

# REA: Robust Cross-lingual Entity Alignment Between Knowledge Graphs

Shichao Pei

King Abdullah University of Science and  
Technology (KAUST), Thuwal, Saudi Arabia  
shichao.pei@kaust.edu.sa

Guoxian Yu

King Abdullah University of Science and  
Technology (KAUST), Thuwal, Saudi Arabia  
School of Software, Shandong University, China  
guoxian85@gmail.com

Lu Yu

King Abdullah University of Science and  
Technology (KAUST), Thuwal, Saudi Arabia  
lu.yu@kaust.edu.sa

Xiangliang Zhang\*

King Abdullah University of Science and  
Technology (KAUST), Thuwal, Saudi Arabia  
xiangliang.zhang@kaust.edu.sa

## ABSTRACT

Cross-lingual entity alignment aims at associating semantically similar entities in knowledge graphs with different languages. It has been an essential research problem for knowledge integration and knowledge graph connection, and been studied with supervised or semi-supervised machine learning methods with the assumption of *clean* labeled data. However, labels from human annotations often include errors, which can largely affect the alignment results. We thus aim to formulate and explore the robust entity alignment problem, which is non-trivial, due to the deficiency of noisy labels. Our proposed method named REA (Robust Entity Alignment) consists of two components: *noise detection* and *noise-aware entity alignment*. The *noise detection* is designed by following the adversarial training principle. The *noise-aware entity alignment* is devised by leveraging graph neural network based knowledge graph encoder as the core. In order to mutually boost the performance of the two components, we propose a unified reinforced training strategy to combine them. To evaluate our REA method, we conduct extensive experiments on several real-world datasets. The experimental results demonstrate the effectiveness of our proposed method and also show that our model consistently outperforms the state-of-the-art methods with significant improvement on alignment accuracy in the noise-involved scenario.

## CCS CONCEPTS

• **Information systems** → *Information integration*; • **Computing methodologies** → *Knowledge representation and reasoning*.

## KEYWORDS

Knowledge Graph; Entity Alignment; Noise Detection

\*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
KDD '20, August 23–27, 2020, Virtual Event, CA, USA

© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-7998-4/20/08...\$15.00  
<https://doi.org/10.1145/3394486.3403268>

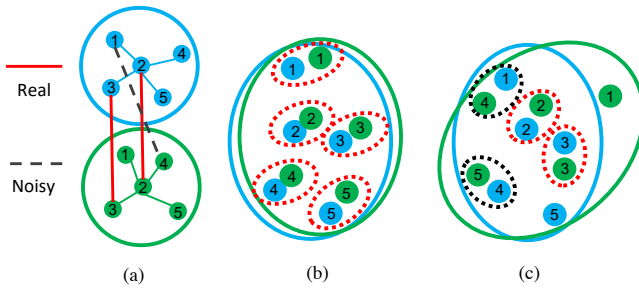
## ACM Reference Format:

Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2020. REA: Robust Cross-lingual Entity Alignment Between Knowledge Graphs. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*, August 23–27, 2020, Virtual Event, CA, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3394486.3403268>

## 1 INTRODUCTION

Knowledge graphs have been playing an essential role to organize and represent the knowledge collected from different domains. However, knowledge graphs (KG), even in the same domain, are typically constructed by different techniques with different languages. The gap of integrity between knowledge graphs and the demand of comprehensively representing the knowledge in a same domain motivate the study of integration of knowledge graphs. It is desirable to integrate multiple knowledge graphs in the same domain. In particular, cross-lingual entity alignment is a task of associating entities in knowledge graphs with different languages if they are semantically similar. Entity alignment has been studied widely in knowledge graphs connection and integration.

Existing methods are often designed to address the alignment problem in a supervised way with human-designed features [23] or with entity representations learned from KG embedding approaches [6, 7, 31, 38]. Also, a few semi-supervised methods [18, 26, 27] were proposed to make use of unlabeled data to enhance the performance of supervised entity alignment. Supervised or semi-supervised alignment methods have made remarkable discovery of semantically related entities. However, they rely on clean labeled entity pairs as the training set, which makes the model vulnerable to noises in the given labeled entity pairs. As shown in Figure 1, unsatisfactory alignment results would be obtained if the model is not aware of noisy labeled entity pairs. There are two sources of noises. 1) Erroneous entity pairs can be produced in the human-involved annotation process, e.g., on the crowd-sourcing or other annotation platforms. 2) Wrongly discovered entity pairs from the current stage of an entity alignment may be used for finding new aligned entity pairs in bootstrapping-based methods such as [32, 50]. These methods have been developed to propose new entity pairs to complement the set of clean labeled entity pairs, yet it is inevitable that the new proposed pairs contain noisy pairs.



**Figure 1: Toy example of entity alignment between knowledge graphs with noise-involved labeled entity pairs. (a) Red solid line denotes the correctly labeled entity pair, black dotted line denotes the noisy entity pair. Entities with the same ID number are those with the same semantic meaning and should be aligned. (b) Expected distribution of entities in an embedding space after alignment. The entities with the same ID number should be close. Red dotted cycles indicate the correct alignment results and black dotted cycles indicate the wrong alignment result. (c) Results from solutions that are not aware of the noisy entity pair (blue 1, green 4). The distance between (blue 1, green 4) is minimized, as a result the unsatisfactory result is conducted where real entity pair (blue 5, green 5) is not aligned but (blue 4, green 5) is aligned. Because the positions of entities in an embedding space are stable after KG embedding, minimizing the distance between (blue 1, green 4) will result in the rotation of embedding space as the green circle shown, the position of green 5 is moved to close blue 4, instead of blue 5. Thus, blue 4 and green 5 are wrongly aligned due to the smallest distance.**

Entity alignment models are expected to be robust to the noise in the given labeled entity pairs, meanwhile to capture the associations between entities in different knowledge graphs. However, existing approaches have quite limited efforts on the robustness study in this regard. To overcome this limitation of existing methods for cross-lingual entity alignment on handling noisy labeled entity pairs, in this paper, we exploit how to incorporate noise detection with entity alignment model, and how to jointly train them to meet the goal of associating entities with awareness of noises. Nevertheless, its materialization is non-trivial, due to two major challenges:

- No explicit knowledge about where the noise might be. It is costly and labor-expensive to distinguish if the labeled entity pair is noise. Therefore, the designed model should have the ability to detect the noisy pair without any explicit supervision. Although graph-based anomaly detection studies [1, 2] have been proposed to detect errors on nodes and edges using structure-based or community-based algorithms [3], these studies only focus on the single graph rather than multiple different graphs with links between them, and knowledge graphs with different languages usually do not have completely analogous structure due to the incompleteness.
- The desire of designing a unified model trained jointly for noise detection and cross-lingual entity alignment. On the one hand, noise detection is pretty helpful to provide reliable entity pairs for entity alignment. On the other hand, high-precise entity alignment model is beneficial to identify the noise. Yet, the difficulty on designing the expected unified model is that the high-precise entity alignment model is not available since it first needs

the clean labeled entity pairs as training data, and the noise is also hard to be detected since it requires the labeled noise data. Besides, devising the feedback from noise detection to entity alignment is problematic even the noise has been realized. Thus, designing the joint training strategy is a hard dilemma.

The aforementioned challenges motivate us to propose **REA**, a novel framework for robust cross-lingual entity alignment. In particular, we first employ graph neural network as the knowledge graph encoder to capture the structure information of knowledge graphs. Then we design a noise-aware entity alignment module based on the knowledge graph encoder and a margin-based ranking loss with trust score to align entities in given labeled entity pairs. The trust score indicates the likelihood of a labeled entity pair being real. Then we propose a novel noise detection module with Generative Adversarial Network (GAN) [10] based adversarial learning for the first challenge. It consists of two components, 1) noisy entity pair acquisition, which is a generator producing the noisy entity pairs that mimic real ones; and 2) noise discrimination, which is a discriminator distinguishing the real entity pairs from the generated noisy ones. Noisy entity pair generator iteratively learns to provide the most difficult cases for the noise discriminator to retrain itself. The two components play a minimax game to boost the capability of generating and distinguishing noise data.

