PASSLEAF: A Pool-bAsed Semi-Supervised LEArning Framework for Uncertain Knowledge Graph Embedding

Zhu-Mu Chen,[1] **Mi-Yen Yeh,** [2] **Tei-Wei Kuo** [1,3,4]

[1] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[3] Department of Computer Science, City University of Hong Kong, Hong Kong
[4] Graduate Institute of Networking and Multimedia, National Taiwan University, Taiwan

Abstract

In this paper, we study the problem of embedding uncertain knowledge graphs, where each relation between entities is associated with a confidence score. Observing the existing embedding methods may discard the uncertainty information, only incorporate a specific type of score function, or cause many false-negative samples in the training, we propose the PASSLEAF framework to solve the above issues. PASSLEAF consists of two parts, one is a model that can incorporate different types of scoring functions to predict the relation confidence scores and the other is the semi-supervised learning model by exploiting both positive and negative samples associated with the estimated confidence scores. Furthermore, PASSLEAF leverages a sample pool as a relay of generated samples to further augment the semi-supervised learning. Experiment results show that our proposed framework can learn better embedding in terms of having higher accuracy in both the confidence score prediction and tail entity prediction.

1 Introduction

Knowledge graph (KG) embedding has drawn plenty of attention in the past decade because the low dimensional representation makes various machine learning models working on the structural knowledge possible. As the emergence of *uncertain* knowledge graphs, which reflect the plausibility of knowledge facts in practice by associating an entity-relation-entity triplet with a confidence score, existing embedding approaches are not suitable because they naively neglect such plausibility information.

To our knowledge, UKGE (Chen et al. 2019) is the first and the only embedding method designed for uncertain knowledge graphs. The main idea is to map the score function, which originally measures the plausibility of a (head, relation, tail) triplet in the vector space, to a scalar predicting the corresponding confidence score and make the loss function the mean square error of the exact and the estimated confidence score.

In light of the success of UKGE, we further find several important issues to be solved so that we can significantly improve the embedding quality of uncertain KG. First, during the training process, all the existing embedding methods will randomly draw samples from the unseen (head, relation, tail) triplets and treat them as negative samples, i.e., triplets with zero confidence score. In fact, there can be lots of triplets with various confidence scores outside the training set. For instance, the following two triplets (elephant, has a, black eye) and (October, has property, cold) may not exist in the training data sets but their existence, in reality, is possible and the corresponding confidence scores should not be zero. We call this a *false-negative problem for training* and aim to solve it. Second, UKGE is specifically tailored for the score function of DisMult (Yang et al. 2015), a kind of semantic-based embedding method. It is not trivial to extend UKGE for other types of embedding methods such as translational-distance methods, some of which are proven to have a surpassing performance on embedding knowledge graphs. Third, UKGE boosts its performance by applying probabilistic soft logic to create extra training samples based on predefined rules, which requires human labor and domain knowledge. We aim to design a training framework with no human intervention while having the same or better performance.

In response to the above issues, we propose PASSLEAF, a Pool-bAsed Semi-Supervised LEArning Framework for the uncertain knowledge graph embedding, completely freed from additional human labor. It consists of two parts, a confidence score prediction model that can adopt different types of existing embedding score functions for a given triplet, and the semi-supervised learning with both in-dataset positive samples and automatically generated negative samples with estimated confidence scores. PASSLEAF further maintains a sample pool to gather the information learned at different time steps.

Extensive experiment results on three open uncertain knowledge graph data sets show that, when compared with various existing knowledge graph embedding methods, PASSLEAF significantly reduces the impact of false-negative samples. We further justify the efficacy of the sample pool to accumulate past experiences acquired from different data. Also, we validate the advantage of an uncertain KG embedding method over a deterministic one in terms of preserving an uncertain graph structure. In the task of tail entity prediction, our model demonstrates up to 50 % reduction in weighted mean rank on CN15K compared to UKGE and about 4% improvement in nDCG on NL27K.

2 Preliminaries

2.1 Problem Statement

An *uncertain knowledge graph* models the triplets in association with a confidence score and can be denoted by $\{(h, r, t, c) \mid h, t \in E, r \in R, c \in [0, 1]\}$, where where E is the entity set, R is the relation set, and c is the corresponding confidence score. Obviously, a deterministic knowledge graph is a special case of an uncertain knowledge graph, where $c = 1$ for all (h, r, t) triplets. *Knowledge graph embedding* refers to learn low-dimensional vector representations of those entities and relations in $\mathbb{R}^k$ and $\mathbb{R}^l$ such that the original graph structure is preserved. The *uncertain knowledge graph embedding* task aims to preserve not only the graph structure but also the corresponding confidence score c of each triplet.

Suppose the embedding vectors of a triplet (h, r, t) is $(\bar{h}, \bar{r}, \bar{t})$. The basic concept of the most existing knowledge graph embedding model is to utilize a score function $S(\bar{h}, \bar{r}, \bar{t})$ to measure the plausibility of a triplet: higher value for the positive samples, which refers to the existing triplets in the training set (the given knowledge graph), and lower value for the negative samples, randomly picked from out-of-dataset triplets. Furthermore, the embedding model exploits a loss function to widen the margin between the estimated scores of positive and negative samples.

