0.5\}$, margin value $\gamma \in \{1, 2, 3, 4, 5\}$ and optimized for best performance.

Results. We compare results with state-of-the-art negative sampling strategies concerning the reported performance comparison in NSCaching [17] work for Bernoulli, KBGAN, NSCaching concerning the training from scratch. Also, we directly consider the reported performance in SANS [1]. The performance comparison on link prediction is summarized in Table 3. When comparing results on translational distance-based, it is evident that the proposed negative sampling strategy gains substantial improvement for both datasets, i.e., WN18RR and FB15K237. When evaluating results for semantic matching-based KGE models, we observe that the proposed strategy outperforms the state-of-the-art negative sampling strategies. One can observe that MDNCaching consistently achieves better results with ComplEx than the state-of-the-art negative sampling strategies for both datasets with substantial improvements (i.e., Hits@10 by 4.40% and 10.91% for WN18RR and FB15K237 respectively). Although some results are competitive, experimental results reflect that MDNCaching enhances link prediction tasks against the state-of-the-art negative sampling strategies. For instance, when considering the Hits@10, we can witness 7.83% and 10.91% improvement with TransE and ComplEx respectively for the FB15K237 dataset while we observe 4.64% and 4.40% improvement with DistMult and ComplEx respectively for the WN18RR dataset. The results evidence that the proposed negative sampling strategy effectively enhances the KGE by generating quality negatives. The substantial improvements in MRR and Hits@10 reflect that the MDNCache successfully overcomes the stated challenges with negative generation.

Table 3. Comparison of state-of-the-art negative sampling strategies on WN18RR and FB15K237 datasets. Note that results of MR and results for TransD and ComplEx embedding models for SANS [1] is not available as the original did not include. We consider SANS with the random walk configuration.

Score function	Negative sampling strategy	WN18RR			FB15K237		
		MRR	MR	Hits@10	MRR	MR	Hits@10
TransE	Bernoulli [17]	0.1784	3924	45.09	0.2556	197	41.89
	KBGAN [17]	0.1808	5356	43.24	0.2926	722	46.59
	SANS [1]	0.2317	–	**53.41**	0.2981	–	48.50
	NSCaching [17]	0.2002	4472	47.83	0.2993	**186**	47.64
	MDNCaching	**0.2390**	**3054**	53.20	**0.3330**	200	**52.30**
TransD	Bernoulli	0.1901	3555	46.41	0.2451	**188**	42.89
	KBGAN	0.1875	4083	46.41	0.2465	825	44.4
	SANS	–	–	–	–	–	–
	NSCaching	**0.2013**	**3104**	**48.39**	**0.2863**	189	**47.85**
	MDNCaching	0.1737	4477	48.36	0.2683	354	46.43
DistMult	Bernoulli	0.3964	7420	45.25	0.2491	280	42.03
	KBGAN	0.2039	11351	29.52	0.2272	276	39.91
	SANS	0.4071	–	49.09	0.2621	–	41.46
	NSCaching	**0.4128**	7708	45.45	**0.2834**	**273**	**45.56**
	MDNCaching	0.3921	**2946**	**51.37**	0.2694	403	44.10
ComplEx	Bernoulli	0.4431	**4693**	51.77	0.2596	238	43.54
	KBGAN	0.3180	7528	35.51	0.1910	881	32.07
	SANS	–	–	–	–	–	–
	NSCaching	0.4463	5365	50.89	0.3021	**221**	48.05
	MDNCaching	**0.4729**	5312	**54.05**	**0.3594**	415	**53.29**

5 Conclusion

The present research proposed MDNCaching, which is an enhanced negative sampling strategy for KGE, addressing the problem of false negatives by reducing latent positives predicting through matrix decomposition. The proposed strategy effectively manages separate caches for *head* and *tail* candidates that contain quality negatives with large similarity scores, adopting the dynamic changes in the embedding space. Experimentally, we evaluated the MDNCaching on two datasets and four scoring functions covering translational-distance and semantic matching models. Experimental results reflect a substantial enhancement with TransE, DistMult, and ComplEx KGE models. Notably, the ComplEx KGE model with MDNCaching improves both datasets considerably. When carefully balanced the exploration and exploitation, MDNCaching requires considerable memory as it explores possible candidates, and utilization of memory handling will proceed as future works. Also, possible enhancements with latent relation prediction will continue for our future works.

References

1. Ahrabian, K., Feizi, A., Salehi, Y., Hamilton, W.L., Bose, A.J.: Structure aware negative sampling in knowledge graphs. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 6093–6101 (2020)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of International Conference on Neural Information Processing Systems, vol. 26, pp. 2787–2795 (2013)
3. Cai, L., Wang, W.Y.: KBGAN: adversarial learning for knowledge graph embeddings. In: Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics, pp. 1470–1480 (2018)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of AAAI Conference on Artificial Intelligence, vol. 32, pp. 1811–1818 (2018)
5. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–610 (2014)
6. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of International Joint Conference on Natural Language Processing, pp. 687–696 (2015)
7. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**, 30–37 (2009)
8. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of International Conference on Machine Learning, pp. 809–816 (2011)
9. Shan, Y., Bu, C., Liu, X., Ji, S., Li, L.: Confidence-aware negative sampling method for noisy knowledge graph embedding. In: Proceedings of IEEE International Conference on Big Knowledge, pp. 33–40 (2018)
10. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of Workshop on Continuous Vector Space Models and their Compositionality, pp. 57–66 (2015)
11. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of International Conference on Machine Learning, pp. 2071–2080 (2016)
12. Wang, P., Li, S., Pan, R.: Incorporating GAN for negative sampling in knowledge representation learning. In: Proceedings of AAAI Conference on Artificial Intelligence, vol. 32, pp. 2005–2012 (2018)
13. Wang, Y., Ruffinelli, D., Gemulla, R., Broscheit, S., Meilicke, C.: On evaluating embedding models for knowledge base completion. In: Proceedings of Workshop on Representation Learning for NLP, pp. 104–112 (2019)
14. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI Conference on Artificial Intelligence, pp. 1112–1119 (2014)
15. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**, 229–256 (1992)
16. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of International Conference on Learning Representations (2015)
17. Zhang, Y., Yao, Q., Shao, Y., Chen, L.: NSCaching: simple and efficient negative sampling for knowledge graph embedding. In: Proceedings of IEEE International Conference on Data Engineering, pp. 614–625 (2019)