World Wide Web
https://doi.org/10.1007/s11280-019-00711-y

Adaptive knowledge subgraph ensemble for robust and trustworthy knowledge graph completion

Guojia Wan[1] · Bo Du[1] · Shirui Pan[2] · Jia Wu[3]

Received: 23 May 2019 / Revised: 11 July 2019 / Accepted: 22 July 2019 /
Published online: 15 October 2019

Abstract

Knowledge graph (KG) embedding approaches are widely used to infer underlying missing facts based on intrinsic structure information. However, the presence of noisy facts in automatically extracted or crowdsourcing KGs significantly reduces the reliability of various embedding learners. In this paper, we thoroughly study the underlying reasons for the performance drop in dealing with noisy knowledge graphs, and we propose an ensemble framework, *Adaptive Knowledge Subgraph Ensemble* (*AKSE*), to enhance the robustness and trust of knowledge graph completion. By employing an effective knowledge subgraph extraction approach to re-sample the sub-components from the original knowledge graph, *AKSE* generates different representations for learning diversified base learners (e.g., *TransE* and *DistMult*), which substantially alleviates the noise effect of KG embedding. All embedding learners are integrated into a unified framework to reduce generalization errors via our *simple* or *adaptive* weighting schemes, where the weight is allocated based on each individual learner's prediction capacity. Experimental results show that the robustness of our ensemble framework outperforms exiting knowledge graph embedding approaches on manually injected noise as well as inherent noisy extracted KGs.

Keywords Trustworthy knowledge graph · Knowledge graph completion · Link prediction · Knowledge graph embedding · Never-ending language learning

1 Introduction

Knowledge graphs, also known as graph-structured knowledge bases (KBs), such as Freebase [1], WordNet [19], and Never-ending language learning (NELL) [20], have attracted wide attention due to their benefits in numerous downstream applications, including question answer [10], information retrieval, recommendation systems [40], etc. Knowledge

This article belongs to the Topical Collection: *Special Issue on Trust, Privacy, and Security in Crowdsourcing Computing*
Guest Editors: An Liu, Guanfeng Liu, Mehmet A. Orgun, and Qing Li

✉ Bo Du
remoteking@whu.edu.cn

Extended author information available on the last page of the article.

graphs are directed graphs consisting of facts. A fact is stored in the form of triplet (h, r, t), where r refers to relation, and h and t denote *head entity* and *tail entity*. For example, (`washington, capital_of, america`) indicates that Washington is the capital of America. Despite of large scale, knowledge graphs still suffer from incompleteness and unreliable facts. Knowledge graph completion (KGC) aims to mine missing facts based on observed knowledge.

Recently there has been increasing methods on knowledge graph embedding approaches [4, 27, 31] for KGC. These methods learn topological connection information and associated entities and relation into a low dimensional continuous vector or matrix. The score function is defined to measure the plausibility of triplets [21]. Then KGC, also known as link prediction, can be carried out simply by means of a ranking procedure.

However, most embedding approaches only perform well on reliable human-curated knowledge graphs. Considering bottom-up automatically constructed knowledge graphs or crowdsourcing knowledge graph, Never Ending Language Learning (NELL) [20], ConceptNet [28], etc., such knowledge construction systems often suffer from the pollution of incorrect facts [34]. Embedding models fall short on such noisy KGs [26]. The noise in the extracted KGs is divided into source noise and extraction errors. The former come from low-quality Web text [38], and the latter is from the limitations of NLP approaches [25, 39]. For instance, the sentence `New York was the capital of America in 1789.` is prone to be misunderstood as knowledge (`new_york,capital_of,america`) by extraction systems (Figure 1). Consequently, these errors render triplets corrupted and cause incorrect links [14–16]. Then the wrong structure information is learned with perturbed supervised label information. As a result, the robustness of the learned models is inevitably degenerated, resulting in dubious link prediction, especially using neural network models [42].

Ensemble approaches, such as *bagging* [3], and *boosting* [5], etc., have successfully achieved impressive performance in various applications. Compared with a single learner, the ensemble methods are able to improve the accuracy, robustness, and stability. However, due to the large size of KGs, the traditional ensemble strategy fail to adapt to the existing knowledge graph embedding approaches. More than that, knowledge graphs are in the form of heterogeneous graph, of which entities and relations have correlations with each other so that disobey the independent and identically distributed assumption.

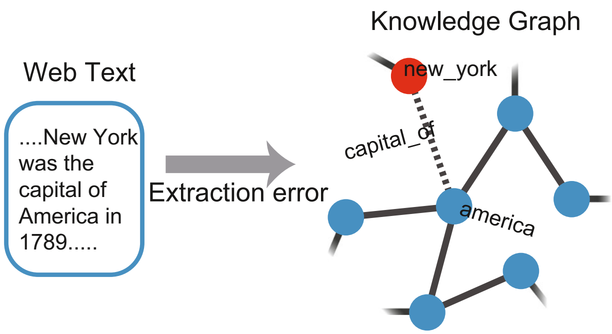

Figure 1 Effect of incorrect facts extracted from Web text on a real-world knowledge graph. The extraction system extracted a fact (`new_york,capital_of,america`). Thus, the wrong link between `new_york` and `america` was made

Inspired by the above observations, in this paper, we propose a simple but efficient ensemble strategy that adaptively captures the diversity of the subgraphs of the knowledge graph and builds a robust embedding learner on noisy KG, hence improving the KGC task [13]. Our basic idea is presented in Figure 2. First, we under-sample a KG G to generate a set of knowledge subgraphs $\{G_1, G_2, G_3, \cdots, G_k\}$ so that partial entities (including true and false facts) are abandoned. It is noted that these knowledge subgraphs not only generate the high-diversity embedding learners but also lower the effect of noisy data on each embedding learner. Then for each knowledge subgraph G_k, we train a learner f_k using the base knowledge graph embedding learner. In the next stage, each learner makes a prediction on an out-of-bag subset, and the empirical variance is obtained to measure prediction capability. We apply softmax probability to assign the weights of every learners adaptively. Finally, multiple knowledge subgraph learners are integrated to generate a unified representation. Based on such joint representation, we conducted link prediction on human-curated KGs and an automatically extracted KG, NELL. The results of this study show that our strategy has more robust performance both on manually injected noise and inborn errors.

Our main contributions are summarized as follows:

- We thoroughly study the problem of noise knowledge graph completion, point out the underlying reasons of the low performance of exiting algorithms, and present a novel ensemble algorithm *ASKE* as a solution.
- We propose a novel ensemble-based algorithm for knowledge graph embedding. Our method essentially reduces the generalization error by an effective knowledge subgraph extraction mechanism and an adaptive weighting scheme.
- We demonstrate that our algorithm significantly enhances the robustness of embedding in dealing with noise KGs.

