Confidence-Aware Negative Sampling Method for Noisy Knowledge Graph Embedding

Yingchun Shan, Chenyang Bu*, Xiaojian Liu, Shengwei Ji, Lei Li
School of Computer Science and Information Engineering,
Hefei University of Technology, Hefei, China
yingcshan@gmail.com, chenyangbu@hfut.edu.cn, {liuxiaojian, swji}@mail.hfut.edu.cn, lilei@hfut.edu.cn

***Abstract*—Knowledge graph embedding (KGE) can benefit a variety of downstream tasks, such as link prediction and relation extraction, and has therefore quickly gained much attention. However, most conventional embedding models assume that all triple facts share the same confidence without any noise, which is inappropriate. In fact, many noises and conflicts can be brought into a knowledge graph (KG) because of both the automatic construction process and data quality problems. Fortunately, the novel confidence-aware knowledge representation learning (CKRL) framework was proposed, to incorporate triple confidence into translation-based models for KGE. Though effective at detecting noises, with uniform negative sampling methods, and a harsh triple quality function, CKRL could easily cause zero loss problems and false detection issues. To address these problems, we introduce the concept of negative triple confidence and propose a confidence-aware negative sampling method to support the training of CKRL in noisy KGs. We evaluate our model on the knowledge graph completion task. Experimental results demonstrate that the idea of introducing negative triple confidence can greatly facilitate performance improvement in this task, which confirms the capability of our model in noisy knowledge representation learning (NKRL).**

Keywords—knowledge graph embedding, representation learning, noise detection

I. Introduction

In recent years, with the vigorous development of knowledge graphs (KG), many KGs have been constructed, such as Freebase [1], WordNet [2] and DBpedia [3], and successfully applied to many real-world applications, from named entity disambiguation [4], to question answering [5, 6]. A KG is typically represented as multi-relational data with numerous triple facts in the form of (head entity, relation, tail entity), abridged as (h, r, t), indicating that h and t are connected by a specific relation r, e.g., (ChristopherNolan, DirectorOf, TheDarkKnight). Though effective in representing structured data, the underlying symbolic nature of such triplets usually makes KGs hard to manipulate [7], which can lead to data sparseness and computational efficiency problems.

To address these issues, a new research direction known as knowledge representation learning (KRL) or knowledge graph embedding (KGE) was proposed, and has quickly gained much attention [7, 8]. The goal of KRL is to embed components of a KG, including entities and relations, into dense, low-dimensional, and real-valued vectors, so as to preserve the intrinsic structural information of the KG, while calculating the complex semantic associations between entities and relations. This is of great significance to the construction, reasoning and application of KG.

The process of KG construction usually involves much human supervision, which is laborious and time consuming [9]. To alleviate this, automation and crowd-sourcing have been introduced to knowledge construction, although these approaches may suffer from possible noises and conflicts, due to insufficient human supervision. Because most conventional KRL methods assume that the knowledge in existing KGs is completely correct, these models will bring potential errors when applied to downstream tasks, such as KG completion [8, 10]. How to create the embedding models with noisy knowledge representation learning (NKRL), and detect possible mistakes, thus becomes an important issue.

To date, TransE [8] has been a representative KRL model. However, TransE and its extensions failed to consider the possible noises in KG, hence these models did not have noise detection ability. Thus, Liu et al. [9] proposed a novel confidence-aware knowledge representation learning (CKRL) framework, which could detect possible noises, and improve the performance of TransE. However, as an extension of TransE, the training process of CKRL required the participation of negative triples, which may easily lead to the zero loss problem [11]. Moreover, the quality of randomly generated negative triples is often very poor, and low quality negatives usually help little in training the true triplets, and can even slow down the convergence [12], leading to low noise detection ability and undesired accuracy in embedding models. Furthermore, the definition of triple quality function may also cause false detection problems, in which known facts are regarded as noisy data, or potential noises are regarded as true facts.

In this paper, we have concentrated on how to generate high quality, negative triplets, and used these triplets to assist model

*Chenyang Bu is the corresponding author. This work was partly supported by the National Key Research and Development Program of China, under grant 2016YFB1000901. Chenyang Bu was also partly supported by the National Natural Science Foundation of China (No. 61573327), the Project funded by the China Postdoctoral Science Foundation (No. 2018M630704), and the Fundamental Research Funds for the Central Universities (No. JZ2018HGBH0279).

978-1-5386-9125-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICBK.2018.00013

training in the detection of possible noises and conflicts located in existing KGs, while simultaneously constructing noise-free knowledge representations. For this reason, we introduced the concept of negative triple confidence to measure the quality of negative triples. Based on this concept, we propose a novel confidence-aware negative sampling method, which can deal with the zero-loss problem. Furthermore, we applied this concept for noise detection, which can alleviate the false detection problem.