When embedding uncertain knowledge graphs, the model should be "confidence-aware". That is to say, the embedding method for uncertain knowledge graphs cannot merely differentiate positive samples from negative ones but need to rank them in line with their plausibility. More precisely, $S(\bar{h}, \bar{r}, \bar{t})$ should be positively correlated to the true confidence score of (h, r, t).

To our knowledge, UKGE (Chen et al. 2019) is the first and the only embedding method of uncertain knowledge graphs that is confidence-aware. UKGE utilizes a score mapping function S' to constrain the estimated score S in [0, 1] and applies mean squared loss to make the estimated score of a triplet close to its ground-truth confidence score.

$$S'(\bar{h}, \bar{r}, \bar{t}) = \frac{1}{1 + e^{-(b + wS(\bar{h}, \bar{r}, \bar{t}))}}. \tag{1}$$

The loss L_{neg} and L_{pos} for randomly drawn negative samples D_{neg} and in-dataset positive samples D_{pos}, respectively, are defined as follows.

$$L_{neg} = \sum_{(\bar{h}, \bar{r}, \bar{t}) \in D_{neg}} \left\| S'(\bar{h}, \bar{r}, \bar{t}) \right\|^2, \text{ and} \tag{2}$$

$$L_{pos} = \sum_{(\bar{h}, \bar{r}, \bar{t}, c) \in D_{pos}} \left\| S'(\bar{h}, \bar{r}, \bar{t}) - c \right\|^2. \tag{3}$$

Together, the overall objective is

$$\min L_{pos} + \frac{1}{N_{gen}} (L_{neg}). \tag{4}$$

Although UKGE successfully preserves the confidence score of triplets for embedding uncertain knowledge graphs, there are still some challenges to be conquered as we stated in Sec. 1. First, UKGE faces the *false-negative problem for training* as other deterministic graph embedding method, which treats the randomly drawn triplets out of the given dataset as negative samples with a confidence score of zero. In fact, those unseen samples may have a certain degree of plausibility. We believe the practice introduces more noise to uncertain KGs than to deterministic ones seeing that uncertain KGs are arguably denser in terms of both in-dataset and unseen triplets due to the low-confidence triplets. Our pool-based semi-supervised learning is to alleviate this issue. Second, the UKGE method is tailored only for one type of score function. To exploit recently proposed designs, we propose a generalized framework for both major types, semantic-based and translational distance based (more details please see Sec.2.2). Third, probabilistic soft logic in UKGE requires human-defined rules as an extra data augmentation instrument; in contrast, PASSLEAF aims to curtail human intervention.

2.2 Related Work

According to the design of the score function, deterministic KG embedding methods can be categorized into translational distance based and semantic-based (Wang et al. 2017). In translational distance based methods, a relation embedding is usually a transition or mapping for entity embeddings. The score function measures the distance between the mapped head entity and the tail entity of a triplet in the embedding space. Representative works include TransE (Bordes et al. 2013), TransH(Wang et al. 2014), TransR(Lin et al. 2015), and RotatE(Sun et al. 2019). For semantic-based methods, the score function evaluates the plausibility based on the latent semantics of entities given a triplet. Representative works include HolE (Nickel, Rosasco, and Poggio 2016), ConvE (Dettmers et al. 2018), DistMult (Yang et al. 2015), and ComplEx (Trouillon et al. 2016).

To our knowledge, UKGE (Chen et al. 2019) is the first and the only one to work on embedding uncertain knowledge graphs, which we have detailed in Sec. 2.1. Note that the report has also proved that UKGE significantly outperforms the method (Hu et al. 2017), which is originally designed for single relation graphs with uncertainties, on embedding uncertain knowledge graphs that are multi-relational. Therefore, in this work, we focus on comparing our work with UKGE.

Negative sampling is commonly used to augment training samples in KG embedding methods so that the model can distinguish the positive samples better. Usually, the negative samples are generated by replacing either the head or tail entity of an in-dataset triplet with a randomly chosen entity. KBGAN(Cai and Wang 2018) generates negative samples by a generative adversarial network that are more difficult for the embedding model to distinguish. RotatE advances this idea and designs a computationally efficient self-adversarial loss function. However, all the existing methods follow the same assumption that the unseen samples are negative, which is contrary to our mentioned *false-negative problem* in uncertain KGs. We cast doubt on the practice to zero the confidence score of all unseen samples.