The rest of this article is organized as follows. Section 2 reviews related works. Section 2 provides basic notations and definitions. In Section 3, we discuss ensemble strategy on

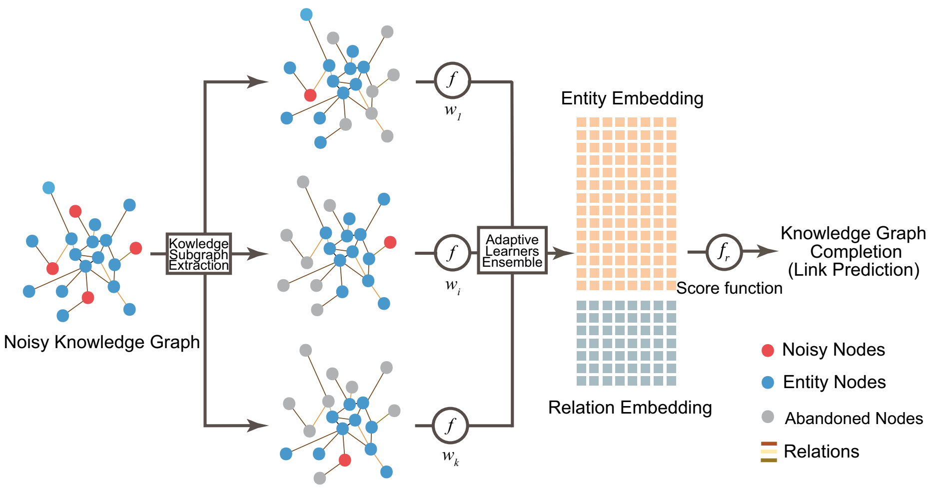

Figure 2 The ASKE framework. From left to right: 1) Knowledge subgraph extraction process, 2) multiple learners training, 3) adaptive weight ensemble to obtain robust entities and relations representation, 4) knowledge graph completion (link prediction)

knowledge graph embedding approaches, then introduce our framework. Section 5 describes our experiment datasets,evaluation protocols and parameter settings. Section 6 discusses our experimental results. In Section 7, we present our conclusions.

2 Related works

2.1 Knowledge graph embedding

KG embedding aims to embed entities and relations into low-dimensional continuous space, so as to simplify computations on the KG. Based on the semantic similarity in the embedding space, KGC is conducted to infer missing facts. Originating from *word2vec* [18], *TransE* [2] assumes that the head entity vector should be closed to the tail entity vector once translated by the relation vector, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Many other efforts have attempted to deal with the poor performance of *TransE* in the cases of 1-N, N-1, N-N relations. *TransR* [11] is proposed to enrich the expression of *TransE* by introducing relation-specific space. The entities are projected into relation vector space by relation-specific matrix. *TransH* [32] shares a similar idea with *TransR*. *Rescal* [23] models triplets by means of bilinear operations over entity vector and relation matrix based on tensor factorization. *DistMult* [37] simplifies *Rescal* by restricting the relation matrix to diagonal matrix. *ComplEx* [29] introduces complex-valued representation to extend *DistMult* to better deal with asymmetric relations. Recently, Demetters [4] first proposed a CNN-based neural networks *ConvE* to learn latent feature on KG. Entities and relation vectors are reshaped and concatenated for CNN filter kernels. Then the feature map is classified by the full connected layer, and the link prediction result is assigned by the classification labels. Another convolution idea neural networks is *RGCN* [27], where the convolution operator capture nodes structure feature map, i.e. entity or relation embedding, by first-order approximation in locality information in KG and self loop mechanism. Although these methods have made significant progress on modeling KG and downstream tasks, the noise attack issues have been less studied.

2.2 Noisy knowledge graph completion

Automatic extracted and crowdsourcing KGs such as NELL [20], and ConcepNet [28], suffer from possible noise and conflicts due to limited human supervision [12, 35]. NELL is a semi-supervised, ontology-driven, iterative system that extracts facts from the Web text of more than one billion documents. In each iteration, NELL learns new facts and assigns a confidence score using seed facts and available evidence. As NELL is based solely on confidence scores, it is difficult to avoid corruption due to incorrect facts. Pujura *et al.* [26] investigated the robustness of classic embedding approaches on several realistic KGs. They found that existing embedding methods are sensitive to the sparsity and the noise in KGs.

A promising direction for this challenge is error detection [25, 41], which tries to select clean facts out of noisy facts. However, these methods are constrained by feature engineering due to the incompleteness of external information. Another direction is to build noise robust embedding. Xie *et al.* proposed *CKRL* [36] by defining local triplets confidence and global path confidence as an extension of *TransE*. The model is tested on FB15K only with additional noise, but in the real-world knowledge graph, the noise attack perturbs the structure information of KG.

2.3 Ensemble learning

Ensemble learning is well known in machine learning and its applications are broad. There are various classifier ensemble approaches, including *bagging* [3], *boosting* [5], etc. Noisy training data tends to increase the variance in the results produced by a given learner. By learning a combination of learners, the variance can be reduced [6]. Variance-reducing methods such as the bagging decision tree have been shown to be robust in the presence of high-level noise [17]. In addition, *bagging* has been shown to perform well on imbalanced classification problems [7]. However, these algorithms have typically been applied to traditional classification tasks, but we focus on embedding learning in this paper.

3 Formulations

In this section we formally define some problems and notations. Let $\mathcal{E} = \{e_1, \cdots, e_n\}$ be the set of all entities and $\mathcal{R} = \{r_1, \cdots, r_m\}$ be the set of all relations. A knowledge graph is a directed graph as $G = (\mathcal{E}, \mathcal{R}, E)$, where E denotes a set of edges as well as facts, $E \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$. A fact stored in the form of triplet (e_1, r, e_2). We also use (h, r, t) to clarify the head entity and the tail entity. In a real world knowledge graph, facts are collected into a positive triplet set $\mathcal{D}^+ \subseteq E$. We model each possible triplet (h_i, r_j, t_k) over $\mathcal{E}$ and $\mathcal{R}$ as a binary random variable $y_{ijk} \in \{0, 1\}$ that indicates its existence.

$$y_{ijk} = \begin{cases} 1, & (h, r, t) \in \mathcal{D}^+ \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Definition 1 (Noisy knowledge graph) Given a noisy knowledge graph $G = (\mathcal{E}, \mathcal{R}, \mathcal{D}^+)$ collected from experts or automatic extract systems, $\exists \mathcal{N} = \{(h_i, r_j, t_k)\} \subset \mathcal{D}^+$, where $\forall (h_i, r_j, t_k) \in \mathcal{N},\ \hat{y}_{ijk} \neq y_{ijk_g}$. $\hat{y}_{ijk}$ is the is the observed value. y_{ijk_g} is the true value.

The presence of noise mislabels the edges and cause incorrect link. Therefore the model learns wrong structure information, leading to reduce the trust of prediction.

Definition 2 (Knowledge graph embedding) Given the input of a knowledge graph $G = (\mathcal{E}, \mathcal{R}, \mathcal{D}^+)$, a knowledge graph embedding is a mapping $f : (e_i, r_i) \to (\mathbb{R}^{d_1},\ \mathbb{R}^{d_2})$, $\forall e_i \in \mathcal{E}$, $\forall r_i \in \mathcal{R}$, such $d_1 \ll |\mathcal{E}|$, $d_2 \ll |\mathcal{R}|$ and the mapping f preserves some proximity measure defined on the knowledge graph G.

The goal of knowledge graph embedding can be stated as producing a model with smallest possible loss $\mathcal{L}$. The prediction for a model f and test example x is $y = f(x)$. Then we can define bias and variance for a knowledge graph embedding model.

