# Adaptive Graph Convolutional Network for Knowledge Graph Entity Alignment

Renbo Zhu[1] Xukun Luo[1] Meng Ma[1,2] Ping Wang[1,2,3] *
[1]School of Software and Microelectronics, Peking University
[2]National Engineering Research Center for Software Engineering, Peking University
[3]Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education
{zhurenbo,luoxukun,mameng,pwang}@pku.edu.cn

## Abstract

Entity alignment (EA) aims to identify equivalent entities from different Knowledge Graphs (KGs), which is a fundamental task for integrating KGs. Throughout its development, Graph Convolutional Network (GCN) has become one of the mainstream methods for EA. The key idea that GCN works in EA is that entities with similar neighbor structures are highly likely to be aligned. However, the noisy neighbors of entities transfer invalid information, drown out equivalent information, lead to inaccurate entity embeddings, and finally reduce the performance of EA. In this paper, we propose a lightweight framework with no training parameters for both supervised and unsupervised EA. Based on the Sinkhorn algorithm, we design a reliability measure for pseudo equivalent entities and propose Adaptive Graph Convolutional Network to deal with neighbor noises in GCN. During the training, the network dynamically updates the adaptive weights of relation triples to weaken the propagation of noises. Extensive experiments on benchmark datasets demonstrate that our framework outperforms the state-of-the-art methods in both supervised and unsupervised settings.

## 1 Introduction

In recent years, Knowledge Graph (KG), an effective structure for organizing and storing data, has received growing attention. It has been widely applied to many knowledge-driven applications such as Search Engine (Yang et al., 2019), Question Answering (Han et al., 2020), and Recommendation (Sha et al., 2021). Due to the limitations of data sources and construction methods, it is difficult for a KG graph to achieve perfect knowledge coverage, which affects the performance of knowledge-driven applications. To solve this problem, knowledge fusion came into being. By capturing the differences and complementarities of multiple KGs, knowledge fusion makes up for the lack of knowledge and ensures knowledge redundancy, improving the completeness of KGs.

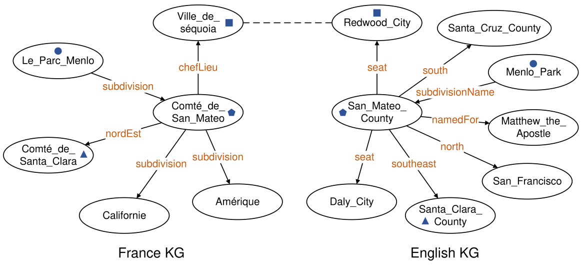


Figure 1: An example of the EA task. In this figure, ellipses indicate entities, directed arrows indicate relations between entities, entities with the same mark are equivalent entities, and dashed lines indicate seed alignments in the supervised setting.

Entity alignment (EA), which aims to find equivalent entities from different KGs, is a fundamental task in knowledge fusion. Figure 1 shows an example of the EA task. There is a partial French KG and a partial English KG. In the supervised EA setting, the equivalent entity pair (*Ville_de_séquoia*, *Redwood_City*) is already known as a seed alignment. The task is to find other equivalent entity pairs from two KGs, e.g., (*Comté_de_San_Mateo*, *San_Mateo_County*). In the unsupervised EA setting, there are no available seed alignments.

Most recently, neural methods have become the dominant approach for EA. They encode entities from two KGs into a unified vector space and then make alignments via measuring the similarity between entity embeddings. Existing neural methods can be divided into two main categories: (1) **TransE-based**. With the assumption that the relation is the translation from the head entity to the tail entity in a relation triple, TransE (Bordes et al., 2013) embeds all relations and entities into a unified vector space for a KG. TransE-based EA methods learn entity embeddings through semantic information. They use seed alignments to construct cross-KG relation triples that connect two graphs

[*]Corresponding author.

*Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6011–6021
December 7-11, 2022 ©2022 Association for Computational Linguistics

and then adopt translation-based KG embedding models on both single-KG and cross-KG relation triples. (2) **GCN-based**. Graph convolutional network (GCN) (Kipf and Welling, 2017) generates node-level embeddings through aggregating information from the neighboring nodes. GCN-based EA methods learn entity embeddings by information diffusion in KGs. They perform the respective graph convolution operations on two KGs and then use seed alignments to align two embedding spaces via margin loss (Wang et al., 2018) or cross-entropy loss (Fey et al., 2020).