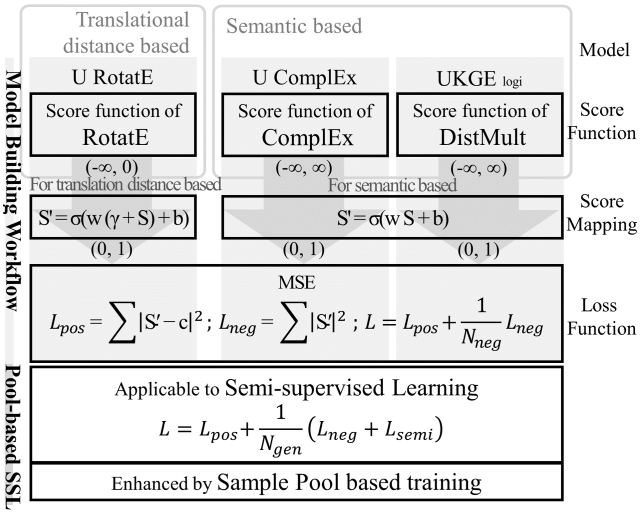

Figure 1: Framework Overview.

3 Methodology

3.1 The Framework Overview

Our proposed framework, PASSLEAF, is illustrated in Fig. 1. It consists of two main parts: the building of uncertainty-predicting models and the pool-based semi-supervised learning.

The goal of the uncertainty-predicting model is to adapt existing score functions of knowledge embedding to predict the confidence score of (h, r, t) samples. We design the corresponding score mapping function for both translational distance based and semantic based methods and equip it with a loss function to form a confidence-aware embedding model. The details are in Section 3.2.

The semi-supervised learning framework provides a better way of dealing with unseen samples to alleviate the false negative problem, not just treating unseen samples as negative samples but reevaluate their potential confidence score.

Moreover, we maintain a sample pool of the latest semi-supervised samples to exploit the past learning experience, which makes the embedding learning quality even better. The details are in Sec. 3.3 and Sec. 3.4.

3.2 The Model Building Workflow and Examples

Seeing that the mean square error (MSE) loss function of UKGE has shown satisfactory performance despite its simplicity, we further extended its score mapping function to support translational distance based methods.

The score function of UKGE is the same as DistMult, a semantic-based method, giving positive scores if a triplet is plausible and negative scores otherwise. Then, the sigmoid score mapping, (1), makes the final score a probability value.

However, in view of many recent works that outperform DistMult, it is desirable to take advantage of them. For example, ComplEx, a variant of DistMult, surpasses its predecessor in several datasets; RotatE, a translational-distance based method, is one of the state-of-the-art models. Therefore, a generalized workflow is highly desired to compose new uncertain KG embedding models ready for the semi-supervised training by incorporating more existing score functions. Nevertheless, solely replacing the score function of UKGE with the new ones can result in degraded performance. The score mapping works poorly for translational distance based score functions where the score is the negative value of a distance, ranging in $[-\infty, 0]$. As a result, to centralize the range of scores, we apply the following score mapping function with one additional hyper-parameter, γ.

$$S'(\bar{h}, \bar{r}, \bar{t}) = \frac{1}{1 + e^{-(b + w(\gamma + S(\bar{h}, \bar{r}, \bar{t})))}}. \quad (5)$$

To conclude, given a score function, PASSLEAF takes two steps to construct a new model. First, depending on whether the score function is semantic-based or translational distance based, the score function will be mapped by the mapping shown in equation (1) and (5) accordingly. Secondly, the score function and the MSE loss together form the model.

As examples, we build Uncertain ComplEx and Uncertain RotatE based on the score function of ComplEx and RotatE. The former is semantic-based and the later is translational distance based. The score function for Uncertain ComplEx is

$$S(\bar{h}, \bar{r}, \bar{t}) = \left\| \bar{h}\bar{t}\bar{r} \right\|^2, \text{where } \bar{h}, \bar{t}, \bar{r} \in \mathbb{C}^k. \quad (6)$$

The score function for Uncertain RotatE is

$$S(\bar{h}, \bar{r}, \bar{t}) = \left\| \bar{h}\bar{r} - \bar{t} \right\|^2, \text{where } \bar{h}, \bar{t}, \bar{r} \in \mathbb{C}^k; \|\bar{r}\| = 1. \quad (7)$$

Additionally, we designed a simplified Uncertain RotatE, denoted as U RotatE-, where the unit length constraint on relation embedding is relaxed. The model exhibits a more stable performance than Uncertain RotatE in our evaluations. The modified score function is as below.

$$S(\bar{h}, \bar{r}, \bar{t}) = \left\| \bar{h}\bar{r} - \bar{t} \right\|^2, \text{where } \bar{h}, \bar{t}, \bar{r} \in \mathbb{C}^k. \quad (8)$$

For both Uncertain RotatE and U RotatE-, we set $\gamma = 2$. These models will be used to verify our methodology in section 4. Please note that our framework is an extension to UKGE so one of its variations, $UKGE_{logi}$, is listed as one of the PASSLEAF models in the experiments.

3.3 The Semi-Supervised Learning

Despite the potential hazard to bring in false-negative samples, negative sampling as data augmentation can effectively complement the lack of negative triplets in most knowledge graphs. However, as stated in Preliminaries, we believe the number of potential false-negative tends to be larger in uncertain KGs. To solve the issue, we introduce semi-supervised samples.