Definition 3 (Bias of knowledge graph embedding) The bias of a knowledge graph embedding model f is $B_f = (\bar{y} - y)^2$

Definition 4 (Variance of knowledge graph embedding) The variance of a knowledge graph embedding model f is $\mathrm{Var}_f = \mathbb{E}[(y - \bar{y})^2]$

Definition 5 (Noise of knowledge graph) The noise of a knowledge graph is $\epsilon = y - y_g$. y_g is the true label.

In words, the bias is the loss incurred by prediction relative to the optimal prediction. The variance is the average loss incurred by predictions relative to the main prediction. Noise is the unavoidable component of the loss, incurred independently of the models.

Knowledge graph completion expects a model f to predict potential correct facts. An embedding based approach defines the score function $f_r((h_i, r_j, t_k))$ to measure the probability if the fact (h_i, r_j, t_k) exists. Bold letters **h**, **r**, **t** denote embedding vector.

4 Methodology

4.1 Knowledge graph embedding model ensemble

Suppose a given KG consists of m relations and n entities. In a typical KG embedding method, score function $f_r(h, r, t)$ is defined to describe the plausibility for an observed fact (h_i, r_j, t_k). Here, we introduce the probability model to interpret the score function,

$$P(y_{ijk} = 1) = \sigma(f_r(h_i, r_j, t_k)), \tag{2}$$

where σ is the sigmoid function.

The model predicts the existence of a triplet (h, r, t) which represents the confidence that a triplet exists. Notably, the presence of specific triplets is correlated with other triplets. It is almost impossible to achieve joint probability distribution on a whole KG. Thus, knowledge graph embedding approaches assume the y of all triplets are conditionally independent. Based on the assumption, the training KG embeddings can be converted into binary classification. The supervised constraint refers to whether or not a triplet belongs to $\mathcal{D}^+$. A general loss function of a KG embedding model can be written as follows:

$$\mathcal{L} = L(f((h, r, t)), y_g), \tag{3}$$

where y_g is true value and L is a certain loss.

Based on the above proposition, we apply bias-variance decomposition to study a KG embedding model [30]. For the sake of simplicity, we study the mean squared error (MSE) loss rather than the pairwise ranking loss that common sight in the previous works. The generalization error $E_G^0 = \mathbb{E}_G[\mathcal{L}]$ of the model f on a knowledge graph G can be written as follows:

$$\begin{aligned} E_G^0 &= \mathbb{E}_G[(f(x) - y_g)^2] \\ &= \mathbb{E}_G[(f(x) - \bar{f}(x))^2] + (\bar{f}(x) - y)^2 \\ &\quad + \mathbb{E}_G[(y - y_g)^2] \\ &= Var_f + B_f^2 + \epsilon^2, \end{aligned} \tag{4}$$

where ϵ^2 denotes the label noise error with zero mean assumption that $\mathbb{E}[\epsilon] = 0$, Var_f denotes the variance, and B_f denotes the bias. In (4), it is clear that the generalization error of a single knowledge graph embedding model depends on its variance, bias, and noise. The noise in the KG caused by incorrect facts are in the form of incorrect labels.

Let $f_1, f_2, \cdots, f_k$ denote K knowledge graph embedding models separately trained on G. The output of the ensemble knowledge graph embedding estimators for input x is defined as the simple average method,

$$f_{ens}(x) = \frac{1}{K}\sum_{k=1}^{K} f_k(x). \tag{5}$$

Then, its generalization error is

$$E_G^{ens} = \mathbb{E}_G[(\frac{1}{K}\sum_{k=1}^{K} f_k(x) - y_g)^2]. \tag{6}$$

If K models are the same and mutually uncorrelated, then we have

$$\begin{aligned} E_G^{ens} &= \frac{1}{K}\mathbb{E}_G[(f(x) - \bar{f}(x))^2] \\ &\quad + (\bar{f}(x) - y)^2 + \mathbb{E}_G[(y - y_g)^2] \\ &= \frac{1}{K}Var_f + B_f^2 + \epsilon^2. \end{aligned} \tag{7}$$

In (7), $E_G^{ens} \leqslant E_G^0$ always holds. $Var_{f_{ens}}$ reduces to $\frac{1}{K}Var_f$, and this reduction decreases the generalization error due to multiple models averaging. The presence of incorrect facts increases the Var and ϵ of knowledge graph embedding model. Therefore, an ensemble strategy has the potential to improve knowledge graph embedding approaches, even if there are many unreliable facts.

4.2 Latent feature level adaptive ensemble strategy

In Section 4.1, we discuss the potential of knowledge graph embedding ensemble strategy in terms of reducing variance in order to dealing with noisy data. However, directly utilizing (5) costs expensive computation due to the large size of candidate set $\mathcal{E}$ that has to be traversed during the prediction procedure. The time complexity is nearly $O(k|\mathcal{E}|f_r)$, where k is the number of learners. For dealing with the problem, we proposed a combination strategy as follows:

$$\begin{aligned} \mathbf{h}_i = \sum_{k=1}^{K} w_{k,r}\mathbf{h}_{k,i}, \quad \mathbf{r}_i &= \sum_{k=1}^{K} w_{k,r}\mathbf{r}_{k,i}, \quad \mathbf{t}_i = \sum_{k=1}^{K} w_{k,r}\mathbf{t}_{k,i}, \\ f_{ens}(x) &= f(\mathbf{h_i}, \mathbf{r_i}, \mathbf{t_i}), \\ s.t. \sum_{k=1}^{K} w_{k,r} &= 1, r \in \mathcal{R}, \end{aligned} \tag{8}$$

where $\mathbf{h}_{k,i}, \mathbf{t}_{k,i}, \mathbf{r}_{k,i}$ are i-th latent representations , i.e. embeddings, trained from k-th knowledge graph embedding models f_k. $w_{k,r}$ is weight coefficient for r-th relation. The weights are normalized for each k. For toy example, 3 base models are trained. For a query $(h, r, ?)$, assuming $w_{k,r} = 1/3$ for r-th relation, then $\mathbf{h} = 1/3 \cdot \mathbf{h}_1 + 1/3 \cdot \mathbf{h}_2 + 1/3 \cdot \mathbf{h}_3$, $\mathbf{r} = 1/3 \cdot \mathbf{r}_1 + 1/3 \cdot \mathbf{r}_2 + 1/3 \cdot \mathbf{r}_3$.

Our strategy can be regarded as a latent feature level fusion method instead of a direct ensemble. Thus, the time complexity of predict procedure is equal to a single knowledge graph embedding model, i.e. $O(|\mathcal{E}|f_r)$. On the other hand, considering the imbalanced relation triplets and the variability of prediction capability on different relations, we reduce the weights granularity from individual model to specific relation.

For assigning the weights of different knowledge graph embedding learners, we propose two combination strategies: *Simple average* and *Adaptive weighting*.

Simple Average *Simple average* is the original methods of *bagging* [3]. The weights are equal to each other from K models, which simply average the embedding vectors in the parameter space:

$$w_{k,r} = \frac{1}{K}, \tag{9}$$

where $r \in \mathcal{R}$, $k = 1, \cdots, K$. This method is named as *KSE*.