TransE and GCN work in EA due to a priori knowledge that there may be multiple pairs of equivalent entities between the respective neighbors of a pair of equivalent entities. Equivalent entities have equivalence transfer to their neighboring entities, respectively. Furthermore, for entities from different KGs, the more pairs of equivalent entities contained between their neighbors, the higher the probability that the two entities are equivalent. However, there still is a critical challenge for EA ignored by most studies. Since equivalent entities may also have non-contributing neighbors, GCN brings additional noises during the message passing process. These noises drown out equivalent information, lead to inaccurate entity embeddings, and finally reduce the performance of EA. We call this phenomenon the neighbor noise problem and these non-contributing connections noisy edges. As an example in figure 1, while entity *Ville_de_séquoia*, *Le_Parc_Menlo*, and *Comté_de_Santa_Clara* contribute to alignment (*Comté_de_San_Mateo*, *San_Mateo_County*), entity *Californie* and *Amérique* bring noises to makes the judgment of equivalence more difficult.

To further confirm the effect of noisy edges on EA performance, we conduct an exploratory experiment with the ground truth alignments. Figure 2 reports the performance of several classic EA methods under different edge settings on dataset DBP15K$_{\text{ZH-EN}}$ (Sun et al., 2017), where MTransE (Chen et al., 2017) belongs to TransE-based methods and GCN-Align (Wang et al., 2018), MRAEA (Mao et al., 2020a), RREA (Mao et al., 2020b), and RAGA (Zhu et al., 2021a) belong to GCN-based methods. After removing all the noisy edges, the performance of both TransE-based and GCN-based methods significantly improves.

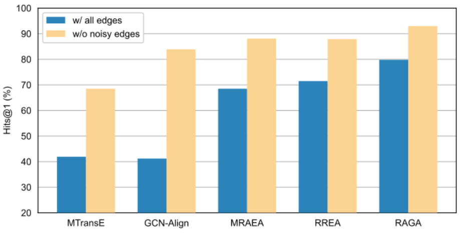


Figure 2: The Hits@1 alignment results under different edge settings.

However, in real scenarios, we cannot gain the ground truth alignments in advance. In this paper, the core idea for dealing with the neighbor noise problem is to reduce the effect of potential noisy edges in KGs. We adopt the Sinkhorn algorithm to design a reliability measure for pseudo equivalent entities. By integrating the reliability measure into GCN, we propose Adaptive Graph Convolutional Network for Entity Alignment, namely AGEA[1]. During the training, it automatically adjusts the weights of relation triples in KGs to weaken the propagation of noisy edges. We summarize the main contributions of this paper as follows:

- We design a measure for pseudo equivalent entities via the Sinkhorn algorithm and propose an adaptive edge weight calculation module to address the neighbor noise problem.
- To leverage the adaptive edge weights, we put forward a lightweight framework with no training parameters in both supervised and unsupervised EA tasks.
- Experimental results on five cross-lingual EA datasets demonstrate that our framework achieves state-of-the-art with high efficiency and interpretability.

## 2 Problem Definition

A KG is formalized as $KG = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ where $\mathcal{E}, \mathcal{R}, \mathcal{T}$ are the sets of entities, relations, and relation triples, respectively. A relation triple $(h, r, t)$ consists of a head entity $h \in \mathcal{E}$, a relation $r \in \mathcal{R}$, and a tail entity $t \in \mathcal{E}$.

Given two KGs, namely $KG_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{T}_1)$ and $KG_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{T}_2)$, we define the task of supervised EA as discovering equivalent entities from different KGs based on a set of seed alignments $\mathcal{A} = \{(e_i, e_j) | e_i \in \mathcal{E}_1, e_j \in \mathcal{E}_2, e_i \leftrightarrow e_j\}$, where $\leftrightarrow$ represents equivalence. And the task of unsupervised EA is defined as the same target without seed alignments $\mathcal{A}$.