Semi-supervised samples are picked in the same way as randomly drawn negative samples, by corrupting either the head or tail entity of an in-training-set triplet. The difference is that the confidence score of each semi-supervised sample will be estimated and specified by the current model instead of zeroing them. Hence, they can be either positive or negative. We believe this will mitigate the false-negative problem. For one thing, the importance of randomly drawn negative samples is diluted; for another, semi-supervised samples

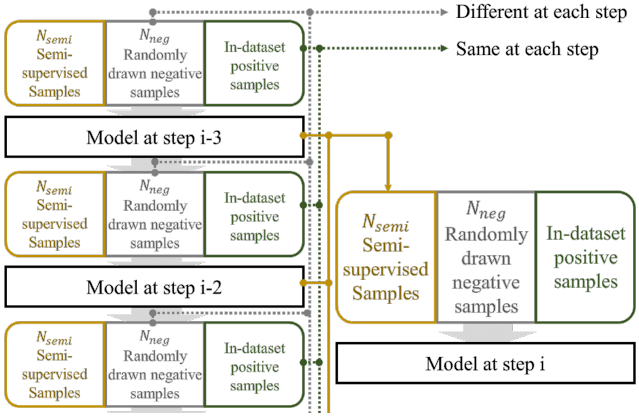

Figure 2: Sample pool: ensemble of paste experiences.

are expected to better estimate the real confidence score of unseen triplets. Furthermore, it serves as a data augmentation instrument, particularly for positive samples.

The MSE loss for semi-supervised samples, D_{semi}, is as follows.

$$L_{semi} = \sum_{(\bar{h},\bar{r},\bar{t},\bar{c})\in D_{semi}} \left\|\sigma\left(S(\bar{h},\bar{r},\bar{t})\right) - \bar{c}\right\|^2. \quad (9)$$

A mixture of negative and semi-supervised samples brings the best performance boost. Therefore, the overall loss function is modified as follows.

$$L = L_{pos} + \frac{1}{N_{gen}}\left(L_{semi} + L_{neg}\right). \quad (10)$$

Instead of training with semi-supervised samples generated at the previous step, we apply a sample pool as a relay for samples, explained in the next section.

For simplicity, randomly-drawn negative samples and semi-supervised samples are collectively called generated samples. The amount of generated samples per positive sample is a predefined hyper-parameter, *generated per positive*, so the total number of generated samples per training step, N_{gen}, is generated per positive $\times$ batch size. When the number of generated samples is the same as that in traditional negative sampling, the computational overhead is mainly from predicting the confidence score for newly-generated samples. In fact, even so, semi-supervised training outstrips pure negative sampling in terms of prediction accuracy, justified in the Experiments section.

3.4 The Sample Pool

To better exploit the benefit of training with past experiences, inspired by DQN(Mnih et al. 2013), PASSLEAF maintains a sample pool to keep C latest semi-supervised samples. For a training epoch i, there are two steps to take. First, $N_{new}(i)$ samples should be generated and stored into a sample pool. Second, $N_{semi}(i)$ samples are randomly fetched from the pool to train the model along with $N_{gen} - N_{semi}(i)$ randomly drawn negative samples according to the loss function shown in (10). To reduce computational overhead, a continuous band of samples in the pool will be selected instead of drawing them one by one.

dataset	entity	rel.	train	test	Avg.C.
PPI5K	4999	7	230929	21720	23.74
NL27K	27221	417	149100	14034	2.15
CN15K	15000	36	204984	19293	3.87
WN18RR	40559	11	86834	3133	1.38
FB15K237	14505	237	272114	20465	2.91

Table 1: Data statistics, including the # of entity and relation types, and # of triplets for training and testing, and the average candidates for the tail-entity prediction task

So far, the hyper-parameters for the sample pool are defined as functions for better generalizability. To simplify the design and limit the number of hyper-parameters, we design N_{new} as a step function with respect to time and N_{semi} to be a clipped linear function, which starts from zero and linearly increments until a given maximum. In that, no semi-supervised sample will be generated before the model is stable enough; also, the weight of semi-supervised samples will increase as the model accumulates more and more experiences. The formulas are as follows:

$$N_{new}(i) = \begin{cases} N_{gen}, & \text{if } i \geq T_{\text{NEW SEMI}} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$N_{semi}(i) = \begin{cases} \max\left(M_{\text{SEMI}}, \lfloor \alpha(i - T_{\text{SEMI TRAIN}}) \rfloor\right), & \\ & \text{if } i \geq T_{\text{SEMI TRAIN}} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$T_{\text{NEW SEMI}}$ and $T_{\text{SEMI TRAIN}}$ are the epoch to start generating semi-supervised sample and the one to begin fetching samples from the pool respectively; M_{SEMI} is the maximum amount of semi-supervised samples per step; α determines how long it takes for the number of semi-supervised samples to reach the maximum. Reasonably, $T_{\text{SEMI TRAIN}}$ must be greater than $T_{\text{NEW SEMI}}$ to accumulate an adequate amount of semi-supervised samples for training. Also, M_{SEMI} should not be the same or exceed the number of generated samples per step to reserve some quota for randomly drawn negative samples. The choice of these parameters depends on the model and the dataset, which we leave for future research.