Adaptive Weighting An alternative combination strategy is to assign different weights to each of models, the rationale being that the models should be weighted according to the prediction capability. We present a measurement of capability as follows:

$$\alpha_{k,r} = \frac{1}{Var(f_k)_{G'_{k,r}}}, \tag{10}$$

where $G'_{k,r}$ is the out-of-bag of r relation, and $G'_{k,r} \subset G'_k$, $G'_k = \mathcal{D}^+ - G_k$. $Var(f_k)_{G'_{k,r}} = (f_k(x; G_k) - \overline{f_{ens}}(x))^2$ means the empirical variance to k-th model on the r-th relation. $\alpha_{k,r}$ is the uncertainty of the specific learners in order to measure prediction capability. Then, we assign weights according to $\alpha_{k,r}$ with softmax probability:

$$w_{k,r} = \frac{e^{\alpha_{k,r}}}{\sum_{i=1}^{K} e^{\alpha_{i,r}}}. \tag{11}$$

In (11), the k learners are weighted by the inverse of the corresponding out-of-bag prediction variance, i.e., with larger variance, the weights are assigned with smaller weights. We name this method *AKSE*.

4.3 Knowledge subgraph extraction

Diversity among all learners has a significant effect on ensemble framework [3]. In order to obtain diverse estimators, we construct a series of diverse subgraphs as input via re-sampling approaches.

However, traditional sampling approaches from machine learning tasks are not suitable for the graph-based data due to the ignorance of correlation among training triplets. Generally, facts in a KG indeed are linked to each other. Therefore we give two definitions to describe the correlation:

Definition 6 (Fact neighbor) Given a positive facts set $\mathcal{D}^+$. The fact neighbor of a fact $T_0 = (h_0, r_0, t_0)$ is a triplet set $\mathcal{B}$, where $\mathcal{B} = \{(h_0, r, t) | (h_0, r, t) \in \mathcal{D}^+, (h_0, r, t) \neq T_0\} \cup \{(h, r, t_0) | (h, r, t_0) \in \mathcal{D}^+, (h, r, t_0) \neq T_0\}$.

Definition 7 (Facts network) A facts network is an undirected graph $G_{\mathcal{D}^+} = (\mathcal{D}^+, E)$, where $\mathcal{D}^+$ denotes the positive triplet set, and E is the set of undirected edges. An edge (T_v, T_u) links two facts from T_v to T_u if $T_v \in \mathcal{B}_{T_u}$.

Based on the concept facts network, we propose an alternative re-sampling approach named *knowledge subgraph extraction* to build diverse input training subgraphs. We keep the nodes at facts level. Therefore graph-based sampling methods can be applied to facts network. Random walk is computationally efficient in terms of both space and time requirements. We describe each jump of the walk as following:

$$P(T_t|T_{t-1}, \bar{S}) = \begin{cases} \frac{\pi_{(t,t-1)}}{Z}, & T_t \in \mathcal{B}_{T_{t-1}} \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

$$P(S = \text{‘}Restart\text{’}) = p, p \in [0, 1], \tag{13}$$

where T_{t-1} is the current node, T_t is the next node. $P(T_t|T_{t-1})$ models the probability that the walk reach to the next node. $\pi_{(t,t-1)}$ is the unnormalized transition probability between facts T_t and T_{t-1}. In the facts network $G_{\mathcal{D}^+}$, we suppose that the weights of each edges are equal so that $\pi_{(t,t-1)} = 1$. Then the transition probability is normalized by degree Z of T_{t-1}.Here we introduce restart mechanism $P(S)$ to control random walk. p is restart probability. Before each jump, the walk randomly chooses ‘Continue’ or ‘Restart’. If $S =$ ‘$Restart$’, the walk restarts with a random triplet, otherwise continue to search the next node in neighbor facts set, as shown in Figure 3. The restart mechanism not only controls the length of random walk but balances the depth and breadth exploration.

Our subgraph extract method can extract local connection information among facts networks. In addition, using random walk as the basis for our algorithm gives us two other benefits: 1) the method is flexible in exploring KG neighborhoods by means of restart probability p and subgraph scale l, 2) the under-sampling mechanism results in that partial noisy facts are stayed in G_k. This means each individual knowledge graph embedding learner are trained under less noise attack than the whole original KG so that increase noise resistance.

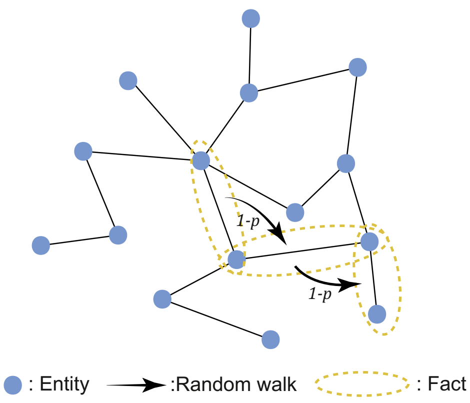

Figure 3 Random walk on facts network

4.4 Base knowledge graph embedding learners

Existing knowledge graph embedding approaches could be divided into shallow models and deep models. Due to the expensive cost of computation and the complexity using deep models, we choose shallow models as the base learners. Specifically, *TransE* and *DistMult* are the two most representative shallow models, and *TransR* is an extension of *TransE*. *TransE* embeds entities and relations into geometric vector space based on semantic translation equivalence. A triplet (h, r, t) is scored as

$$f_r(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_q, \tag{14}$$

where q is L_1 or L_2 norm.

TransR makes an assumption that entity vector and relation vector should not be in the same space due to semantic difference. Therefore relation-specific space $\mathbf{M_r}$ is introduced to project entities into relation space, then translation vector is modeled in that space. Given a triplet (h, r, t), the score function is defined as

$$\begin{aligned} &\mathbf{h}_\perp = \mathbf{M_r}\mathbf{h}, \mathbf{t}_\perp = \mathbf{M_r}\mathbf{t}, \\ &f_r(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_q, \end{aligned} \tag{15}$$

where $\mathbf{h},\ \mathbf{t} \in \mathbb{R}^d,\ \mathbf{r} \in \mathbb{R}^n, \mathbf{M_r} \in \mathbb{R}^{n\times d}$ is relation-specific space.

Algorithm 1 ASKE algorithm.

Input: Knowledge graph facts set $\mathcal{D}^+$, The base learner f;
Parameter: Restart probability p; Subgraph scale l; Number of learners K; The hyper-parameters of base learner θ_f;
Output:Entity embedding $\mathbf{E}$, Relation embedding $\mathbf{R}$;

for $i = 1\ to\ K$ **do**
  G_k=**KnolwegdeSubgraphExtraction**$(\mathcal{D}^+, p, l)$
  $E_i, R_i = f(G_k, \theta_f)$
  for each relation $r \in \mathcal{R}$ **do**
    $\alpha_{i,r} = OutofBagEstimation(G'_{k,r})$
Merge E_i, R_i according to (11)
return E, **R**

KnowledgeSubgraphExtraction$(\mathcal{D}^+, p, l)$

$G_k = \emptyset$; $T_0 = Sample(\mathcal{D}^+)$
repeat
  if p **then**
    $T_t = RandomWalk(T_0)$;
    Add T_t into G_k
    $T_0 = T_t$
  else
    $T_0 = Sample(\mathcal{D}^+)$
until $|G_k| = l|G|$
return G_k

DistMult is based on tensor factorization, which associate relation r to a diagonal matrix $\mathbf{R_r} \in \mathbb{R}^{d\times d}$. The score function is

$$f_r(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \mathbf{h^T R_r t}. \tag{16}$$

Notably, the recent improvement on *DistMult* presents competitive performance on standard benchmarks [9], which indicates an improvement gap for shallow models.