In experiments, we evaluated our approach on two common KGE datasets (FB15K and WN18 [8]), with different noise ratios, and used the confidence-aware negative sampling method to train the improved CKRL model. Experimental results demonstrated that our method achieved better performance than the original model, in link prediction tasks, which further proved that considering negative triple confidence could greatly improve the noise detection ability of embedding models. We therefore considered that our contributions were three-fold:

- We proposed the concept of negative triple confidence, and used this concept to select high quality, negative triplets, for translation-based model training.
- We used negative triple confidence to redefine the triple quality function in the CKRL framework, and improved the capability of knowledge representation in noise detection.
- Considering negative triple confidence, our method showed significant performance with common datasets with different noise ratios.

The remainder of the paper is structured as follows. Section II introduces related work. The motivation of this study and the details of the proposed algorithm are given in Section III. The performance of our model is demonstrated in Section IV. Section V concludes this study and introduces future work.

II. Related Work

In this section, firstly, related work on noise detection for knowledge graphs is introduced, and then translation-based methods are described.

A. Noise Detection for Knowledge Graph

It is inevitable that noises exist in large-scale KGs, and so the study of noise detection for KGs has begun to receive attention. For instance, a novel task named Wikidata vandalism was used to detect deliberate destruction in KGs [13]; DBpedia created its mappings to Wikipedia info boxes via a worldwide crowd-sourcing effort [3], and a crowd-sourced, human curation software, in which contributors can reject or approve a statement, was also adopted in Wikidata [14]. These noise detection tasks, in large-scale KGs, usually involve much human effort, and are therefore time consuming and laborious.

For this reason, researchers have focused on automatic KG noise detection [15], however most existing methods have concentrated mainly on feature selection from contents, users, items, and revisions [16]. Recently, Liu et al. [9] proposed a novel, confidence-aware framework, by considering the confidence of triple facts - to discover potential noises or conflicts, and to learn better embedding models, simultaneously. However, the triple quality function defined by CKRL was too harsh, and easily caused false detection problem (detailed in Section III).

B. Translation-Based Methods

Recently, effort has been concentrated on learning distributed representations for KGs, among which the translation-based methods (TransE [8] and its extensions) are both straightforward and effective, with state-of-the-art performance. The aim of these models was to embed both entities and relations into a continuous, low-dimensional vector space, interpreting relations as translating operations between head and tail entities.

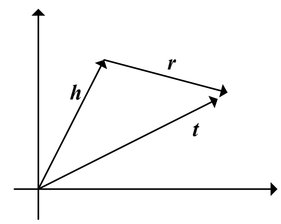


Fig. 1. Illustration of theTransE model (reproduced from [10])

Fig. 1 is a brief illustration of the TransE model, in which the translation assumption of TransE requires that embeddings should satisfy the equation $h + r \approx t$ for each fact (h, r, t). The score function used by TransE for a triplet (h, r, t) is defined as

$$f(h, r, t) = \| h + r - t\|_{1/2} \quad (1)$$

where $\|x\|_{1/2}$ represents the Manhattan or Euclidean distance of the vector x. The score is expected to be small if (h, r, t) holds. Namely, the embedding of the tail entity t should be close to the embedding of the head entity h, plus the embedding of the relationship r. To learn such embeddings and enhance the distinguishing ability, TransE takes a pairwise ranking loss function as its optimizing objective, which is given as follows:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h,r,t)}} [\gamma + f(h,r,t) - f(h',r,t')]_+ \quad (2)$$

where $[x]_+$ denotes the positive part of x, γ is a margin hyperparameter, and

$$S'_{(h,r,t)} = \{(h',r,t) \mid h' \in E\} \cup \{(h,r,t') \mid t' \in E\} \quad (3)$$

The set of negatives, $S'_{(h, r, t)}$, is constructed by corrupting either the head or tail of the triplets with a random entity from **E**. The loss function favors lower values of the score for training triplets, than for corrupted triplets.

Compared to traditional methods, the TransE model was more effective and efficient, while the translation assumption was over-simplified, to deal with 1-to-N, N-to-1, or N-to-N relations. TransH [10] proposed to project the entity to the relation-specific hyperplane w_r and thus an entity could have different representations, depending on what relation evolved. In detail, TransH transformed the entity embeddings as

$$h_{\perp} = h - w_r^{\mathrm{T}} h w_r \,,\; t_{\perp} = t - w_r^{\mathrm{T}} t w_r \tag{4}$$

with the score function defined as

$$f(h,r,t) = \| h_{\perp} + r - t_{\perp} \|_{1/2} \tag{5}$$

Both TransE and TransH assumed that entities and relations were in the same vector space. Then TransR [17], and TransD [18], extended TransE, by projecting the embedding vectors of entities into various spaces.