The pool-based training can achieve a similar effect to a temporal ensemble, proposed by (Laine and Aila 2017), despite some fundamental differences in the methods. The prerequisite of temporal ensemble is that the training data should differ between training epochs. In (Laine and Aila 2017), noise augmented images is used. As illustrated in Figure 2, instead, we use negative samples randomly drawn at each step. Unlike (Laine and Aila 2017), our semi-supervised samples do not undergo binarization. Semi-supervised samples generated at different time steps retain experiences accumulated from seeing different randomly drawn negative samples. Therefore, the pool based design can be regarded as an ensemble of past models, which further strengthens the effectiveness of semi-supervised samples.

4 Experiments

In this section, we try to verify the following statements: Is our pool-based semi-supervised training more effective than pure negative sampling? Does the pool-based design boost the semi-supervised training? Does uncertain KG embedding methods outperform their deterministic counterpart on uncertain KGs?

4.1 Settings

Datasets. The summary of the datasets is in Table 1. We follow UKGE (Chen et al. 2019) and evaluate our models on three open uncertain knowledge graphs: PPI5K, a protein to protein interaction ontology from the STRING database (Szklarczyk et al. 2017), NL27K, a sub-graph of the NELL database(Mitchell et al. 2018), and CN15K, a subgraph of a multi-linguistic and commonsense KG, ConceptNet (Speer, Chin, and Havasi 2017). We follow UGKE to partition the dataset into 85% for training, 7%for validation, and 8% for testing. However, we apply no extra data filtering so the reported scores may differ from those of UKGE. In addition, we also use two popular deterministic KGs, WN18RR(Dettmers et al. 2018) and FB15K237(Toutanova and Chen 2015) for evaluations. *Avg.C.* indicates the average number of candidate tail entity for a given set of head entity and relation. Consistent with our belief, the uncertain datasets we used are slightly denser than the commonly used deterministic datasets in terms of the *Avg.C.* value. Specifically, *Avg.C.* of PPI5K is more than 20.

Tasks and Metrics. We use two tasks for evaluation, confidence score prediction and tail entity prediction.

Confidence score prediction (CSP) is to predict the confidence score given a triplet, requiring the model to be *uncertainty-predicting*. This experiment helps discern the performance differences in positive and negative triplets. For the former, the ground-truth data is in-dataset samples. For the later, we use randomly drawn triplets given zero confidence scores. This is contradictory to our belief but labeling unseen samples is costly and labor-intensive. Nevertheless, the metric is a viable indicator of excessive false-positives when the value is high. We calculate mean squared error (MSE) for both.

Tail entity prediction(TEP) is a conventional evaluation task for knowledge graph embedding. The goal is to predict the tail entities given a head entity and a relation. On uncertain KG data, the task is no longer a hit-or-miss classification but a ranking job to order candidates according to their true plausibility. Therefore, our metrics, including Hit@K, mean rank, and nDCG, are linearly weighted by the confidence score and denoted by WH@K, WMR, and nDCG, respectively. As an example, for a testing data set, D, the WMR is as follows:

$$\text{WMR} = \frac{\sum_{(h,r,t,c)\in D} c \cdot \text{rank}_{(h,r,t)}}{\sum_{(h,r,t,c)\in D} c}, \tag{13}$$

where rank is the predicted rank of a triplet. Linear weighted nDCG is as follows:

$$nDCG = \frac{\sum_{(h,r,t,c)\in D} \frac{c}{\log_2(\text{rank}_{(h,r,t)}+1)}}{\sum_{(h,r,t,c)\in D} \frac{c}{\log_2(\text{rank}'_{(h,r,t)}+1)}}, \tag{14}$$

models	WMR	WH@20	WH@40	nDCG
		CN15K		
$UKGE_{logi}$	1676.0	32.1%	38.5%	29.7%
+ SS	1326.3	34.2%	41.3%	30.4%
U ComplEx	1791.9	31.7%	38.3%	29.6%
+ SS	1229.2	35.4%	42.5%	**30.8%**
U RotatE	1017.2	34.9%	43.2%	27.8%
+ SS	**866.2**	**35.5%**	**44.0%**	28.4%
U RotatE-	1031.3	34.7%	41.8%	29.0%
+ SS	949.4	35.3%	42.6%	30.3%
		NL27K		
$UKGE_{logi}$	288.57	70.42%	76.77%	71.65%
+ SS	242.28	71.78%	77.85%	74.52%
U ComplEx	296.28	70.27%	76.78%	71.67%
+ SS	223.64	**72.03%**	**78.42%**	**75.31%**
U RotatE	493.61	63.62%	70.83%	63.32%
+ SS	438.32	60.13%	67.06%	60.41%
U RotatE-	451.445	69.72%	75.71%	70.97%
+ SS	**206.09**	69.97%	76.11%	74.45%
		PPI5K		
$UKGE_{logi}$	38.59	42.64%	68.81%	43.87%
+ SS	**34.89**	45.06%	**70.57%**	**44.51%**
U ComplEx	38.8	42.41%	68.32%	43.49%
+ SS	35.53	**45.16%**	69.85%	43.93%
U RotatE	40.44	44.71%	70.52%	43.16%
+ SS	40.9	43.41%	68.23%	41.21%
U RotatE-	49.63	41.10%	68.38%	42.04%
+ SS	35.58	44.81%	69.39%	43.98%