4.5 The AKSE algorithm

The pseudo-code for *AKSE* is given in Algorithm 1. *AKSE* consists of two main components: first, a subgraph generator, and second, a parallel training framework. In a subgraph generator, the random walk starts with an initial fact T_0 that uniformly sampled from G. Then the walk uniformly samples from the neighbors of the fact. If p, the walk restarts, otherwise continue to explore the next node. The random walk is repeated running until $\|G_k\| = l\|G\|, l \in [0, 1]$. Then we obtain a series of subgraphs $\{G_1, G_2, G_3, \cdots, G_k\}$. In the model training stage, each of the k-th learner is independently trained on G_k. In the next, the weights for multiple learners are determined by an out-of-bag estimation on $G'_{k,r}$. Finally, all latent representations of entities and relations are merged into a unified ensemble embedding.

5 Experimental settings

Datasets We conducted experiments on human-curated KGs with injected manual noise: WN18 and FB15K. Due to the test set leakage problems [4], we also conducted experiments on FB15K-237 and WN18RR. Furthermore, the models are evaluated on the inborn noisy KG, NELL. Because of its large scale, we extracted a subset from the 1115th dump of NELL.[1] In recent evaluations [20], NELL facts had a precision level ranging from 0.75-0.85 for confident extractions. Thus NELL79K preserved partial unreliable facts from NELL. Table 1 summarizes the datasets used in this paper.

Training To better evaluate our ensemble strategy, we selected and compared eight classical knowledge graph embedding methods: *TransE* [2], *TransR* [11], *HoLE* [22], *Distmult* [9, 37], *ComplEx* [29], *CKRL(LT)* [36], *RGCN* [27], and *ConvE* [4].[2] They all based on public implementation [8].[3] These approaches mainly based on energy function framework:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{D}^+} \sum_{(h',r',t')\in S'_{(h,r,t)}} max(0, \gamma + f_r(h, r, t) - f_r(h', r', t')), \tag{17}$$

where γ is a margin hyper-parameter. $S'_{(h,r,t)}$ is the negative sampling triplet set of (h, r, t). In the baselines training stage, we followed the hyper-parameter settings of previous works. For our *AKSE* and *KSE*, the number of base learner k was in $\{10, 25, 50, 75, 100\}$. Restart probability p was in $\{0.05, 0.1, 0.2, 0.3, 0.5, 0.6\}$. Subgraph scale l was fixed at 0.7. The hyper-parameters about the base learners of *AKSE* and *KSE* are re-gridsearched.

[1] https://rtw.ml.cmu.edu/rtw/
[2] https://github.com/TimDettmers/ConvE
[3] https://github.com/thunlp/OpenKE

Table 1 Statistics of the datasets

Datasets	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	Train	Test	Valid
FB15K	14951	1345	483142	59071	50000
WN18	40943	18	141442	5000	5000
FB15K-237	14541	237	272115	20466	17535
WN18RR	40943	11	86835	3134	3034
NELL79K	79222	810	610147	10000	10000

Evaluation Protocol *Link prediction* is a widely used protocol for evaluation the performance of knowledge graph completion tasks. It aims to predict an entity given an incomplete fact $(h, r, ?)$ or $(?, r, t)$. Suppose the i-th test triplet $(h, r, ?)$ for predicting the possible tail entities, we generate a set of all possible facts $\mathcal{P} = \{(h, r, t_i)|\forall t_i \in \mathcal{E}\}$. In this task, candidate set are ranked according to their predicted scores. The rank of the i-th test triplets is as follows:

$$Rank_i = \sum_{x_j \in \mathcal{P}} I[f_r(x_j) \leq f_r(\widetilde{x_i})], \tag{18}$$

where $I[Q]$ is 1 if the condition Q is true, and 0 otherwise. The rank of head/tail entity is also referred to *left/right rank*. Following previous works [2, 11], we report two measures as our evaluation metrics: the average rank of all the correct entities (Mean Rank) and Hit@10.

$$MR = \frac{1}{2|\mathcal{T}|} \sum_{x_i \in \mathcal{T}} Rank_i^{left} + Rank_i^{right}, \tag{19}$$

$$Hit@k\ (\%) = \frac{100}{2|\mathcal{T}|} \sum_{x_i \in \mathcal{T}} I[rank_i^{left} \leq k] + I[rank_i^{right} \leq k], \tag{20}$$

where $\mathcal{T}$ is test set.

We also conduct *triplet classification* to evaluate the performance. *triplet classification* aims to make a judgment on a triplet (h, r, t), which is a binary classification task. Based on the plausibility of triplets, a threshold τ is learned as the classification boundary. Following previous works, we learn the τ by maximizing the classification accuracy on the validation set.

Considering that a triplet (h', r, t) or (h, r, t') may also exist in the KG, the metrics may be underestimated. Therefore, we filtered the triplets that appeared in the train set during the test process. This is referred to as the *filtered* setting in Bordes et al. [2].

6 Results and discussions

6.1 Evaluation on KG with injected noise

In this section, we evaluate the robustness of knowledge graph embedding methods on benchmark datasets with injected noise. $x\%$ triplets were replaced by corrupted triplets, where the head entity or tail entity was randomly selected from entity candidate set $\mathcal{E}$. Then, the embedding approaches were trained with a noisy version of the benchmark.

The results are presented in Figure 4 and Table 2. As Table 2 shows, when KGs are free of noise, the ensemble strategy still presents competitive enhancements to shallow models, especially in inverse relation filtered KGs, FB15K-237, and WN18RR. Once incorrect facts