ManifoldE [19] embedded a triple as a manifold rather than a point. GTrans [20] used dynamic and static weighting strategies to describe entities, and also incorporated a dynamic relation space for each relation, which not only enabled the flexibility of an embedding model, but also reduced the noise from other relation spaces. puTransE [21] generated multiple embedding spaces, through a semantically and structurally aware triplet selection scheme. CKRL [9] extended TransE by introducing triple confidence, which enabled the capability of noises detection for KGE.

All the above knowledge-embedding models constructed negative triplets by random sampling, and were trained to separate the scores between positive and negative triplets. In addition, most of these studies assumed that all triples in KG were correct, which was inappropriate, especially for those KGs constructed automatically, with less human supervision. In this paper, we describe our proposal to incorporate negative triple confidence, for better negative sampling in CKRL.

III. Proposed Algorithm

A. Motivation

Most existing translation-based models have not taken potential noises or conflicts in KG into consideration, and therefore do not have the ability to detect noises. Thus, Liu et al. [9] proposed the CKRL framework, to discover mistakes in KG. However, CKRL has the following two issues.

1) Zero Loss Problem of CKRL

The negative triplets used in the CKRL model are constructed by replacing either the head or tail entities, of positive triplets, with entities randomly sampled from the whole set of entities **E**. Such a random sampling method is effective at the beginning of the training, because most of the negatives are still within the margin of the positive triplets. However, if corrupted triplets are still generated by randomly sampling, as the training process proceeds, they are very likely to be outside of the margin, which leads to a zero loss problem [11].

The zero loss problem refers to the scores (calculated based on Eq. 1) of those randomly generated negative triplets are out of the margin (i.e., the parameter γ in Eq. 2) to the scores of positive triplets, and would bring about zero loss in Eq. (2) after training for a period. These negative triplets make no contribution to improving the embeddings. Thus, such a sampling method would cause very slow convergence [11], which could further result in reduced accuracy, and in a lowered capability of noise detection in the embedding model.

2) False Detection Problem of CKRL

In order to test whether a triplet is noisy or not, CKRL defined a triple quality function, based on the pairwise ranking loss function; in this function, the quality of a triplet was calculated by applying Eq. (6).

$$Q(h, r, t) = -(\gamma + f(h, r, t) - f(h', r, t')) \tag{6}$$

In Eq. (6), the triple with a low value of $Q(h, r, t)$ was regarded as a possible noisy triple. The judged quality, based on $Q(h, r, t)$ for each triple, was related to the score of the selected negative triple (i.e., $f(h', r, t')$). That is, according to (6), if the score of the corrupted triplet, i.e., $f(h', r, t')$, was very high, the triple was more likely to be regarded as a convincing fact.

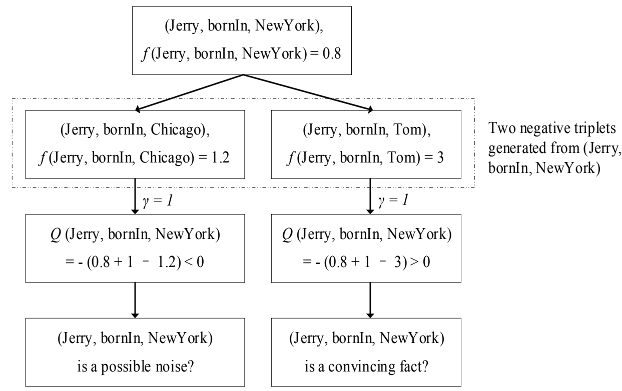


Fig. 2. Illustration of the false detection problem on a triple (Jerry, bornIn, NewYork).

Thus, the triple quality function adopted in CKRL may lead to the false detection problem, as illustrated in Fig. 2. For instance, given a triple (Jerry, bornIn, NewYork), assume that two negative triplets (Jerry, bornIn, Chicago) and (Jerry, bornIn, Tom) were generated by the current negative sampling method. Because the tail entity of (Jerry, bornIn, Tom) is not a location, the negative triplet (Jerry, bornIn, Tom) has a lower quality than the negative triplet (Jerry, bornIn, Chicago). Namely, f(Jerry, bornIn, Tom) is probably larger than f(Jerry, bornIn, Chicago), based on the translation assumption (i.e., Eq. 1). Thus, based on Eq. (6), the quality value of Q(Jerry, bornIn, NewYork) calculated based on the selected negative triplet (Jerry, bornIn, Tom) could be regared as more convincing than the value calculated based on the negative triplet (Jerry, bornIn, Chicago).