Table 2: Tail entity prediction

where $rank'$ is the true rank of a triplet. We exclude candidates from the training set to prevent the testing set from being dominated by seen data, resulting in different values than that from UKGE. However, training data are not excluded from the calculation of ranking so the randomly assigned order to break ties may influence the resulting value. Additionally, for Hit@K, to prevent the number of positive candidates from exceeding the value K in dense graphs, we choose larger Ks, 20 and 40.

Models. To demonstrate the generalizability, we test several score functions for PASSLEAF and also one of the variants of UKGE, $UKGE_{logi}$, that performs best in TEP. Semantic-based and translational distance based models are exemplified by Uncertain ComplEx and Uncertain RotatE. U RotatE-, (8), is included for its stability.

Hyper-Parameters. The sample pool introduces additional hyper-parameters. Hence, to simplify the experiment, we choose a fixed set of hyper-parameters seemingly reasonable for all datasets and models. We use Adam optimizer and a fixed batch size of 512. We explore the dimensions of embedding vectors in 256, 512, 1024. Usually, the larger dimensions result in better performance in most metrics. However, to reduce the computational overhead, we fix this value at 512, a commonly used embedding size. Settings for the sample pool are not fine-tuned except for M_{SEMI}. $T_{\text{NEW SEMI}}$ is 20; $T_{\text{SEMI TRAIN}}$ is 30; M_{SEMI}

datasets	CN15K		NL27K		PPI5K	
models	pos	neg	pos	neg	pos	neg
$UKGE_{logi}$	28.2	0.17	7.9	**0.32**	0.76	0.28
+ SS	23.8	0.36	5.5	0.38	0.51	0.30
U ComplEx	30.3	**0.13**	8.0	0.38	0.78	0.28
+ SS	23.9	0.31	4.5	0.42	0.62	0.31
U RotatE	19.0	1.09	4.5	0.72	0.44	0.29
+ SS	**14.4**	8.28	**3.0**	1.42	**0.35**	0.36
U RotatE-	25.6	0.27	7.4	0.36	0.58	0.29
+ SS	22.7	0.34	6.5	0.37	0.62	0.29

Table 3: Confidence score prediction. (In 0.01)

Head	Rel	Tail	Conf
algeria	part of	africa	81.81%
tunisia	part of	africa	77.83%
croatia	part of	europe	82.06%
harpsichord	used for	play music	85.03%
harpsichord	is a	instrument of music	75.88%
court	is a	place	90.55%
court	used for	judge	78.75%

Table 4: Found missing triplets in CN15K

is $0.8 \times$ batch size; pool capacity, C, is 10000000 triples; $alpha$ is 0.02; *generated per positive*, is 10. The implementation is based on Tensorflow 1.14 on a CentOS-7 machine with 48 Core Xeon processors and Tesla P100-SXM2.

4.2 Effectiveness of Semi-Supervised Samples

First of all, we would like to verify whether the pool-based semi-supervised training is more effective than traditional negative sampling. Uncertain KG embedding models with and without applying the pool-based semi-supervised training are examined and juxtaposed over three uncertain knowledge graphs. For fairness, the number of generated samples per step is the same for all models, with/without semi-supervised learning. We choose the best models according to the validation hit@20 for the TEP task and MSE on positive triplets for the CSP task. We will elaborate on results for TEP, CSP, worst case for CSP, and missing triplet discovery in order.

Tail Entity Prediction The results of TEP on three datasets are shown in Table 2. On all the datasets, models with semi-supervised samples consistently make improvements in all metrics. Especially, on NL27K, with pool-based semi-supervised training, WMR of U RotatE- reduces by about 50% and nDCG of U ComplEx improves by almost 4%. Also, on CN15K, WMR of the best model with semi-supervised training is nearly half of that of $UKGE_{logi}$ without semi-supervised training.