Next, we design an iterative training strategy for the second challenge. The noisy entity pair generator needs to leverage the learned embeddings of entities to produce the noise data, and the trained noise discriminator can be utilized to provide the trust score for the entity pairs in the given labeled pairs. Hence, we first train the noise-aware entity alignment module with a knowledge graph encoder to update the entity embeddings, then use the learned embeddings to optimize the noise detection module. The trust score provided by the noise detection module can be fed back for training the noise-aware entity alignment in the next iteration. The mutually reinforced manner can boost the performance of both noise detection and noise-aware entity alignment.

Our contributions in this work are summarized as follows:

- We are the first to formulate and explore the problem of **robust cross-lingual entity alignment**, in order to detect noise in the given labeled data, then to provide clean data for cross-lingual entity alignment.
- We propose a novel framework REA that combines noise detection and entity alignment in a mutually reinforced manner, with a GAN-based adversarial noise detection module to overcome the deficiency of labeled noise data.
- We perform extensive experiments on several real-world datasets and show the effectiveness of our proposed model for the noise-involved entity alignment problem. Also results demonstrate that REA has the superior performance over the state-of-the-art methods, with significant improvement on noise-involved entity alignment accuracy.

## 2 RELATED WORK

In this section, we review the related works in entity alignment, anomaly detection and robust representation learning on graphs.

## 2.1 Entity Alignment

Entity alignment is an essential task for knowledge graph integration. The earliest works used crowdsourcing [17, 35] and well-designed hand-crafted features [23]. Later, extra resources are leveraged, such as OWL properties [13], entity descriptions [44], information of entities and relations [15, 25, 30]. However, such approaches are complex, labor-expensive, usually limited by the availability of the extra information.

Recently, embedding-based approaches become the most popular solution for entity alignment, involving translation-based and graph neural network based methods. 1) Translation-based methods. MTransE [7] is the pioneering work which encodes entities and relations of each KG in a separated embedding space, and learns transition matrices to map each embedding vector to its cross-lingual counterparts in the other space. Then several methods, such as ITransE [50], IPTransE [50] and BootEA [32] leverage the bootstrapping strategy to address the lack of labeled data. Further, semi-supervised approaches [18, 26, 27] were developed to utilize the unlabeled data for enhancing the performance of entity alignment. Moreover, several works consider to jointly model the structure and attribute information of KGs [6, 31, 34], or the structure and relation information [29, 51] or the structure, relation and attribute information together [49]; 2) Graph Neural Network-based methods. Recently, GCNs have been firstly used by [38] to embed entities of KGs into a unified vector space combining structure and attributes information. Then MuGNN [5] combined GNN with self attention to encode KGs, and contextual information also can be utilized with a GNN-based graph matching model [43]. Moreover, simple relation information can be encoded with the GNN models [40, 41, 45], and MRAEA [24] further takes the meta semantic information of relations into consideration.

All aforementioned works ignore the existence of noise in the given labeled entity pairs, and regard these pairs as completely clean data. Therefore, the noise in the labeled entity pairs would have the adverse influence on the performance of entity alignment, just as illustrated in Figure 1. Our work is the first to consider the robustness of entity alignment.

## 2.2 Anomaly Detection and Robust Representation Learning on Graphs

Noise in labeled entity pairs can be regarded as the anomalous edges between graphs. We first review some anomaly detection works on graphs. Graph-based anomaly detection [3] mainly focuses on plain graphs [1, 2], attributed graphs [9, 20] and dynamic graphs [46, 47]. However, the relevant methods only aim to detect the anomalies without iterative learning by using the detected anomalies to obtain more accurate representation.

Although confidence-based approaches [28, 42] were proposed to incorporate triple confidence into knowledge graph representation learning model to jointly detect the noise and learn the embedding, this kind of methods cannot be easily used in our entity alignment scenario. The reason is that our task is to detect the noisy links between two knowledge graphs rather than triples within a single knowledge graph. Thus, above methods are not appropriate for the purpose of designing a robust entity alignment model.

Recently, generative adversarial networks (GANs) [10] have demonstrated its superior performance in many tasks [21, 37]. A

few methods leverage the adversarial training to learn more robust network representation [8, 12, 36, 48]. However, these methods cannot deal with the explicit noise in labeled entity pairs and also cannot be directly used for the entity alignment scenario.

## 3 PRELIMINARIES

In this section, we give some related definitions and formalize the problem of robust cross-lingual entity alignment.

### Cross-lingual Knowledge Graphs

A knowledge graph can be denoted as  $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ , where  $\mathcal{E}$  is the set of entities,  $\mathcal{R}$  is the set of relations, and  $\mathcal{T}$  is the set of triples, each of which is a triple  $(e_x, r, e_y)$ , including the head entity  $e_x$ , the relation  $r$  and the tail entity  $e_y$ . By KG embedding, each triple can be presented as  $(\mathbf{e}_x, \mathbf{r}, \mathbf{e}_y)$ , in which boldfaced  $\mathbf{e}_x$ ,  $\mathbf{r}$ , and  $\mathbf{e}_y$  represent the embedding vectors of head  $e_x$ , relation  $r$ , and tail  $e_y$ , respectively. Cross-lingual knowledge graphs are a set of KGs with the language set  $LA$  denoted as  $\mathcal{G}_{LA} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_i\}$ , where  $\mathcal{G}_i$  denotes the KG with language  $i \in LA$ .

### Cross-lingual Entity Alignment

In our work, we only consider the 1-to-1 entity alignment between two cross-lingual KGs. Let  $\mathcal{G}_i = (\mathcal{E}_i, \mathcal{R}_i, \mathcal{T}_i)$  and  $\mathcal{G}_j = (\mathcal{E}_j, \mathcal{R}_j, \mathcal{T}_j)$  be two KGs in different language  $i$  and  $j$ . A set of labeled entity pairs (pre-aligned seeds) are given in  $\mathcal{AS} = \{(e_i, e_j) | e_i \in \mathcal{E}_i^L, e_j \in \mathcal{E}_j^L\}$ , where  $e_i$  in  $\mathcal{G}_i$  shares same meaning with its counterpart  $e_j$  in  $\mathcal{G}_j$ . Entity alignment is a task to find and align the remaining entities  $\{e_i \in \mathcal{E}_i^U\}$  and  $\{e_j \in \mathcal{E}_j^U\}$  which share same meaning, where  $\mathcal{E}_i^U = \mathcal{E}_i \setminus \mathcal{E}_i^L$  and  $\mathcal{E}_j^U = \mathcal{E}_j \setminus \mathcal{E}_j^L$ .