Based on the above analysis, we proposed to consider the confidence of negative triplets. We therefore selected quality negative triplets, with high confidence, for training the model, and used the concept of negative triple confidence to redefine the triple quality function, so as to improve its noise detection ability and accuracy.

B. Details of the Proposed Algorithm

In this section, the concept of negative triple confidence is first introduced to measure the quality of a negative triplet, which can be used to generate high quality negative triplets as much as possible. Based on this concept, a modified negative sampling method is described, which can be used to deal with the zero loss problem. Then we applied the negative triplet confidence for noise detection, which can alleviate the false detection problem. Details can be found in Algorithm 1.

Algorithm 1. Confidence-aware negative sampling for noise detection.	
Data: Training set of positive fact triples S = {(h, r, t)}, and noise triples S_{noise} = {(h, r, t)}	
Input: Pre-trained TransE model with parameters margin γ, embeddings dimension k, learning rate λ	
Output: TransE model with noise detection ability in KG	
1:**initialize**	
2:	$LT(h, r, t) \leftarrow 1$ for each triple $(h, r, t) \in S$ //local triple confidence initialized to 1
3: **loop**	
4:	Sample a mini-batch of data S_{batch} from S
5:	$T_{batch} \leftarrow \varnothing$ // initialize the set of pairs of triplets
6:	**for** $(h, r, t) \in S_{batch}$ **do**
7:	$T_{NG} \leftarrow$ randomly **sample *NG* negative triples**
8:	Calculate each **negative triple's quality** $NQ(h', r, t')$ **and confidence** $NC(h', r, t')$ in T_{NG}
9:	**Sample one negative triple** (h', r, t') with the **highest confidence** from T_{NG}
10:	Obtain each **triple's quality** according to $PQ(h,r,t) = -(\gamma + f(h,r,t) - \sum_{(h',r,t') \in T_{NG}} NC(h',r,t') f(h',r,t'))$
11:	$T_{batch} \leftarrow T_{batch} \cup \{(h, r, t), (h', r, t')\} \cup \{(h, r, t), (h, r', t)\}$
12:	Update embeddings w.r.t $\sum_{((h,r,t),(h',r',t')) \in T_{batch}} \nabla[\gamma + f(h,r,t) - f(h',r',t')]_+ LT(h,r,t)$
13:	Update $LT(h, r, t)$
14:	**end for**
15: **end loop**	

1) Confidence-aware Negative Sampling

Because a high quality negative triple is more likely to satisfy the translational rule than a low quality negative triple, the quality value of a negative triple (h', r, t') is first defined as follows, where a higher $NQ(h', r, t')$ indicates that the negative triple has a higher quality.

$$NQ(h',r,t') = -f(h',r,t') \tag{7}$$

Then the confidence of each negative triple is calculated based on the softmax function. Here, the softmax function is used to calculate the probability distribution of a group of candidate negative triplets. The confidence of a negative triple (h', r, t') is defined as follows.

$$NC(h',r,t') = \frac{\exp NQ(h',r,t')}{\sum_{(h_i',r_i,t_i') \in T_{NG}} \exp NQ(h_i',r_i,t_i')} \tag{8}$$

Where $NQ(h', r, t')$ is the quality of the negative triplet (i.e., Eq. 7); T_{NG} is the set of candidate negative triplets, where h' and t' are randomly selected from the whole set of entities to replace h or t; and $(h', r, t') \in T_{NG}$. Ideally, the set of negative candidates T_{NG} should contain all possible negatives. However, considering the time complexity, the size of the set T_{NG} is limited to a pre-defined parameter NG.

Based on the concept of negative triple confidence defined above, for a positive fact (h, r, t), the process of generating a negative triple with a high quality is given as follows. First, choose the position of a triplet to be replaced (i.e., the head entity or the tail entity) with a specific probability. Here the uniform random method is adopted. Second, generate a set of negative candidates (i.e., T_{NG}), and calculate the confidence of each negative triple according to Eq. (8). Finally, select the negative triplet with the highest confidence from T_{NG} for training use. The details are described in Steps 7-9 of Algorithm 1.

The modified negative sampling method with the concept of negative triple confidence can alleviate the zero loss problem of translation-based models. It is because the negative triplet with the highest quality value is selected from the group of candidates, instead of using a single negative triple generated randomly.