In terms of the models, Uncertain RotatE performs worst except in CN15K while Simplified Uncertain RotatE has consistent performance across all datasets. One possible reason is that our experiment uses a fixed dimension for all models and the dimension of the relation embeddings of RotatE is 256, half of that of other models due to its unit length constraint.

datasets	CN15K		NL27K		PPI5K	
M_{SEMI}*	pos	neg	pos	neg	pos	neg
0 (no SS)	30.3	0.13	8.0	0.38	0.78	0.28
0.8 (def)	23.9	0.31	4.5	0.42	0.6	0.31
1.0	8.8	8.98	1.6	2.39	0.51	1.15

Table 5: False-positive worst case analysis: CSP. (In 0.01)

datasets	CN15K		NL27K		PPI5K	
models	pos	neg	pos	neg	pos	neg
U ComplEx	30.3	0.13	8.0	0.38	0.78	0.28
+ SS_{-pool}	28.4	0.20	7.5	0.35	0.78	0.30
+ SS	23.9	0.31	4.5	0.42	0.62	0.31

Table 6: Ablation test: CSP. (In 0.01)

Confidence Score Prediction With satisfactory performance in ranking samples properly according to the plausibility, we proceed to verify the individual performances on positive and negative samples. The results of CSP are in Table 3. *pos* and *neg* are the MSE on in-dataset positive samples and randomly drawn negative samples, respectively. +SS indicates the model after applying semi-supervised training to the model in the previous row.

MSEs on in-dataset positive samples improve by more than one fifth across all models and datasets after applying semi-supervised samples. The result supports the idea that pool-based semi-supervised training alleviates the noises brought in by the false-negative samples and further improves the prediction accuracy on unseen in-dataset positive samples.

On the other hand, the MSE values on negative samples increase slightly after training with semi-supervised samples. Still, the result is expected since the ground-truth of this measurement is randomly drawn negative samples, which we believe is prone to false-negatives. In fact, a slightly higher MSE value can imply the model is capable of detecting false-negative samples. Undeniably, this result raises the concern to cause excessive false-positives predictions. Nevertheless, The performance in TEP lifts the concern. Arguably, this indicates that its potential impact of false-positives is outstripped by its merits to avoid false-negative samples and the ensemble of past experiences. To further support this argument, we have another experiment to find bounds for the MSE on negative samples.

Upper Bound for False-Positives This extended experiment is to find an upper bound for potential false-positives. Uncertain ComplEx under different maximum semi-supervised samples per step, M_{SEMI}, is tested. The results of CSP are shown in Table5 individually. The values in the first column indicate the proportion of M_{SEMI} in the number of generated samples per step. The default is 0.8 and a value of 0 implies no semi-supervised sample. Contrarily, under $M_{SEMI} = 1.0$, no randomly drawn negative sample will be used after the given training step, which is the most extreme case and prone to false-positives. Under the setting, the MSEs on negative samples are an appropriate estimation of the upper bound for false-positives.

datasets	CN15K			NL27K			PPI5K		
models	WMR	WH@20	nDCG	WMR	WH@20	nDCG	WMR	WH@40	nDCG
U ComplEx	1791.9017	31.67 %	29.56 %	296.2822	70.27 %	71.67 %	38.7985	68.32 %	43.49 %
+ SS$_{-pool}$	1870.1333	32.85 %	29.86 %	291.6816	70.55 %	72.14 %	37.1732	68.34 %	43.81 %
+ SS	1229.1969	35.39 %	30.78 %	223.64	72.03 %	75.31 %	35.5303	69.85 %	43.93 %

Table 7: Ablation test: Tail entity prediction

		CN15K			NL27K			PPI5K		
threshold	model	WMR	WH@20	nDCG	WMR	WH@20	nDCG	WMR	WH@40	nDCG
0.3	ComplEx	**1,425.07**	**38.66%**	**31.28%**	**222.66**	68.64%	69.62%	34.00	68.44%	40.83%
0.3	U C, + SS	1,458.79	35.23%	30.31%	223.35	**72.05%**	**75.37%**	**33.08**	**72.74%**	**43.82%**
0.5	ComplEx	1,487.14	**39.41%**	**31.65%**	229.52	69.97%	71.59%	**25.39**	81.62%	44.04%
0.5	U C. + SS	**1,236.50**	37.39%	31.47%	**201.95**	**72.75%**	**76.75%**	27.18	**84.76%**	**46.54%**
0.7	ComplEx	1,623.06	**38.21%**	**28.76%**	291.18	69.88%	71.10%	22.68	91.70%	47.31%
0.7	U C. + SS	**1,490.07**	35.17%	28.36%	**201.67**	**72.73%**	**76.84%**	**21.31**	**93.34%**	**48.51%**

Table 8: Deterministic vs uncertain KG embedding

As the results indicate, even under the extreme case, the MSEs on negative samples are still controlled, suggesting acceptable upper bounds. In addition, there is a considerable margin between our default setting and the worst case.

Case Study Finally, we use Uncertain ComplEx as an example to show some missing triplets found by PASSLEAF models. Missing triplets refer to unseen triplets absent from the dataset, requiring human verification. Table 4 shows some plausible triplets found by Uncertain ComplEx with semi-supervised training on CN15K. None of them is found without applying semi-supervised samples. Although false-positive predictions are seemingly much, a far greater number of missing triplets are uncovered with pool-based semi-supervised training.