### Robust Cross-lingual Entity Alignment

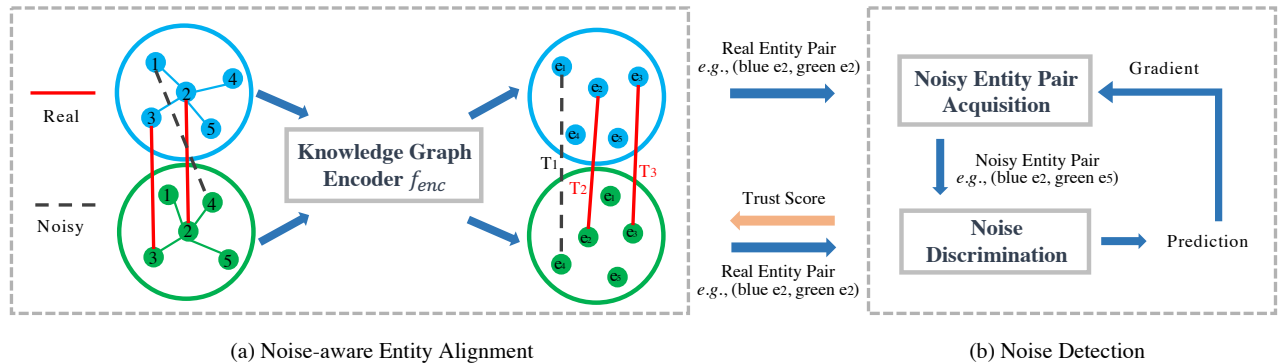
In the noise-involved scenario,  $\mathcal{AS}$  is not completely correct because of the existence of mistakes in the labeled entity pair collection process. Let  $\mathcal{AS}^T$  be a set of trusted entity pairs which are noise-free entity pairs and all correctly labeled to have the same semantic meaning, and  $\mathcal{AS}^T$  is known for alignment models. We denote  $\mathcal{AS}^U = \mathcal{AS} \setminus \mathcal{AS}^T$  as a set of unreliable labeled entity pairs, which include real and noisy entity pairs but unknown for alignment models. More specifically, alignment models only receive  $\mathcal{AS}^T$  as the trusted real entity pairs, and are not aware of which pairs in  $\mathcal{AS}^U$  are real or noise. Given noise-involved entity pairs  $\mathcal{AS}^U$  and trusted entity pairs  $\mathcal{AS}^T$ , robust entity alignment is to find and align the remaining entities with the awareness of noise in  $\mathcal{AS}^U$ .

## 4 THE PROPOSED APPROACH: REA

In this section, we present the proposed method REA, a novel robust cross-lingual entity alignment framework. We start with the overall framework, followed by elaborations on each component of our framework. Lastly, we discuss the optimization and inference.

### 4.1 Overall Framework of REA

As shown in Figure 2, our proposed framework consists of two modules, **noise-aware entity alignment** module and **noise detection** module. First, we develop a GNN-based encoder to encode the structure information of two given knowledge graphs. Then, a noise-aware entity alignment module is designed based on the



**Figure 2: Framework of the proposed method for robust cross-lingual entity alignment.** The framework consists of noise-aware entity alignment module and noise detection module. The noise-aware entity alignment module provides the embeddings of entities for noise detection module, which also feeds back the trust score to alignment module for updating embeddings.  $T_1, T_2$  and  $T_3$  denote the trust score for labeled entity pairs. In detail, noisy pair acquisition utilizes the real ones in  $\mathcal{AS}^T$  to generate noisy entity pairs while noise discrimination tries to distinguish the real pairs from generated ones. The gradient from noise discrimination will update the noisy pair acquisition.

encoder to minimize the distance between two entities in a labeled entity pair with a trust score in order to avoid the noise. Second, for the noise detection module, we first have to solve the demand of noisy entity pair acquisition due to the lack of labeled noise data, then noise discrimination is required to distinguish the real and noisy entity pairs. Meanwhile, two components in noise detection are desired to interact to make use of the noisy entity pair acquisition and the feedback from the noise discrimination to guide the optimization of itself. Next, we describe each module in detail.

## 4.2 Noise-aware Entity Alignment

**4.2.1 Knowledge Graph Encoder.** Inspired by the superior performance of GNN-based methods [5, 38] over translation-based methods, we build our model based on the basic GNN [14], as we concentrate on the awareness of noise in alignment problem, rather than KG encoding. A knowledge graph encoder maps entities of knowledge graph  $\mathcal{G}_i$  and  $\mathcal{G}_j$  into a same embedding space. We define the encoder as:

$$\mathbf{E}^i, \mathbf{E}^j = f_{enc}(\mathcal{G}_i, \mathcal{G}_j) \quad (1)$$

where  $\mathbf{E}^i$  is the embedding of entities in  $\mathcal{G}_i$ ,  $\mathbf{E}^j$  is the embedding of entities in  $\mathcal{G}_j$ , and  $f_{enc}$  is the encoding model. Taking knowledge graph  $\mathcal{G}_i$  as an example, we build upon the message-passing architecture of GNN [11] to capture contextual information along the graph structure and refine the distributional representation of entities. We encode graph  $\mathcal{G}_j$  in the same way with the same GNN in order to embed both graphs into a same embedding space. Next, we illustrate the design of high-order propagation which benefits the graph representation learning by stacking more propagation layers.

For a connected entity-relation-entity triple  $(e_x, r, e_y)$  in graph  $\mathcal{G}_i$ , we define the message from  $e_x$  to  $e_y$  at  $l$ -th propagation as:

$$\mathbf{m}_{y \leftarrow x}^l = f_m(\mathbf{e}_x^{l-1}, \mathbf{e}_y^{l-1}, p_{xy}) \quad (2)$$

where  $\mathbf{e}_x^{l-1}$  and  $\mathbf{e}_y^{l-1}$  are the embedding of  $e_x$  and  $e_y$  generated from the  $(l-1)$ -th message propagation, and they also contain the messages propagated from its  $(l-1)$ -hop neighbors,  $\mathbf{m}$  is the information

to be propagated and  $\mathbf{m}_{y \leftarrow x}^l$  is the message embedding. Function  $f_m$  is the message encoder which takes  $\mathbf{e}_x^{l-1}$  and  $\mathbf{e}_y^{l-1}$  as the input. We do not consider the relation  $r$  in  $f_m$ , following the work in [38] for simplicity. A decay factor  $p_{xy}$  is introduced to control the message propagation over the path length. We set  $p_{xy}$  as the graph Laplacian norm  $1/\sqrt{|\mathcal{N}_x| |\mathcal{N}_y|}$  following GCN [14], where  $\mathcal{N}_x$  and  $\mathcal{N}_y$  refer to the first-hop neighbors of entity  $e_x$  and  $e_y$ , respectively. Then,  $f_m$  is implemented as follow:

$$\mathbf{m}_{y \leftarrow x}^l = \frac{1}{\sqrt{|\mathcal{N}_x| |\mathcal{N}_y|}} (\mathbf{W}_1^l \mathbf{e}_x^{l-1} + \mathbf{W}_2^l (\mathbf{e}_x^{l-1} \odot \mathbf{e}_y^{l-1})) \quad (3)$$

where  $\mathbf{W}_1^l$  and  $\mathbf{W}_2^l$  are trainable weight parameters. We design in Eq. (3) that the propagated message not only depends on the  $e_x$ , like the conventional GCN models, but also encodes the affinity between  $e_x$  and  $e_y$ , with  $\odot$  indicating the element-wise product in Eq. (3). This operator encodes the message with more information about the pair, instead of only the source entity.

Next we define the message aggregation operator. In particular, we aggregate messages propagated from  $j$ 's neighborhood to update  $j$ 's representation as follow:

$$\mathbf{e}_y^l = \text{LeakyReLU}(\mathbf{m}_{y \leftarrow y}^l + \sum_{x \in \mathcal{N}_y} \mathbf{m}_{y \leftarrow x}^l) \quad (4)$$

where message passed over the activation function of LeakyReLU [22], and  $\mathbf{e}_y^l$  denotes the embedding of entity  $j$  obtained after the  $l$ -th propagation layer. All messages coming from  $\mathcal{N}_y$  (neighbors of  $y$ ) are aggregated in Eq. (4). Meanwhile, we take the self-connection of  $y$  into consideration to maintain original feature information. The self-connection message is:

$$\mathbf{m}_{y \leftarrow y}^l = \mathbf{W}_1^l \mathbf{e}_y^{l-1} \quad (5)$$

where  $\mathbf{W}_1^l$  is the same weight parameter with that in Eq. (3).