2) Negative Triple Confidence in Noise Detection

To adddress false detection problems, the triple quality function is redefined as Eq. (9), instead of using Eq. (6). Here the expectation of a group of candidates' scores is adopted, instead of a single negative triple's score. Thus, the results of the noise detection algorithm could be more stable.

$$PQ(h,r,t) = -(\gamma + f(h,r,t) - \sum_{(h',r,t') \in T_{NG}} NC(h',r,t') f(h',r,t')) \tag{9}$$

With our improved triple quality function, the probability of false detections could be reduced. For instance, in Fig. 2, if there is another negative triplet (Jerry, bornIn, Earth) with score of 2.5, then we can obtain the probability distribution over (Jerry, bornIn, Chicago), (Jerry, bornIn, Earth) and (Jerry, bornIn, Tom) with 0.695, 0.190 and 0.115 according to Eq. (8), respectively. Thus, the expectation of these three negative triplets is 1.653, and the quality value of (Jerry, bornIn, NewYork) is less than zero. Therefore, according to Eq. (9), (Jerry, bornIn, NewYork) is regarded as a noise. However, if we only use a single negative triple, e.g. (Jerry, bornIn, Tom) or (Jerry, bornIn, Chicago), to evaluate the original triplet, we might infer that (Jerry, bornIn, NewYork) might be both a convincing fact and a possible noise.

It should be pointed out that the original model [9] designed three kinds of triple confidence, considering both local and global structural information. In this study, we only considered the local triple confidence mode for simplicity. As a result, the modified local triple confidence is calculated as follows.

$$LT(h,r,t) = \begin{cases} \alpha LT(h,r,t), & PQ(h,r,t) \le 0 \\ LT(h,r,t) + \beta, & PQ(h,r,t) > 0 \end{cases} \tag{10}$$

In Eq. (10), α and β are hyperparameters introduced in CKRL, which can control the descent or ascent pace of local triple confidence. And the objective function is modified accordingly as shown in Eq. (11) (Step 12 in Algorithm 1).

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'_{(h,r,t)}} [\gamma + f(h,r,t) - f(h',r',t')]_+ LT(h,r,t) \tag{11}$$

Where the local triple confidence, i.e., $LT(h, r, t)$, instructs our model to pay more attention to more convincing facts. Negative triplets used in (11) were obtained by applying the confidence-aware negative sampling method. The set of negative triplets was given as follows.

$$S'_{(h,r,t)} = \{(h',r,t) \mid h' \in \mathbf{E}\} \cup \{(h,r,t') \mid t' \in \mathbf{E}\} \cup \{(h,r',t) \mid r' \in \mathbf{R}\} \quad (12)$$

where **E** and **R** denote the entity set and the relation set, respectively.

C. Differences with the Concept of Triple Confidence

The concept of negative triple confidence proposed in this paper is different with the triple confidence concept proposed in [9]. The concept of negative triple confidence is proposed for a different purpose. That is, the concept negative triple confidence in this paper is proposed for selecting high quality negatives, while the triple confidence concept proposed in [9] is to measure the correctness and importance of each triple. And the negative triples in [9] were still generated randomly. Moreover, the implement details are also different. Interested readers can refer to the original reference [9] for details.

IV. EXPERIMENTS

Link prediction is typically referred to as the task of predicting an entity that has a specific relation with another given entity, i.e., predicting *h* given (*r*, *t*) or *t* given (*h*, *r*). This is essentially a KG completion task that concentrates on the quality of knowledge representations. To verify the capability of our method in distinguishing noise and conflicts in KGs, we used link prediction as our evaluation task because noisy triples are harmful to embedding models. If we could obtain a representation of entities and relations with higher accuracy, it meant that the quality of the learned model was comparatively high, which further indicated that the model had noise detection ability.

A. Datasets

In this paper, we have evaluated our method on FB15K and WN18 [8], which are two typical benchmark datasets, extracted from Freebase and WordNet respectively. However, there are no explicitly labeled noises or conflicts in FB15K or WN18. Therefore, Xie et al. [9] generated three datasets (FB15K-N1, FB15K-N2, and FB15K-N3) based on FB15K with different noise rates (10%, 20% and 40%), to simulate the real-world KGs constructed automatically with less human effort. All three noisy datasets shared the same entities, relations, validation, and test sets. The statistics of FB15K and its noises are listed in TABLE I and TABLE II, in which #rel, #ent, #train, #valid, and #test represent relations, entities, training set, validation set and test set, respectively.