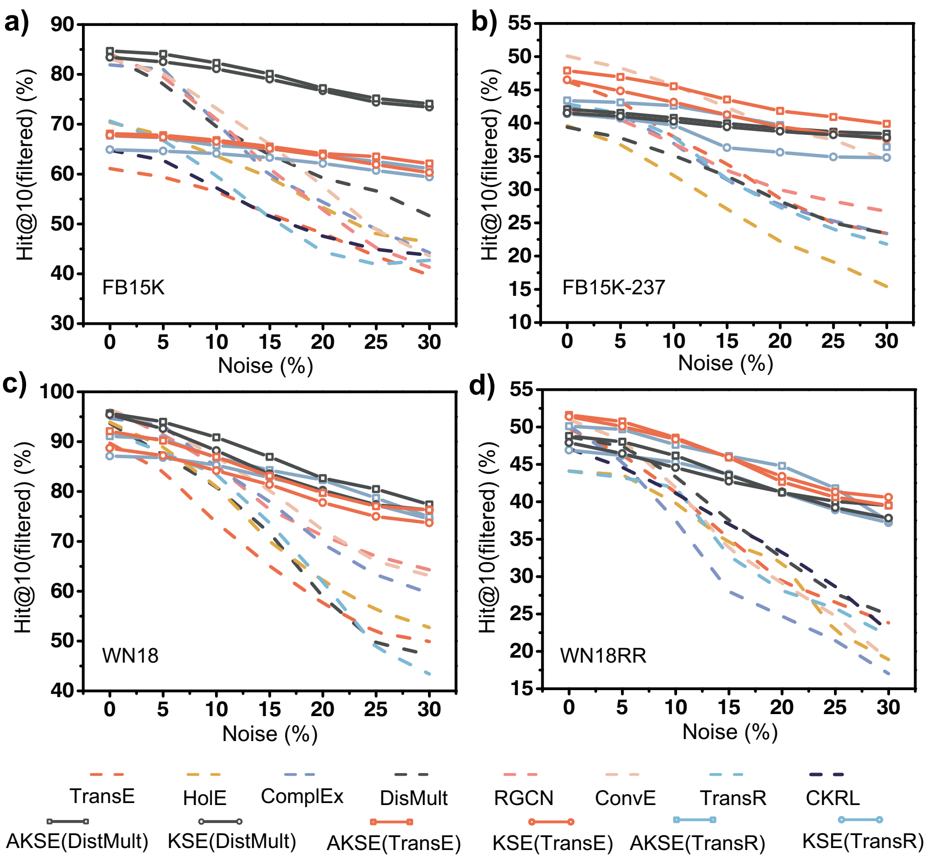

Figure 4 Hit@10 metric vs. injected noise ratio. **a** FB15K; **b** FB15K-237; **c** WN18;d)WN18RR

attack, the performance of knowledge graph embedding models has degenerated. However, our ensemble strategy significantly improves the performance of knowledge graph embedding models. As Figure 4 shows, with an increase in noise, the performance of all knowledge graph embedding approaches sharply decreases, indicating neural-based models are unstable to noisy training facts. In contrast, as the solid line shows, ensemble methods *KSE* and *AKSE* on *TransE*, *DistMult*, and *TransR* have a lower descending rate, demonstrating more robust performance in the event of noise increase. Additionally, both base learners *TransE*, *DistMult* and *TransR* are greatly enhanced by the ensemble strategy. *AKSE* shows superior performance than *KSE* in most cases. It reveals that our weight strategy is useful in terms of improving its prediction capacity by adaptively discounting weights assignment according to individual leaner prediction uncertainty. Further analysis and a visualization about *AKSE* are presented in Section 6.3.

6.2 Evaluation on extracted knowledge graph

Due to input data errors or extraction errors, bottom-up constructed KGs are polluted by many incorrect facts. In this section, we applied our ensemble methods on such noisy KGs.

Table 2 Link prediction on four standard benchmarks with 0%/30% injected noise. Bold face represent the best performance in column

Methods	FB15K		WN18		FB15K-237		WN18-RR	
	MR	Hit@10	MR	Hit@10	MR	Hit@10	MR	Hit@10
TransE	124/452	61.1/39.7	243/3038.9	89.7/49.9	253/674	42.3/22.4	3417/7021	49.8/23.8
TransR	117/476	71.2/42.7	239/3401	92.0/43.4	267/783	42.9/21.8	4267/8567	44.1/22.1
HoLE	97/331	70.4/46.3	224/2152.7	93.9/56.2	316/1102	39.6/15.4	4583/7315	43.6/18.9
ComplEx	78/235	81.9/44.2	231/1514.4	94.7/59.6	279/819	42.3/23.4	3175/5964	50.8/24.6
DistMult	81/277	83.6/51.6	215/2418.0	93.7/47.2	314.4/717	39.4/23.4	3684/5320	47.4/24.9
R-GCN	–	84.1/41.3	–	95.6/64.3	–	41.9/26.7	–	–
ConvE	69/282	83.1/43.6	374/1905.3	95.6/63.1	244/432	49.1/34.4	4187/8132	51.0/18.9
CKRL(LT)	141/584	64.8/43.7	336/2753	88.8/51.4	254/659	43.6/24.6	5321/10523	47.1/22.7
KSE(TransE)	102/185	67.8/60.3	248/674	88.7/73.7	247/**364**	44.8/37.6	3744/**4165**	**51.6**/39.5
AKSE(TransE)	96/167	68.1/62.1	223/549	92.1/76.3	**235**/373	**46.9/39.9**	**2803**/4302	51.4/**40.6**
KSE(TransR)	113/203	64.9/59.4	244/683	87.1/74.5	268/398	41.4/34.8	5349/5415	46.9/37.2
AKSE(TransR)	103/195	67.7/61.1	247/638	91.1/74.9	273/406	43.4/36.4	3820/4978	50.1/39.1
KSE(DistMult)	79/**134**	83.4/73.4	233/**513**	95.4/76.3	289/394	41.5/37.8	4015/4730	47.9/37.8
AKSE(DistMult)	**78**/141	**84.7/74.1**	**210**/565	**95.7/77.4**	264/377	42.1/38.4	4510/4902	48.8/39.5

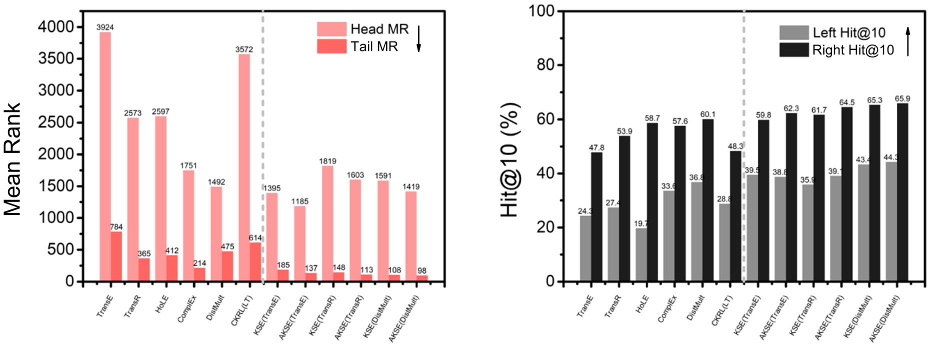

Figure 5 Link prediction on NELL79K, **a** Mean rank (MR), **b** Hit@10. Our methods are shown in the right of dash line

In order to eliminate the effect of sparsity, we filtered entities and relations whose degree were less than 3. Due to the large scale of entity set, we failed to apply recent state-of-the-art approaches (*ConvE*, *RGCN*) to NELL79K. Thus we only consider shallow models as the baselines. We applied our ensemble strategy to *TransE* and *DistMult*. The results are reported in Figure 5. As the results show, ensemble strategy yields an enhancement for base learners, where *AKSE(DistMult)* significantly outperforms other models. Notes that the unbalance of 1 to 1 and 1 to M relations cause the performance gap between left and right prediction results.