With the representation  $\mathbf{e}_y^l$  obtained from  $l$ -th propagation layer, we regard it as the representation for entity  $e_y$ , the representations of other entities in  $\mathcal{G}_i$  can be obtained in the same way.

**4.2.2 Objective of Noise-aware Entity Alignment.** After obtaining entity embeddings of graph  $\mathcal{G}_i$  and  $\mathcal{G}_j$  from GNN-based encoder, entities in a labeled entity pair in  $\mathcal{AS}$  should be forced as close as possible in the feature space due to the same semantic meaning. For instance,  $\mathbf{e}_x \approx \mathbf{e}_y$  when  $(e_x, e_y)$  holds, so the labeled entity pairs can be regarded as the training set to train the knowledge graph encoder. However, most of all existing works assume that all labeled entity pairs are correct. It is inevitable that noises exist in the labeled entity pairs owing to different labeled pair collection processes as discussed before. Inspired by confidence-based methods [28, 42], we introduce a **trust score** into our proposed model to describe the likelihood of a labeled entity pair  $(e_x, e_y)$  being real. Hence, the margin-based ranking objective function of our noise-aware entity alignment can be defined with the trust score as follows:

$$\mathcal{L}_{EA} = \sum_{\substack{(e_x, e_y) \\ \in \mathcal{AS}}} \sum_{\substack{(e'_x, e'_y) \\ \in \mathcal{AS}'_{(e_x, e_y)}}} \text{TS}(e_x, e_y) [\gamma + f_d(\mathbf{e}_x, \mathbf{e}_y) - f_d(\mathbf{e}'_x, \mathbf{e}'_y)]_+ \quad (6)$$

where  $\text{TS}(e_x, e_y)$  indicates the trust score for entity pair  $(e_x, e_y)$ ,  $[x]_+ = \max\{0, x\}$  denotes the positive part of  $x$ , and  $\gamma$  is a margin hyper-parameter which is greater than 0, and  $f_d(\cdot)$  refers to the energy function:

$$f_d(\mathbf{e}_x, \mathbf{e}_y) = \|\mathbf{e}_x - \mathbf{e}_y\|_1 \quad (7)$$

and  $\mathcal{AS}'$  denotes the negative sample set for the entity pair  $(e_x, e_y)$ :

$$\mathcal{AS}'_{(e_x, e_y)} = \{(e'_x, e_y) | e'_x \in \mathcal{E}_i\} \cup \{(e_x, e'_y) | e'_y \in \mathcal{E}_j\} \quad (8)$$

where  $(e'_x, e_y)$  and  $(e_x, e'_y)$  are the Bernoulli negative-sampled entity pairs by replacing  $e_x$  or  $e_y$  in  $(e_x, e_y)$ . Then, we will describe how to obtain the trust score in the ensuing discussion.

### 4.3 Noise Detection

With the objective of noise detection, we first need to overcome the deficiency of noisy labeled entity pairs by designing a strategy to obtain the noisy entity pairs. Then a discrimination model is required to distinguish the obtained noisy entity pairs from the labeled real entity pairs, as such the discrimination model can be employed to classify entity pairs in  $\mathcal{AS}$  after convergence. In addition, to allow the feedback signal to guide the optimization of noise acquisition strategy, it is indispensable to develop a joint model to iteratively train both of them, because the obtained noisy entity pairs can assist the update of the discrimination model. Contrariwise, the discrimination model also can guide the optimization of the noise acquisition strategy to acquire the noisy entity pairs, which are most difficult to be distinguished for the discrimination model. The interactive way can improve the noisy entity pair acquisition and noise discrimination iteratively.

**4.3.1 Overall Objective.** Illuminated by GAN [10], we can concrete our idea by unifying the noise acquisition and the noise discrimination via a minimax game. Formally, we define the game as follows:

$$\mathcal{L}_{ND} = \max_{\varphi} \min_{\theta} \sum_{(e_x, e_y) \in \mathcal{AS}^T} \mathbb{E}_{(e_x, e_y) \sim \mathcal{AS}^T} [\log D((e_x, e_y); \varphi)] + \mathbb{E}_{(e'_x, e'_y) \sim G(\cdot | (e_x, e_y); \theta)} [\log(1 - D((e'_x, e'_y); \varphi))] \quad (9)$$

where  $D(\cdot; \varphi)$  is the noise discrimination model with the real entity pairs  $(e_x, e_y)$  or generated noisy entity pairs  $(e'_x, e'_y)$  as the input.  $G(\cdot | (e_x, e_y); \theta)$  denotes the noisy entity pairs acquisition function which generates the noisy pairs  $(e'_x, e'_y)$  with the given real entity pair  $(e_x, e_y)$ . Besides, real entity pairs are sampled from the set of trusted entity pairs  $\mathcal{AS}^T$ , following the similar operation in works [16, 19]. Next, we illustrate each component in detail.

**4.3.2 Noisy Entity Pair Acquisition.** Due to the lack of clear indication on which labeled entity pairs in  $\mathcal{AS}^U$  are wrong, we exploit to generate the requisite noisy entity pairs. The entity pair generator should have the ability to fit the underlying relevance distribution over entity pairs, and sample the most "real" entity pairs from a set of negative pairs based on the relevance distribution for each pair in  $\mathcal{AS}^T$  as the noisy (negative) samples to fool the noise discriminator. More specifically, entity pair generator makes use of the entity embeddings of  $(e_x, e_y)$  learned from noise-aware entity alignment module to generate the noisy pairs  $(e'_x, e'_y)$  which are as close as possible to the real one in underlying distribution, in order to minimize the log-likelihood of  $(e'_x, e'_y)$  being recognized as the noise with  $D(\cdot; \varphi)$  fixed. The noisy entity pair generator can be optimized as follows:

$$\theta^* = \arg \min_{\theta} \sum_{(e_x, e_y) \in \mathcal{AS}^T} \underbrace{\mathbb{E}_{(e'_x, e'_y) \sim G(\cdot | (e_x, e_y); \theta)} [\log(1 - D((e'_x, e'_y); \varphi))]}_{\mathcal{L}_G(e_x, e_y)} \quad (10)$$

where  $\mathcal{L}_G(e_x, e_y)$  is the loss function for one entity pair  $(e_x, e_y)$ .

We define the probability of generating a noisy entity pair  $(e'_x, e'_y)$  as follows:

$$G((e'_x, e'_y) | (e_x, e_y); \theta) = \frac{\exp(f_{\theta}(e'_x, e'_y))}{\sum \exp(f_{\theta}(e'_x, e'_y))} \quad (11)$$

$$(e_x^*, e_y^*) \in N(e_x, e_y) \subset \mathcal{AS}'_{(e_x, e_y)} \quad (12)$$

where  $f_{\theta}(x, y)$  is a two-layer neural network with ReLU as activation function, and its input is  $\|x - y\|_1$ . The probability of generating  $(e'_x, e'_y)$  is based on the probability of generating all  $(e_x^*, e_y^*)$  in  $N(e_x, e_y)$ . Ideally,  $N(e_x, e_y)$  should contain all possible negative samples for  $(e_x, e_y)$ . However, to reduce the computational load, we use a small subset of  $\mathcal{AS}'_{(e_x, e_y)}$  as the set of negative samples. In addition, multinomial sampling is employed to sample  $(e'_x, e'_y)$  from  $G(\cdot | (e_x, e_y); \theta)$ .