TABLE I. STATISTICS OF FB15K

Dataset	#rel	#ent	#train	#valid	#test
FB15K	1,345	14,951	483,142	50,000	59,071

TABLE II. DIFFERENT RATIOS OF NEGATIVES ON FB15K

Dataset	FB15K-N1	FB15K-N2	FB15K-N3
#neg triples	46,408	93,782	187,925

Following the practice of generating noises on FB15K, we directly utilized entity type information in WN18 and constructed three noisy datasets with negative triples, to be 10%, 20%, 40% of positive triples, which were denoted as WN18-N1, WN18-N2, WN18-N3, respectively. The statistics of WN18 and its noises are listed in TABLE III and TABLE IV, in which #rel, #ent, #train, #valid, and #test, represent relations, entities, training set, validation set, and test set, respectively.

TABLE III. STATISTICS OF WN18

Dataset	#rel	#ent	#train	#valid	#test
WN18	18	40,943	141,442	5,000	5,000

TABLE IV. DIFFERENT RATIOS OF NEGATIVES ON WN18

Dataset	WN18-N1	WN18-N2	WN18-N3
#neg triples	14,144	28,289	56,577

B. Evaluation Protocols

Following the examples of previous work, we used three measures as our evaluation metrics:

- For each testing triplet (*h*, *r*, *t*), the head *h* was replaced by every entity *e* in the KG, and its score was calculated according to the scoring function on the corrupted triplet (*e*, *r*, *t*). Then, ranking the scores of all corrupted triplets, we obtained the rank of (*h*, *r*, *t*), denoted as the Mean Rank(head). Similarly, we obtained another rank for (h, r, t), by corrupting the tail *t*, denoted as Mean Rank(tail). Then the average of Mean Rank(head) and Mean Rank(tail) was noted as Mean Rank [8]. Notice that the corrupted triplet should be eliminated before getting the rank of each testing triplet, if it already exists in the KG. This setting was called "Filter", with the original - without the removing operation - called "Raw". The lower the Mean Rank, the better.
- Hits@10 [8] indicated the proportion of correct answers ranked in the top 10, preferring a higher value. This metric has Raw and Filter settings as well.
- Mean reciprocal ranking (MRR), which has a range between (0, 1], can fully reflect the prediction ability of the model, and the higher the result, the better. We also followed the different evaluation settings of Raw and Filter, utilized in [22].

C. Experimental Settings

We used stochastic gradient descent (SGD) method to perform optimization in our approach. Before training, all entity and relation embeddings could be either initialized randomly, or pre-trained with TransE, and in this paper, we have referred to the practice of CKRL, which used the TransE model as the initialization input.

We adopted TransE and CKRL as baselines, as we utilized the source code provided by CKRL, which can be obtained from "*https://github.com/thunlp/CKRL*". We trained our model on FB15K and WN18, using mini-batch SGD with margin γ empirically set as 1.0. For fair comparisons, we directly set the learning rate to λ = 0.001, the number of candidate negative triples *NG* to 20, and the dimensions of both entity and relation embeddings to 50 in all models, unless specified. Moreover, we performed parameter analysis on the dataset FB15K-N1, where the margin γ was selected from {0.5, 1, 2, 3}, and the learning rate λ from among {0.001, 0.005, 0.01, 0.05}. In addition, the L1 norm was employed, and the "bern" [10] strategy was adopted, to decide whether to replace the head or the tail entity, with different probabilities, during the training.

D. Experimental Results

TABLE V. ENTITY PREDICTION ON FB15K

FB15K-N1	Mean Rank		MRR		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter
TransE	249	155	0.122	0.234	45.2	59.9
CKRL(LT)	**230**	**133**	0.136	0.278	45.9	62.5
Proposed	278	178	**0.138**	**0.334**	**47.9**	**68.3**

FB15K-N2	Mean Rank		MRR		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter
TransE	251	157	0.114	0.209	43.3	56.6
CKRL(LT)	**236**	**140**	0.128	0.255	44.7	60.4
Proposed Algorithm	287	188	**0.133**	**0.305**	**46.8**	**65.9**

FB15K-N3	Mean Rank		MRR		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter
TransE	268	175	0.099	0.176	40.4	51.9
CKRL(LT)	**246**	**150**	0.119	0.232	43.1	57.4
Proposed Algorithm	304	206	**0.127**	**0.281**	**45.6**	**63.1**

TABLE VI. ENTITY PREDICTION ON WN18

WN18-N1	Mean Rank		MRR		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter
TransE	**603**	**590**	0.106	0.139	**78.1**	87.2
CKRL(LT)	615	602	0.335	0.473	75.3	85.1
Proposed Algorithm	650	635	**0.348**	**0.494**	76.7	**88.1**