6.3 Visualization of the adaptive weighting

Our two link prediction experiments both show the superiority of *Adaptive weighting*. To further understand how *Adaptive weighting* works, we present a visualization of weight vector and corresponding Hit@10 metric in Figure 6. Our experiments conducted *AKSE*

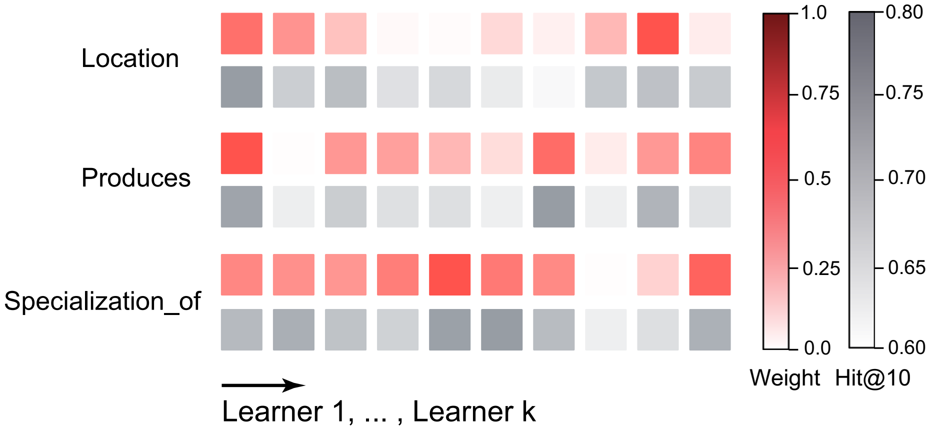

Figure 6 Visualization of weight and Hit@10 on relation, `location`, `Produce`, `Specialization_of`. Red square means assigned weights. Gray squares mean Hit@10 performance on the validation set

(DistMult) on FB15K with 30% injected noise. For intuition, we set k as 10 and selected three relation cases, `Location`, `Produce`, `Specialization_of`. We tested Hit@10 of all learners on the three relations on the validation set, shown as gray squares. Red square means the weights. As the results show, the learners exhibit different prediction capabilities. Meanwhile, a series of variant weights are assigned for each of individual learners according to learners' out-of-bag variance. By observing weights and metrics, the value of the weights is significantly correlated to the performance of the specific learner in most cases. Therefore, our method can capture the prediction capability of a single learner, and adaptively assign weights, finally resulting in better performance than *Simple average*. Additionally, *Adaptive weighting* is free of hyper-parameters and sufficient to utilize out-of-bag data.

6.4 Triplet classification

We conduct triplet classification on FB15K, WN18, and NELL79K. FB15K and WN18 are human-curated knowledge graphs, and NELL79K is a inborn noisy knowledge graph. Therefore random incorrect triplets noise were injected into FB15K and WN18. Our experimental results are shown in Table 3. Despite of no obvious improvement on noise-free KGs (FB15K, WN18), our methods exhibit superior performance on noisy datasets. It means our methods are more robust than the baselines, which could find the correct candidate sets with high plausibility.

6.5 Parameter analysis

The *AKSE* algorithm involves two key parameters. In this section, we examine how the different choices of parameters affect the performance of *AKSE* and *KSE* on FB15K dataset.

In order to examine how changes in the number of learners affects our proposed model's performance in the link prediction task, we trained a series of *KSE* and *AKSE* models, of

Table 3 Triplet classification on FB15K, WN18, and NELL79K. 0%/30% noise are injected into FB15K and WN18. Bold letters indicate a best performance in the column

Methods	FB15K	WN18	NELL79K
TransE	0.842/0.735	0.931/0.786	0.737
TransR	0.797/0.697	0.921/0.773	0.648
HoLE	0.81/0.714	0.946/0.803	0.749
ComplEx	**0.886**/0.756	**0.953**/0.813	0.783
DistMult	0.857/0.773	0.935/0.821	0.812
CKRL(LT)	0.731/0.647	0.873/0.834	0.778
KSE(TransE)	0.851/0.832	0.929/0.842	0.841
AKSE(TransE)	0.853/0.847	0.933/0.869	**0.873**
KSE(TransR)	0.841/0.827	0.913/0.839	0.829
AKSE(TransR)	0.848/0.821	0.923/0.847	0.821
KSE(DistMult)	0.859/0.843	0.941/**0.871**	0.842
AKSE(DistMult)	0.874/**0.862**	0.937/0.869	0.815

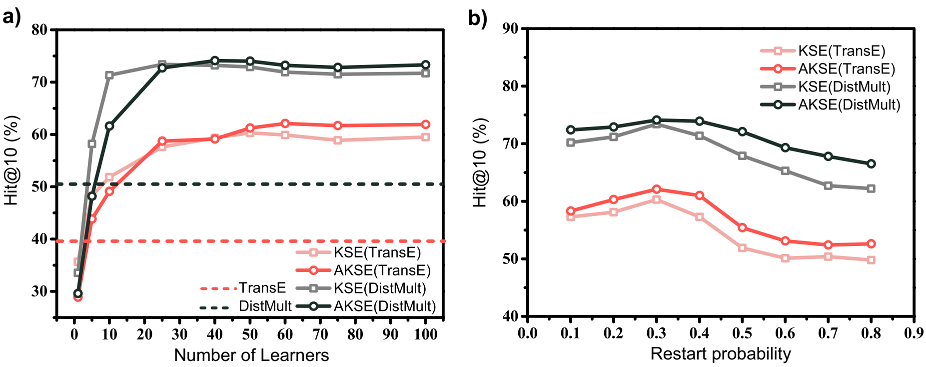

Figure 7 Parameter sensitivity of **a** number of learners and **b** restart probability

which k are ranging from 1 to 100. The other hyper-parameters are the same as the link prediction experiment. As Figure 7a shows, *AKSE* and*KSE* both present a rapid increase in Hit@10 with an increase in k. The optimized k of *DistMult*-based methods is 50 and *TransE*-based is 60. In particular, even using $\leqslant$ 10 learners, the robustness of the base learner can be significantly improved. In addition, *AKSE* outperforms KSE in the corresponding optimal k, the performance concerning which is shown in Table 2.

We also measure how the restart probability p affect the performance. The restart probability p control the likelihood of immediately revisiting a fact during sampling. Setting it to a low value enable to explore deep nodes in the facts network. On the other hand, if p is low, it would keep the walk in the local nodes. Figure 8 is a series of subgraphs that sampled from the original KG, which start with the triplet `(Walk The Line, language, Russian)`. When $p = 0.2$, the walk visits nodes which are further away from the starting triplet, indicating a DFS-like exploration. When $p = 0.5$, the walk shows a BFS-like exploration and visits neighboring nodes. The numerical results are shown in Figure 7b. We observe a performance peak around 0.3 both on *AKSE* and *KSE*. It can be explained as the exploration-exploitation trade-off.