Because the generated entity pair  $(e'_x, e'_y)$  from generator is discrete, it cannot be optimized by gradient descent based optimizer. A common solution is to use policy gradient based reinforcement learning algorithm [37, 39]. Its gradient of  $\mathcal{L}_G$  for entity pair  $(e_x, e_y)$  can be derived as follows:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_G(e_x, e_y) &= \nabla_{\theta} \mathbb{E}_{(e'_x, e'_y) \sim G(\cdot | (e_x, e_y); \theta)} [\log(1 - D((e'_x, e'_y); \varphi))] \\ &= \mathbb{E}_{(e'_x, e'_y) \sim G(\cdot | (e_x, e_y); \theta)} [\nabla_{\theta} \log G(\cdot | (e_x, e_y); \theta) \log(1 - D((e'_x, e'_y); \varphi))] \\ &\approx \frac{1}{K} \sum_K \nabla_{\theta} \log G(\cdot | (e_x, e_y); \theta) \log(1 - D((e'_x, e'_y); \varphi)) \end{aligned} \quad (13)$$

where the last approximate equality means that we employ a sampling approximation by sampling  $k$  negative entity pairs  $(e'_x, e'_y)$

from the current generator. More specifically,  $(e_x, e_y)$  can be regarded as the *state*,  $G(\cdot|(e_x, e_y))$  is the *policy*,  $(e'_x, e'_y)$  is the *action*,  $\log(1 - D((e'_x, e'_y); \varphi))$  is the *reward*. Hence, the generator, an *agent*, can interact with discriminator - the *environment* by performing actions according to the current states and the policy, then update itself by maximizing the reward from environment as the response for performed actions. Besides, we introduce a widely-used baseline [33] into the reward function to reduce the variance, and update the reward as follows:

$$\mathcal{R}_G = \log(1 - D((e'_x, e'_y); \varphi)) - \mathbb{E}_{(e'_x, e'_y) \sim G(\cdot|(e_x, e_y); \theta)} [\log(1 - D((e'_x, e'_y); \varphi))] \quad (14)$$

where the term  $\mathbb{E}_{(e'_x, e'_y) \sim G(\cdot|(e_x, e_y); \theta)} [\log(1 - D((e'_x, e'_y); \varphi))]$  is the baseline function in policy gradient.

Moreover, the noisy entity pairs generated from the generator can also be used as the negative samples to replace the random negative sample in Eq. (8), and to boost the performance of noise-aware entity alignment module.

**4.3.3 Noise Discrimination.** The goal for noise discriminator is to distinguish the real entity pair and noisy entity pair from the noise-involved entity pairs  $\mathcal{AS}^U$ . To obtain training data to update the noise discriminator, we first utilize the samples generated from entity pair generator as negative samples, then regard the entity pairs from  $\mathcal{AS}^T$  as positive samples. Then we define the noise discriminator as a binary classifier. The objective function of the discriminator is to maximize the log-likelihood of correctly distinguishing the positive pairs from generated negative pairs with  $G(\cdot|(e_x, e_y); \theta)$  fixed. The optimization of noise discriminator is defined as follows:

$$\varphi^* = \arg \max_{\varphi} \sum_{(e_x, e_y) \in \mathcal{AS}^T} \mathbb{E}_{(e_x, e_y) \sim \mathcal{AS}^T} [\log D((e_x, e_y); \varphi)] + \mathbb{E}_{(e'_x, e'_y) \sim G(\cdot|(e_x, e_y); \theta)} [\log(1 - D((e'_x, e'_y); \varphi))] \quad (15)$$

$$D((e_x, e_y); \varphi) = \sigma(f_{\varphi}(e_x, e_y)) = \frac{\exp(f_{\varphi}(e_x, e_y))}{\exp(f_{\varphi}(e_x, e_y)) + 1} \quad (16)$$

where  $f_{\varphi}(x, y)$  is a two-layer neural network with ReLU as activation function, and its input is  $\|x - y\|_1$ ;  $\sigma(x)$  is the Sigmoid function. Because  $f_{\varphi}$  is differentiable with respect to  $\varphi$ , the objective function can be updated by stochastic gradient descent.

Next, we can define the trust score mentioned in Section 4.2.2 as follows:

$$\text{TS}(e_x, e_y) = \begin{cases} 1 & \sigma(f_{\varphi}(e_x, e_y)) \geq \delta \\ 0 & \sigma(f_{\varphi}(e_x, e_y)) < \delta \end{cases} \quad (17)$$

where  $\sigma(f_{\varphi}(e_x, e_y))$  is the output of discriminator,  $\delta$  is a threshold value which separates  $\mathcal{AS}^U$  as two sets, one is regarded as a set of real entity pairs, and another one is distinguished as a set of noisy entity pairs. Therefore, we set  $\text{TS}(e_x, e_y)$  as a binary score which indicates whether the entity pair is real or not. In addition, the set  $\mathcal{AS}^T$  will be augmented according to the change of trust scores of entity pairs in  $\mathcal{AS}^U$ . We will describe it in next section.

## 4.4 Optimization and Inference

We employ the iterative optimization strategy to train our proposed model REA. We first initialize the trust scores of real entity pairs in

**Table 1: Statistics of DBP15K and DWY100K.**

Datasets		Entities	Relations	Rel. triples
DBP15K <sub>ZH-EN</sub>	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
DBP15K <sub>JA-EN</sub>	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
DBP15K <sub>FR-EN</sub>	French	66,858	1,379	192,191
	English	105,889	2,209	278,590
DWY100K <sub>WD</sub>	DBpedia	100,000	330	463,294
	Wikidata	100,000	220	448,774
DWY100K <sub>YG</sub>	DBpedia	100,000	302	428,952
	YAGO3	100,000	31	502,563

$\mathcal{AS}^T$  as 1, and noise-involved entity pairs in  $\mathcal{AS}^U$  as 0. In each iteration, we first train the noise-aware entity alignment module to obtain the embedding for entities, then alternatively train entity pair generator and noise discriminator. Specifically, with the discriminator  $D$  fixed, the generator  $G$  is optimized to generate noisy entity pairs to fool  $D$ ; with  $G$  fixed,  $D$  is optimized to improve the performance of distinguishing the real entity pairs from noise ones. At the end of each iteration, we leverage the trained noise discriminator to update the trust score of each entity pair in  $\mathcal{AS}^U$ , then complement  $\mathcal{AS}^T$  using the entity pairs  $(e_x, e_y)$  in  $\mathcal{AS}^U$  with trust score  $\text{TS}(e_x, e_y) = 1$ . With the new trust scores, the embedding of entities can be updated by training the noise-aware entity alignment module in next iteration. We repeat the above process with enough iterations until the whole model converges.

The overall optimization process of our model REA is outlined in Algorithm 1 which is shown in the Appendix of the reproducibility supplement. The embeddings of KGs and parameter matrices are initialized by drawing from a truncated normal distribution. SGD is employed as the optimizer and all embeddings are normalized by  $L_2$  norm. In the inference stage, for an entity  $e$  in  $\mathcal{G}_i$ , we compute the distances between  $e$  and all the entities in  $\mathcal{G}_j$ , then output a list of ranked entities in  $\mathcal{G}_j$ , and select the top one as the aligned entity. Similarly, an entity  $e$  in  $\mathcal{G}_j$  can also be aligned to  $\mathcal{G}_i$  by the same inference step.

## 5 EXPERIMENTS

In this section, we conduct experiments on several real-world datasets with different sizes, and evaluate our proposed framework for noise-involved entity alignment.

### 5.1 Experimental Setup

**5.1.1 Datasets.** We used five cross-lingual knowledge graphs datasets from the DBP15K and the DWY100K following [31, 32, 38]. DBP15K consists of three different cross-lingual knowledge graphs which are DBP15K<sub>ZH-EN</sub>, DBP15K<sub>JA-EN</sub> and DBP15K<sub>FR-EN</sub>. Each dataset contains 15,000 clean labeled entity pairs. For DBP100K, it contains two large-scale datasets extracted from DBpedia, Wikidata and YAGO3, denoted by DWY100K<sub>WD</sub> and DWY100K<sub>YG</sub>. Each dataset contains 100,000 clean labeled entity pairs. Table 1 outlines the detail information of the datasets.