WN18-N2	Mean Rank		MRR		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter
TransE	**596**	**583**	0.061	0.079	74.4	82.8
CKRL(LT)	643	630	0.301	0.400	71.9	81.0
Proposed Algorithm	661	646	**0.308**	**0.414**	**74.5**	**84.6**

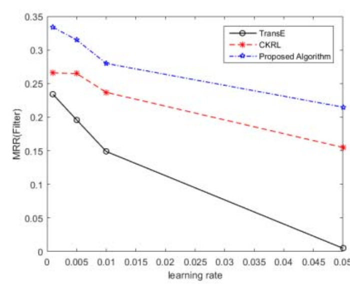
WN18-N3	Mean Rank		MRR		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter
TransE	**646**	**633**	0.042	0.051	66.8	73.6
CKRL(LT)	678	665	**0.203**	**0.259**	66.1	73.3
Proposed Algorithm	740	725	0.176	0.227	**68.7**	**77.9**

The results from our experiments, as well as baselines on link prediction with different noise rates, are shown in TABLE V and TABLE VI, from which we can observe that our model outperformed all baselines on FB15K, with evaluation metrics MRR, and Hits@10. Although the results under the Mean Rank metric were no better than TransE, and CKRL, the accuracy of top 10 ranking for the correct triplet was considerably high. The results for the CKRL model on WN18 were worse than those for the TransE model, even if the local triple confidence was taken into account by CKRL, while our approach still achieved good performance overall.

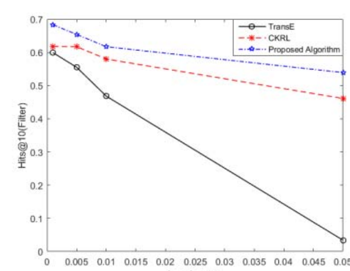
Note that we did not use the results directly from the original paper on FB15K, because some results were not listed under the MRR evaluation metric. We used the source code to rerun the experiment, and achieved an experimental result that was consistent with the original work.

In order to further understand the performance of our model with different parameters, we conducted a parametric analysis on learning rate λ, and margin γ, using FB15K-N1, and the outcomes are displayed in Fig. 3, Fig. 4 and Fig. 5 (we only considered the "Filter" setting). As these graphs show, our proposed model achieved better performance than both TransE and CKRL, with evaluation metrics MRR and Hits@10. The recommended parameters for our model were λ=0.001 and γ=1. Moreover, we paid attention to the number of negative triplets, which may affect our model. Due to limited computing resources, we only selected *NG* from among {10, 15, 20, 25}. As a result, our model showed preference for a higher *NG*, for more accurate outcomes.

Comparing evaluation results between different noisy datasets, we found that the improvement introduced by the proposed algorithm became more significant as the noise rate in KGs rose, which confirmed the capability of our approach in NKRL. With the confidence-aware negative sampling method, we could consistently provide high quality negative triplets for model training, that is to say, our method could settle the zero-loss problem to some extent, and could learn a better embedding model. Furthermore, a higher accuracy of MRR and Hits@10 meant that our model has a stronger ability to detect noises, by reducing false detections in KG.

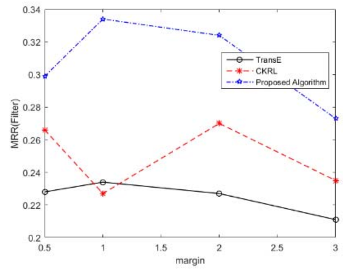


(a) The effect of the parameter learning rate on the metric MRR.

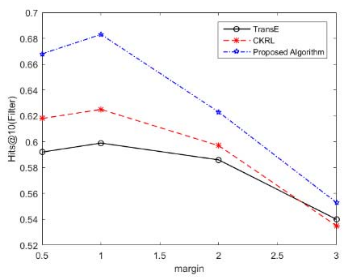


(b) The effect of the parameter learning rate on the metric Hits@10.

Fig. 3. Parameter analysis on the learning rate (λ), where γ and *NG* were fixed at 1 and 20, respectively.

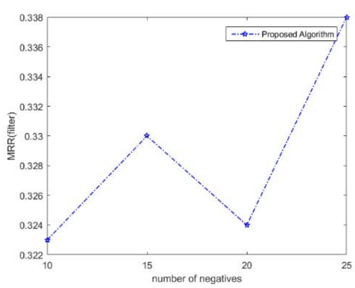


(a) The effect of parameter margin on the metric MRR.

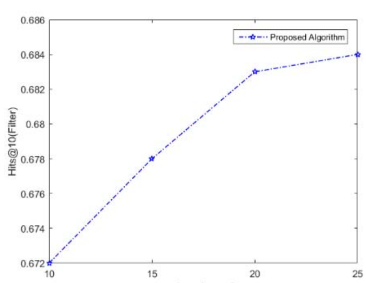


(b) The effect of the parameter margin on the metric Hits@10.