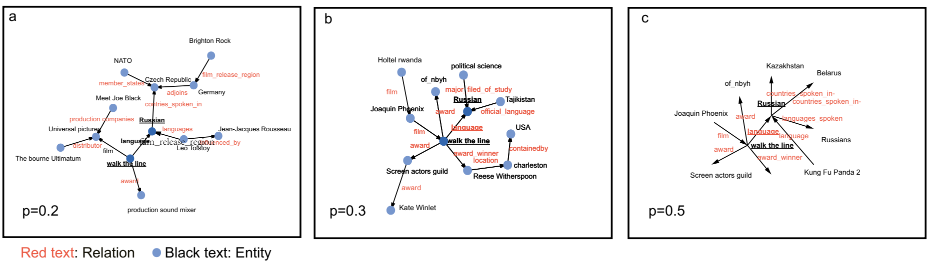

Figure 8 Visualization of the part of subgraph extraction procedure. Start with the triplet `(Walk The Line, language, Russian)`

7 Conclusion

In this paper, we analyzed the potential of multiple model ensemble to lower generalization error even with the presence of facts noise in KG. We proposed an ensemble strategy to enhance the robustness and trust of existing knowledge graph embedding approaches on noisy KGs. The strategy has two parts. 1) *Knowledge subgraph extraction*. The re-sampling method is flexible and structures sensitive in terms of constructing a new training set with high diversity. Meanwhile, it reduces the attack of noisy facts during the model training stage. 2) *Combination strategy*. We proposed *Adaptive weighting* to combine multiple shallow models according to their prediction capability adaptively . Experimental results show that the ensemble strategy significantly benefits the robustness of two classic models, *TransE*, *DistMult* and *TransR*.

In future, we will integrate our model into graph neural networks [24, 33] to learn better representation for knowledge graph completion.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China under Grants 61822113, the Natural Science Foundation of Hubei Province under Grants 2018CFA050.

References

1. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD, pp. 1247–1250 (2008)
2. Bordes, A., Usunier, N., García-durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
3. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI, pp. 1811–1818 (2018)
5. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
6. Fumera, G., Roli, F., Serrau, A.: A theoretical analysis of bagging as a linear combination of classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **30**(7), 1293–1299 (2008)
7. Galar, M., Fernández, A., Tartas, E.B., Sola, H.B., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. Part C **42**(4), 463–484 (2012)
8. Han, X., Cao, S., Lv, X., Lin, Y., Liu, Z., Sun, M., Li, J.: Openke: an open toolkit for knowledge embedding. In: EMNLP, pp. 139–144 (2018)
9. Kadlec, R., Bajgar, O., Kleindienst, J.: Knowledge base completion: baselines strike back. In: Rep4NLP@ACL, pp. 69–74 (2017)
10. Kolomiyets, O., Moens, M.F.: A survey on question answering technology from an information retrieval perspective. Inform. Sci. **181**(24), 5412–5434 (2011)
11. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, pp. 2181–2187 (2015)
12. Liu, A., Wang, W., Shang, S., Li, Q., Zhang, X.: Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica **22**(2), 335–362 (2018)
13. Liu, G., Liu, Y., Liu, A., Li, Z., Zheng, K., Wang, Y., Zhou, X.: Context-aware trust network extraction in large-scale trust-oriented social networks. World Wide Web **21**(3), 713–738 (2018)
14. Liu, G., Liu, Y., Zheng, K., Liu, A., Li, Z., Wang, Y., Zhou, X.: Mcs-gpm: multi-constrained simulation based graph pattern matching in contextual social graphs. IEEE Trans. Knowl. Data Eng. **30**(6), 1050–1064 (2017)
15. Liu, G., Wang, Y., Orgun, M.A.: Optimal social trust path selection in complex social networks. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
16. Liu, G., Wang, Y., Orgun, M.A., Lim, E.P.: Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks. IEEE Trans. Serv. Comput. **6**(2), 152–167 (2011)

17. Melville, P., Shah, N., Mihalkova, L., Mooney, R.J.: Experiments on ensembles with missing and noisy data. In: MCS, pp. 293–302 (2004)
18. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
19. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
20. Mitchell, T.M. et al.: Never-ending learning. Commun. ACM **61**(5), 103–115 (2018)
21. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)
22. Nickel, M., Rosasco, L., Poggio, T.A.: Holographic embeddings of knowledge graphs. In: AAAI, pp. 1955–1961 (2016)
23. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: ICML, pp. 809–816 (2011)
24. Pan, S., Hu, R., Fung, S.f., Long, G., Jiang, J., Zhang, C.: Learning graph embedding with adversarial training methods. arXiv:1901.01250 (2019)
25. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. Semantic Web **8**(3), 489–508 (2017)
26. Pujara, J., Augustine, E., Getoor, L.: Sparsity and noise: where knowledge graph embeddings fall short. In: EMNLP, pp. 1751–1756 (2017)
27. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC, pp. 593–607 (2018)
28. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. In: AAAI, pp. 4444–4451 (2017)
29. Trouillon, T., Dance, C.R., Gaussier, É., Welbl, J., Riedel, S., Bouchard, G.: Knowledge graph completion via complex tensor factorization. J. Mach. Learn. Res. **18**, 130:1–130:38 (2017)
30. Ueda, N., Nakano, R.: Generalization error of ensemble estimators. In: ICNN, vol. 1, pp. 90–95. https://doi.org/10.1109/ICNN.1996.548872 (1996)
31. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
32. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119 (2014)
33. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. arXiv:1901.00596 (2019)
34. Xiao, M., Ma, K., Liu, A., Zhao, H., Li, Z., Zheng, K., Zhou, X.: Sra: secure reverse auction for task assignment in spatial crowdsourcing. IEEE Transactions on Knowledge and Data Engineering (2019)
35. Xiao, M., Wu, J., Huang, L., Cheng, R., Wang, Y.: Online task assignment for crowdsensing in predictable mobile social networks. IEEE Trans. Mob. Comput. **16**(8), 2306–2320 (2016)
36. Xie, R., Liu, Z., Lin, F., Lin, L.: Does William Shakespeare REALLY Write Hamlet? Knowledge Representation Learning with Confidence. In: AAAI, pp. 4954–4961 (2018)
37. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
38. Zhai, D., Sun, Y., Liu, A., Li, Z., Liu, G., Zhao, L., Zheng, K.: Towards secure and truthful task assignment in spatial crowdsourcing. World Wide Web pp. 1–24 (2018)
39. Zhao, Y., Liu, G., Zheng, K., Liu, A., Li, Z., Zhou, X.: A context-aware approach for trustworthy worker selection in social crowd. World Wide Web **20**(6), 1211–1235 (2017)
40. Zhu, F., Wang, Y., Chen, C., Liu, G., Orgun, M.A., Wu, J.: A deep framework for cross-domain and cross-system recommendations. In: IJCAI, pp. 3711–3717 (2018)
41. Zhu, J., Jiang, W., Liu, A., Liu, G., Zhao, L.: Effective and efficient trajectory outlier detection based on time-dependent popular route. World Wide Web **20**(1), 111–134 (2017)
42. Zhu, X., Wu, X.: Class noise vs. attribute noise: a quantitative study. Artif. Intell. Rev. **22**(3), 177–210 (2004)

Affiliations

Guojia Wan[1] · Bo Du[1] · Shirui Pan[2] · Jia Wu[3]

✉ Shirui Pan
Shirui.Pan@monash.edu

Guojia Wan
guojiawan@whu.edu.cn

Jia Wu
jia.wu@mq.edu.au

[1] National Engineering Research Center for MultimediaSoftware, School of Computer Science, Wuhan University, Wuhan, China

[2] Faculty of Information Technology, Monash University, Melbourne, Australia

[3] Department of Computing, Macquarie University, Sydney, Australia