**5.1.2 Dataset Preprocessing.** To inject noise in above datasets for our experiments, for all datasets, we randomly split 30% of the

**Table 2: Noise-involved entity alignment performance comparison on five real-world datasets with widely-used evaluation metrics Hits@K and MRR (the best results of baselines are marked as \*). The results in bold are our framework.**

Methods	DBP15K <sub>ZH-EN</sub>			DBP15K <sub>JA-EN</sub>			DBP15K <sub>FR-EN</sub>			DWY100K <sub>WD</sub>			DWY100K <sub>YG</sub>		
	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR
MTransE	0.169	0.362	0.216	0.148	0.345	0.198	0.143	0.338	0.192	0.154	0.325	0.203	0.137	0.318	0.186
ITransE	0.185	0.394	0.258	0.174	0.386	0.242	0.181	0.402	0.269	0.193	0.414	0.296	0.158	0.346	0.223
GCN-Align	0.223	0.424	0.316	0.223	0.439	0.321	0.231	0.462	0.337	0.293	0.461	0.372	0.354	0.505	0.428
AlignEA	0.263	0.457	0.342	0.254	0.451	0.338	0.278	0.471	0.357	0.331	0.487	0.392	0.376	0.525	0.448
MuGNN	0.274*	0.471*	0.361*	0.279*	0.481*	0.368*	0.284*	0.485*	0.372*	0.348*	0.503*	0.417*	0.401*	0.554*	0.475*
REA-KE	0.235	0.437	0.319	0.236	0.451	0.334	0.229	0.456	0.332	0.312	0.468	0.379	0.352	0.513	0.432
<b>REA</b>	<b>0.289</b>	<b>0.486</b>	<b>0.380</b>	<b>0.293</b>	<b>0.498</b>	<b>0.388</b>	<b>0.304</b>	<b>0.539</b>	<b>0.403</b>	<b>0.368</b>	<b>0.547</b>	<b>0.444</b>	<b>0.426</b>	<b>0.577</b>	<b>0.494</b>
Improv. best %	5.47	3.18	5.26	5.02	3.53	5.43	7.04	11.13	8.33	5.75	8.75	6.47	6.23	4.15	4.00
Improv. REA-KE %	22.98	11.21	19.12	24.15	10.42	16.17	32.75	18.20	21.39	17.95	16.89	17.15	21.02	12.48	14.35

given clean labeled entity pairs as the training set  $\mathcal{AS}$ , and leave the remaining as testing data.

Then, we randomly split 60% of the training set  $\mathcal{AS}$  as the noise-free entity pairs  $\mathcal{AS}^P = \{(e_i, e_j) | e_i \in \mathcal{E}_i^P, e_j \in \mathcal{E}_j^P\}$  and replace the rest 40% with a set of corrupted entity pairs  $\mathcal{AS}^N$ . Specifically, we replace each entity pair  $(e_1, e_2)$  in the rest set with  $(e'_1, e'_2)$  which is sampled from  $\{(e'_1, e_2) | e'_1 \in \mathcal{E}_i^N\} \cup \{(e_1, e'_2) | e'_2 \in \mathcal{E}_j^N\}$  where  $\mathcal{E}_i$  is the set of entity of KG  $\mathcal{G}_i$ ,  $\mathcal{E}_j$  is the set of entity of KG  $\mathcal{G}_j$  and  $\mathcal{E}^N = \mathcal{E}^L \setminus \mathcal{E}^P$ . So we obtain the noise-free entity pairs  $\mathcal{AS}^P$  and noisy entity pairs  $\mathcal{AS}^N$ . Next, we randomly sample 55% of entity pairs in  $\mathcal{AS}^P$  to form the set of trusted entity pairs  $\mathcal{AS}^T$ .

Overall, the set of trusted entity pairs  $\mathcal{AS}^T$  can be utilized as positive samples to train the noise detection module, and the set  $\mathcal{AS}^U = \{\mathcal{AS}^P \setminus \mathcal{AS}^T\} \cup \mathcal{AS}^N$  denotes the union set of the real and noisy entity pairs, which are unknown for alignment models. The responsibility of noise detection is to distinguish real and noisy entity pairs from the union set. In addition, with little abuse of notion  $\mathcal{AS}$ , we denote  $\mathcal{AS}$  as  $\mathcal{AS}^T \cup \mathcal{AS}^U$  in other sections of the paper. For later parameter analysis, we follow the same procedure to prepare the different datasets for experiments.

**5.1.3 Baselines.** There is no existing work focusing on the noise-aware entity alignment, we compare our proposed framework REA with some classical entity alignment methods to evaluate the effectiveness of our proposed framework. We consider two categories of entity alignment methods: Translation-based (MTransE, ITransE, AlignEA), GCN-based (GCN-Align, MuGNN). Although the relation and attribute in knowledge graphs are also useful for enhancing the performance, our work focuses on the robust entity alignment, rather than developing a better encoder for cross-lingual knowledge graphs. Hence, we only consider structure information of knowledge graphs for simplicity, and only leverage the structure information for baselines for fair comparison. The brief introduction of baselines are as follows:

- MTransE [7]: a representative translation-based work that encodes entities of each KG in a separated embedding space and learns the translation matrix to map them.
- ITransE [50]: a TransE-based [4] method that encodes entities of KGs in a unified embedding space by an iterative and parameter sharing method.

**Table 3: Evaluation results on noise detection (Binary classification on labeled entity pairs  $\mathcal{AS}^U$ ) with 20% and 40% of  $\mathcal{AS}$  as noise.**

Noise %	20%			40%		
	Measures	Pre.	Rec.	F1	Pre.	Rec.
DBP15K <sub>ZH-EN</sub>	0.935	0.731	0.820	0.871	0.760	0.812
DBP15K <sub>JA-EN</sub>	0.929	0.714	0.808	0.874	0.738	0.800
DBP15K <sub>FR-EN</sub>	0.930	0.750	0.831	0.867	0.778	0.820
DWY100K <sub>WD</sub>	0.944	0.733	0.825	0.859	0.811	0.834
DWY100K <sub>YG</sub>	0.930	0.740	0.824	0.848	0.817	0.832

- AlignEA [32]: a translation-based method which applies truncated negative sampling and parameter swapping to enhance its performance.
- GCN-Align [38]: a GCN-based method that employs GCNs to encode the structure of KGs.
- MuGNN [5]: a multi-channel GNN-based method which performs KG inference and alignment jointly.
- REA-KE: a variant of our REA without the detection module.
- REA: our proposed framework.

**5.1.4 Experimental Settings.** The experimental setting details, including hyper-parameter settings and experimental environment, will be discussed in the reproducibility supplement.

## 5.2 Performance Comparison

In this section, we study whether our proposed REA outperforms the state-of-the-art approaches in the noise-involved scenario. Table 2 shows the experimental results. First, we can observe that our proposed REA consistently outperforms all compared methods on all datasets under different evaluation metrics. Specifically, different cross-lingual knowledge graphs encoders still have different performance in the noise-involved scenario. The performance of MTransE and ITransE which apply the basic translation-based encoder is inferior compared with the advance encoders like GCN-Align and MuGNN. However, though the advanced encoder can capture more detailed information to describe the connection between KGs, it is inevitable that the kind of models will be misled by the given noisy labeled entity pairs, which have significantly adverse influence on the performance of entity alignment. Our proposed REA is superior

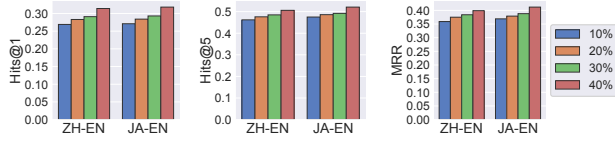


Figure 3: Effect of the proportion of  $\mathcal{AS}^T$ .