Fig. 4. Parameter analysis on margin (γ), where λ and *NG* remained at 0.001 and 20, respectively.



(a) The effect of parameter *NG* on the metric MRR.



(b) The effect of parameter *NG* on the metric Hits@10.

Fig. 5. Parameter analysis on number of negatives (*NG*). We kept λ and γ unchanged, at 0.001 and 1, respectively.

V. Conclusion and Future Work

In this paper, we introduced the concept of negative triple confidence to generate high quality negative triplets, which were used to support model training for detecting noises, while learning robust knowledge representations, simultaneously. To cope with the zero loss and false detection issues, we applied the concept of negative triple confidence to the CKRL model, and evaluated our model on link prediction tasks. The experimental results demonstrated that our model achieved good performance for most cases. In the future, in addition to structural information, we will explore ontological information to learn improved representation for noise detection.

References

[1] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of SIGMOD*, 2008, pp. 1247-1250.

[2] G. A. Miller, "WordNet: a lexical database for English," *Communications of The ACM,* vol. 38, no. 11, pp. 39-41, 1995.

[3] J. Lehmann *et al.*, "DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web,* vol. 6, no. 2, pp. 167-195, 2015.

[4] Z. Zheng, X. Si, F. Li, E. Y. Chang, and X. Zhu, "Entity Disambiguation with Freebase," in *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2012, pp. 82-89.

[5] A. Bordes, J. Weston, and N. Usunier, "Open Question Answering with Weakly Supervised Embedding Models," in *Proceedings of ECMLPKDD*, 2014, vol. 8724, pp. 165-180.

[6] A. Bordes, S. Chopra, and J. Weston, "Question Answering with Subgraph Embeddings," in *Proceedings of EMNLP*, 2014.

[7] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge Graph Embedding: A Survey of Approaches and Applications," *IEEE Transactions on Knowledge and Data Engineering,* 2017.

[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of NIPS*, 2013, pp. 2787-2795.

[9] R. Xie, Z. Liu, F. Lin, and L. Lin, "Does William Shakespeare REALLY Write Hamlet? Knowledge Representation Learning with Confidence," in *Proceedings of AAAI*, 2018.

[10] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," in *Proceedings of AAAI*, 2014, pp. 1112-1119.

[11] P. Wang, S. Li, and R. Pan, "Incorporating GAN for Negative Sampling in Knowledge Representation Learning," in *Proceedings of AAAI*, 2018.

[12] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of CVPR*, 2015, pp. 815-823.

[13] S. Heindorf, M. Potthast, B. Stein, and G. Engels, "Towards Vandalism Detection in Knowledge Bases: Corpus Construction and Analysis," in *Proceedings of SIGIR*, 2015, pp. 831-834.

[14] T. P. Tanon, L. Pintscher, L. Pintscher, and L. Pintscher, "From Freebase to Wikidata: The Great Migration," in *Proceedings of WWW*, 2016, pp. 1419-1428.

[15] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A Review of Relational Machine Learning for Knowledge Graphs," *Proceedings of the IEEE,* vol. 104, no. 1, pp. 11-33, 2015.

[16] S. Heindorf, M. Potthast, B. Stein, and G. Engels, "Vandalism Detection in Wikidata," in *Proceedings of CIKM*, 2016, pp. 327-336.

[17] M. Fan, Q. Zhou, E. Chang, and T. F. Zheng, "Transition-based Knowledge Graph Embedding with Relational Mapping Properties," in *Proceedings of PACLIC*, 2014, pp. 328-337.

[18] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge Graph Embedding via Dynamic Mapping Matrix," in *Proceedings of ACL*, 2015, pp. 687-696.

[19] H. Xiao, M. Huang, and X. Zhu, "From one point to a manifold: Knowledge graph embedding for precise link prediction," in *Proceedings of IJCAI*, 2016.

[20] Z. Tan, X. Zhao, Y. Fang, and W. Xiao, "GTrans: Generic Knowledge Graph Embedding via Multi-State Entities and Dynamic Relation Spaces," *IEEE Access,* vol. 6, no. 99, pp. 8232-8244, 2018.

[21] Y. Tay, A. T. Luu, and S. C. Hui, "Non-Parametric Estimation of Multiple Embeddings for Link Prediction on Dynamic Knowledge Graphs," in *Proceedings of AAAI*, 2017, pp. 1243-1249.

[22] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of ICML*, 2016, pp. 2071-2080.