4.3 Ablation Test of the Sample Pool

To evaluate the contributions that the sample pool makes to the improvements, we compare the pool-based semi-supervised training to a naive approach with no sample pool, in which semi-supervised samples are generated at the previous step. Both model have the same number of N_{semi}. The baselines are an ablated model without any semi-supervised sample and the naive approach. The results of CSP and TEP are shown in Table 6 and Table 7 respectively.

The naive approach makes limited improvements on most metrics while the pooled-based approach achieves a more prominent boost. For example, in confidence score prediction MSE on CN15K, the naive approach contributes a 2% reduction while the pooled-based method has more than 6% We deduce that the improvements by the naive approach are mainly due to the reduced importance of false-negative samples and the avoidance of drawing them. For the pooled-based method, we attribute to the ensemble of several models trained with varied data as stated in Sec. 3.4.

4.4 Performance of Deterministic Knowledge Graph Embedding Methods

The third statement we try to prove is that models tailored for uncertain knowledge graphs do outstrip deterministic models on uncertain KGs. In this test, we compare PASSLEAF methods with their *deterministic counterpart*, meaning the corresponding deterministic KG embedding model where the score function originated. Applying deterministic KG methods requires binarization. So, separate models are trained for each threshold while there is only one uncertain embedding model. The results under several binarization thresholds are shown in Table 8. For simplicity, only Uncertain ComplEx and ComplEx are shown.

Uncertain ComplEx consistently outperforms ComplEx, whose loss function is more complicated, on NL27K and PPI5K in most metrics and thresholds, buttressing the notion that PASSLEAF models can handle uncertainty better. To find the contributing factor, we analyzed the improvements made by Uncertain ComplEx over ComplEx with respect to thresholds. As the threshold rises, both models improve. Also, the gap between their performance remains relatively constant in WH@K and nDCG except on PPI5K. However, in WMR, the gap widens as the threshold increases on most datasets. We credit this to the retained low-confidence triplets for the PASSLEAF models as an additional source of information. As for the significant increase in WH@40 on PPI5K as the threshold rises, we tend not to over-explain it because the metric can be strongly influenced by the decrease in the number of candidates.

5 Conclusion and Future Work

PASSLEAF generalizes the process of building uncertain KG embedding models and boost the performance by avoiding false-negative samples and by ensembling experiences learned at previous time steps. We left the design of sample size functions for the sample pool, the choice of hyperparameters, and more sophisticated loss functions for future studies. Also, we believe our idea may benefit deterministic KGs as well, which is another topic worth studying.

Acknowledgments

This study was supported in part by the Ministry of Science and Technology (MOST) of Taiwan, R.O.C., under Contracts 106-3114-E-002-008, 107-2221-E-001-009-MY3, 108-2218-E-002-048, and 108-2221-E-002-062-MY3.

References

Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, 2787–2795. Red Hook, NY, USA: Curran Associates Inc.

Cai, L.; and Wang, W. Y. 2018. KBGAN: Adversarial Learning for Knowledge Graph Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1470–1480. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1133. URL https://www.aclweb.org/anthology/N18-1133.

Chen, X.; Chen, M.; Shi, W.; Sun, Y.; and Zaniolo, C. 2019. Embedding Uncertain Knowledge Graphs. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*.

Dettmers, T.; Pasquale, M.; Pontus, S.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, 1811–1818. URL https://arxiv.org/abs/1707.01476.

Hu, J.; Cheng, R.; Huang, Z.; Fang, Y.; and Luo, S. 2017. On Embedding Uncertain Graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, 157–166. New York, NY, USA: Association for Computing Machinery. ISBN 9781450349185. doi:10.1145/3132847.3132885. URL https://doi.org/10.1145/3132847.3132885.

Laine, S.; and Aila, T. 2017. Temporal Ensembling for Semi-Supervised Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL https://openreview.net/forum?id=BJ6oOfqge.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2181–2187. AAAI Press. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; and Welling, J. 2018. Never-Ending Learning. *Commun. ACM* 61(5): 103–115. ISSN 0001-0782. doi:10.1145/3191513. URL https://doi.org/10.1145/3191513.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari With Deep Reinforcement Learning. In *NIPS Deep Learning Workshop*.

Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic Embeddings of Knowledge Graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 1955–1961. AAAI Press.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 4444–4451. AAAI Press.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

Szklarczyk, D.; Morris, J. H.; Cook, H.; Kuhn, M.; Wyder, S.; Simonovic, M.; Santos, A.; Doncheva, N.; Roth, A.; and Bork, P. 2017. The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Research* ISSN D362-D368. doi:10.1093/nar/gkw937.

Toutanova, K.; and Chen, D. 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*, 57–66. Beijing, China: Association for Computational Linguistics. doi:10.18653/v1/W15-4007. URL https://www.aclweb.org/anthology/W15-4007.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, volume 48, 2071–2080.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29(12): 2724–2743.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In Brodley, C. E.; and Stone, P., eds., *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, 1112–1119. AAAI Press. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531.

Yang, B.; Yih, S. W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*. URL https://www.microsoft.com/en-us/research/publication/embedding-entities-and-relations-for-learning-and-inference-in-knowledge-bases/.