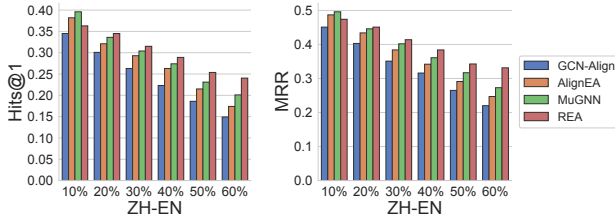


Figure 4: Effect of the intensity of noise (10%-60%).

to MuGNN due to the noise-aware design, although MuGNN would surpass the GCN-based KG encoder in REA in the scenario with clean labeled entity pairs. In addition, we can see that our variant REA-KE without noise detection module under-performs REA. This facts the effectiveness of REA on noise-aware entity alignment. REA-KE and GCN-Align adopt similar KG encoders, so they have the on-par performance as expected.

We further investigate how many noises have been correctly detected. We employ the converged discriminator to do binary classification on the entity pairs in  $\mathcal{AS}^U$ , which contains unknown real and noisy entity pairs. As Table 3 shown, the high precision indicates that most of noise has been correctly detected. The recall shows our discriminator can distinguish 71%- 82% of real entity pairs in  $\mathcal{AS}^U$  as positive on different datasets. Notably, due to the incompleteness of the studied KGs, entities with same semantic meaning in different KGs may be still far away after KG embedding and reported as noise. Overall, these results express that our noise detection module has a stable performance and is quite helpful for noise-involved entity alignment task.

### 5.3 Further Analysis of REA

In this section, we investigate the impact of different parameters on the performance of noise-involved entity alignment.

**5.3.1 Effect of the Proportion of  $\mathcal{AS}^T$ .** We vary the proportion of trusted entity pairs  $\mathcal{AS}^T$  in our framework to evaluate the effect of  $\mathcal{AS}^T$ . Figure 3 shows that the performance is improved as the proportion of  $\mathcal{AS}^T$  in  $\mathcal{AS}$  increases. In particular, we can see that the model with 40% of  $\mathcal{AS}$  as the trusted entity pairs achieves the best performance. The reason is that a large  $\mathcal{AS}^T$  can provide more information about the real entity pairs to assist the noise detection module to distinguish the real pairs from  $\mathcal{AS}^U$ . On the contrary, the noise detection module would be underfitting with a small proportion of  $\mathcal{AS}^T$  (10%) and would mistakenly recognize some real entity pairs as negative.

**5.3.2 Effect of the Intensity of Noise .** We explore the influence of the intensity of noise on the performance of entity alignment.

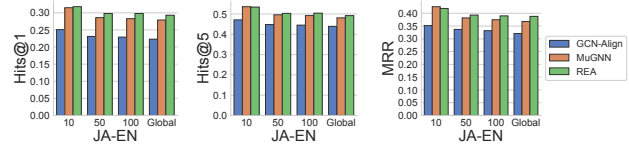


Figure 5: Effect of different types of noise .

We randomly replace 10% - 60% of  $\mathcal{AS}$  with noisy entity pairs. As Figure 4 shown, all methods have increased performance with the decreased proportion of noise. Our REA is preponderant compared with other baselines when the proportion of noise is 20%-60%, showing our method can effectively mitigate the effect of noise. REA is inferior to AlignEA and MuGNN when the proportion is 10%, because the adverse influence is less when the proportion of noise is small. Another cause is that AlignEA and MuGNN adopt the more powerful and complex KGs encoder than the basic GCN-based encoder adopted by REA. Yet, REA still has a better performance than GCN-Align, which uses the similar KG encoder as REA. This observation also expresses the effectiveness of our model with a small proportion of noise entity pairs.

**5.3.3 Effect of Types of Noise.** We investigate the effect of types of noise by controlling the scope of noisy sample generation. Specifically, we sample a negative entity  $e'_1$  for an entity  $e_1$  in a real entity pair  $(e_1, e_2)$  from its  $k$ -nearest entities in KG. In this experiment, we set  $k=10, 50, 100$  and  $N$  (all other entities). Figure 5 reveals that the results of alignment with different types of noise are similar when  $k = 50, 100$  and  $N$ . The reason may be that the negative sample  $e'_1$  is still far from the positive  $e_1$  in embedding space, and it is hard for GCN-Align and MuGNN to capture the correct association between entities. We can find that all methods achieve a better performance when  $k=10$ , especially for MuGNN, because the negative sample  $e'_1$  sampled from the small scope of  $e_1$ ' neighbors has the relatively small distance from  $e_1$ . Although these alignment models have a greater chance to obtain the right one, the noisy pairs still have a non-negligible impact on the performance, even with a small  $k$ . Overall, these results show that REA can effectively handle the situations with different types of noise.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel framework called REA for robust cross-lingual entity alignment problem to mitigate the adverse impact of noise in the given labeled training data. We designed the noise-aware entity alignment module and noise detection module to jointly associate entities in different knowledge graphs. Specifically, we devised the noise-aware entity alignment module by applying the graph neural networks as the knowledge graph encoder. Next, we designed the noise detection to generate the noisy data and distinguish the generated noisy data with real data following the adversarial training principle. Extensive experimental results have verified the effectiveness of REA on the noise-involved entity alignment task. Nevertheless, our proposed REA still have two limitations. First, REA needs a set of trusted entity pairs as the positive samples which require human effort to some extent. Second, REA misrecognized a few labeled real entity pairs as noisy



pairs and a few noisy entity pairs as real pairs, so that the small number of misrecognized entity pairs still have adverse impact on the performance of entity alignment. In future work, we are going to design a more advanced strategy to overcome the above limitations, and expand the exploration and discussion about noise in the attribute of entities and knowledge graphs, meanwhile to develop a comprehensive framework for combating the noise problem in entity alignment task.

## ACKNOWLEDGMENTS

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST), under award number URF/1/3756-01-01 and KAUST AI Initiative, and NSFC No. 61828302.

## REFERENCES

- [1] Leman Akoglu and Christos Faloutsos. 2013. Anomaly, event, and fraud detection in large network datasets. In *WSDM*. 773–774.
- [2] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*. 410–421.
- [3] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *DAMI* 29, 3 (2015), 626–688.
- [4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*. 2787–2795.
- [5] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. 2019. Multi-Channel Graph Neural Network for Entity Alignment. In *ACL*. 1452–1461.
- [6] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-lingual Entity Alignment. In *IJCAL*. 3998–4004.
- [7] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2016. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954* (2016).
- [8] Quanyu Dai, Qiang Li, Jian Tang, and Dan Wang. 2018. Adversarial network embedding. In *AAAI*. 2167–2174.
- [9] Kaize Ding, Jundong Li, and Huan Liu. 2019. Interactive anomaly detection on attributed networks. In *WSDM*. 357–365.
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*. 2672–2680.
- [11] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*. 1024–1034.
- [12] Binbin Hu, Yuan Fang, and Chuan Shi. 2019. Adversarial Learning on Heterogeneous Information Networks. In *KDD*. 120–129.
- [13] Wei Hu, Jianfeng Chen, and Yuzhong Qu. 2011. A self-training approach for resolving object coreference on the semantic web. In *Web Conf*. 87–96.
- [14] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [15] Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. 2013. Sigma: Simple greedy matching for aligning large knowledge bases. In *KDD*. 572–580.
- [16] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*. 5447–5456.
- [17] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.
- [18] Chengjiang Li, Yixin Cao, Lei Hou, Jiabin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised Entity Alignment via Joint Knowledge Embedding Model and Cross-graph Model. In *EMNLP-IJCNLP*. 2723–2732.
- [19] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. 2019. Learning to learn from noisy labeled data. In *CVPR*. 5051–5059.
- [20] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. 2019. SpecAE: Spectral AutoEncoder for Anomaly Detection in Attributed Networks. In *CIKM*. 2233–2236.
- [21] Yan Li and Jieping Ye. 2018. Learning adversarial networks for semi-supervised text classification via policy gradient. In *KDD*. 1715–1723.
- [22] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. 3.
- [23] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*. 1–11.
- [24] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: An Efficient and Robust Entity Alignment Approach for Cross-lingual Knowledge Graph. In *WSDM*. 420–428.
- [25] Thanh Nguyen, Viviane Moreira, Huong Nguyen, Hoa Nguyen, and Juliana Freire. 2011. Multilingual schema matching for Wikipedia infoboxes. *Proc. VLDB Endowment* 5, 2 (2011), 133–144.
- [26] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Web Conf*. 3130–3136.
- [27] Shichao Pei, Lu Yu, and Xiangliang Zhang. 2019. Improving cross-lingual entity alignment via optimal transport. In *IJCAI*. 3231–3237.
- [28] Yingchun Shan, Chenyang Bu, Xiaojian Liu, Shengwei Ji, and Lei Li. 2018. Confidence-aware negative sampling method for noisy knowledge graph embedding. In *ICBK*. 33–40.
- [29] Xiaofei Shi and Yanghua Xiao. 2019. Modeling Multi-mapping Relations for Precise Cross-lingual Entity Alignment. In *EMNLP-IJCNLP*. 813–822.
- [30] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Web Conf*. 697–706.
- [31] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*. 628–644.
- [32] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *IJCAI*. 4396–4402.
- [33] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*. 1057–1063.
- [34] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI*. 297–304.
- [35] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Comm. of the ACM* 57, 10 (2014), 78–85.
- [36] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Graphgan: Graph representation learning with generative adversarial nets. In *AAAI*. 2508–2515.
- [37] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*. 515–524.
- [38] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In *EMNLP*. 349–357.
- [39] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8, 3-4 (1992), 229–256.
- [40] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAL*. 5278–5284.
- [41] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. Jointly Learning Entity and Relation Representations for Entity Alignment. In *EMNLP-IJCNLP*. 240–249.
- [42] Ruobing Xie, Zhiyuan Liu, Fen Lin, and Leyu Lin. [n.d.]. Does william shakespeare really write hamlet? knowledge representation learning with confidence. In *AAAI*. 4954–4961.
- [43] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *ACL*. 3156–3161.
- [44] Yang Yang, Yizhou Sun, Jie Tang, Bo Ma, and Juanzi Li. 2015. Entity matching across heterogeneous sources. In *KDD*. 1395–1404.
- [45] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI*. 4135–4141.
- [46] Minji Yoon, Bryan Hooi, Kijung Shin, and Christos Faloutsos. 2019. Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach. In *KDD*. 647–657.
- [47] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 2018. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *KDD*. 2672–2681.
- [48] Wenchao Yu, Cheng Zheng, Wei Cheng, Charu C Aggarwal, Dongjin Song, Bo Zong, Haifeng Chen, and Wei Wang. 2018. Learning deep network representations with adversarially regularized autoencoders. In *KDD*. 2663–2671.
- [49] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. In *IJCAI*. 5429–5435.
- [50] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*. 4258–4264.
- [51] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *IJCAI*. 10–16.

## A PSEUDOCODE OF REA TRAINING PROCEDURE

The pseudocode of REA training procedure is described in Algorithm 1. After training, the optimized embeddings  $E^i$  and  $E^j$  can be utilized to do the inference.

## B EXPERIMENTAL SETTINGS.

### B.1 Environment

We implement the proposed framework using the Python library Tensorflow and conduct all the experiments on Linux server with GPUs (GeForce RTX 2080 Ti) and CPU (Intel Xeon). We will release the source code of REA at <https://github.com/scpei/REA>. And the public datasets used in the paper are available at: <http://ws.nju.edu.cn/jape/>.

### B.2 Parameter Settings

In this work, we adopt widely-used evaluation metrics, Hits@K and MRR for evaluating entity alignment results as existing works did. We set the dimension of entity embedding as 200 for all methods. For baselines, we find the optimal parameters or follow the settings in original papers to achieve the best performance. For our REA method, we apply a grid search for hyper-parameters and find the best configuration: margin hyper-parameter  $\gamma$  is 3.0, the number of GCN layers is 2, and MLPs used in generator and discriminator have two-layers with 100 and 30 hidden units, respectively. We set  $m_{EN}$  as 1000,  $m_D$  and  $m_G$  as 1000. In addition, we set the number of negative samples  $k$  as 10. Moreover, we set trust scores as binary value, and threshold  $\delta$  as 0.01, in order to achieve a better balance between the performance of entity alignment and the impact of noise. We use Adam optimizer to optimize the loss function in Eq. 15 and Eq. 10 with learning rate 0.01, and apply SGD to optimize the loss function in Eq. 6. Each evaluation is repeated 5 times and averaged results are reported.

---

### Algorithm 1: Model training for REA

---

**Input:** KG  $\mathcal{G}_i$  and  $\mathcal{G}_j$ , the set of labeled entity pairs  $\mathcal{AS}^U$ , the set of trusted entity pairs  $\mathcal{AS}^T$ .

**Output:** Embeddings of entities in  $\mathcal{G}_i$  and  $\mathcal{G}_j$ .

- 1 Initialize embeddings  $E^i, E^j$ , and parameters of  $G, D$  and  $f_{enc}$ ;
- 2 Initialize the trust scores for  $\mathcal{AS}^U$  and  $\mathcal{AS}^T$ ;
- 3 **while** *not converge* **do**
  - 4     // Noise-aware entity alignment training
  - 5     **for**  $m = 0; m < m_{EN}$  **do**
    - 6         Sample a batch of entity pairs from  $\mathcal{AS}^T$ ;
    - 7         Update  $\mathcal{L}_{EA}$  according to Eq. 6
  - 8     **end**
  - 9     // Noise discriminator training
  - 10    **for**  $m = 0; m < m_D$  **do**
    - 11       Sample a batch of entity pairs in  $\mathcal{AS}^T$ , i.e.,  $(e_x, e_y)$ ;
    - 12       Generate  $m_s$  noisy entity pairs  $(e'_x, e'_y) \sim G(\cdot | (e_x, e_y))$  for each  $(e_x, e_y)$ ;
    - 13       Update  $\varphi$  according to Eq. 15 using  $(e_x, e_y)$  and  $(e'_x, e'_y)$ ;
  - 14    **end**
  - 15    // Noisy entity pair generator training
  - 16    **for**  $m = 0; m < m_G$  **do**
    - 17       Sample a batch of entity pairs in  $\mathcal{AS}^T$ , i.e.,  $(e_x, e_y)$ ;
    - 18       Generate  $m_s$  noisy entity pairs  $(e'_x, e'_y) \sim G(\cdot | (e_x, e_y))$  for each  $(e_x, e_y)$ ;
    - 19       Update  $\theta$  according to Eq. 10 using  $(e_x, e_y)$  and  $(e'_x, e'_y)$ ;
  - 20    **end**
- 21    Update the trust scores of entity pairs in  $\mathcal{AS}^U$ ;
- 22    Add entity pairs in  $\mathcal{AS}^U$  with TS = 1 into  $\mathcal{AS}^T$ ;

---