[1] Our code is publicly available at https://github.com/zhurboo/AGEA.

# 3 Related Work

## 3.1 Supervised Entity Alignment

TransE-based methods adopt translation-based KG embedding models such as TransE to learn entity embeddings. These methods represent entities by estimating the plausibility of relation triples using a scoring function. MTransE (Chen et al., 2017) encodes entities and relations of each KG in separated embedding space and provides transitions to align the embedding spaces. JAPE (Sun et al., 2017) jointly embeds the structures of two KGs into a unified vector space and further refines it by leveraging attribute correlations in KGs. TransEdge (Sun et al., 2019) contextualizes relation representations in terms of specific head-tail entity pairs. BootEA (Sun et al., 2018) expands seed alignments in a bootstrapping way to learning alignment-oriented KG embeddings. While these methods take advantage of relation semantics, they cannot preserve the global structure information.

GCN-based methods learn entity embeddings by recursively aggregating neighbor features. Since it focuses more on global information than semantic information, it is necessary to design structures incorporating relation semantics into entity embeddings. GCN-Align (Wang et al., 2018) is the first attempt to learn entity embeddings by encoding information from their neighborhoods via GCN. MRAEA (Mao et al., 2020a), RREA (Mao et al., 2020b), and RAGA (Zhu et al., 2021a) incorporate relation semantics into entity representations based on Graph Attention Networks (Velickovic et al., 2018). To explicitly leverage relation information, RDGCN (Wu et al., 2019a) and HGCN (Wu et al., 2019b) approximate relation semantics based on adjacent entity representations. DGMC (Fey et al., 2020) employs synchronous message passing networks to iteratively re-rank the soft correspondences. SoTead (Luo et al., 2022) considers a global optimal entity matching by solving the optimal transport problem.

To our best knowledge, there are three ways relevant to dealing with the neighbor noise problem. (1) **Pair-based**. GMM (Xu et al., 2019) and EPEA (Wang et al., 2020) generate pair-wise entity embeddings, which avoids the neighbor noise problem to some extent. (2) **Weight-based**. NMN (Wu et al., 2020) employs a graph sampling strategy for identifying the most informative neighbors during the training. In such methods, the edge weights and entity embeddings of KGs affect each other, which requires efficient model architectures. (3) **Align-based**. RNM (Zhu et al., 2021b) utilizes neighborhood matching to enhance the entity alignment after the training, and the accuracy of its alignments is influenced by neighbor entities. Our framework AGEA belongs to Weight-based methods. It calculates adaptive edge weights during the training with pseudo equivalent entities. Apart from entity embeddings, no additional network parameters need to be trained.

Furthermore, there is much additional information that can be leveraged to improve the performance of EA, such as entity names (Wu et al., 2019b; Zhu et al., 2021b), entity descriptions (Liu et al., 2021b; Zeng et al., 2020), attribute triples (Sun et al., 2017; Tang et al., 2020), images (Liu et al., 2021a), and text corpora (Chen et al., 2021) . Since entity names are commonly used in the EA task, we only employ entity names to get the initial entity embeddings in this paper.

## 3.2 Unsupervised Entity Alignment

Neural EA methods are superior but are primarily based on supervised learning. These supervised methods depend on seed alignments, which require massive manual and may not be available in practice. To solve the above problem, some unsupervised EA methods were proposed. EVA (Liu et al., 2021a) combines visual knowledge with graph structure information and provides an entirely unsupervised solution by leveraging the visual similarity of entities to create initial seed alignments. SoTead (Luo et al., 2022) adopts entity name textual embeddings to obtain pseudo alignments. Nevertheless, these methods obtain pseudo alignments for training by simple use of additional information of KGs. SEU (Mao et al., 2021) assumes that both entities of two KGs are isomorphic, transform the EA problem into an assignment problem, and make alignments with the Sinkhorn algorithm. SelfKG (Liu et al., 2021b) takes advantage of uni-space learning, relative similarity metric, and self-negative sampling, and performs a self-supervised EA. In our framework AGEA, we define the reliabilities of pseudo alignments via the Sinkhorn algorithm. On the basis that SEU performs training-free alignments, we incorporate the Sinkhorn algorithm into the training process.

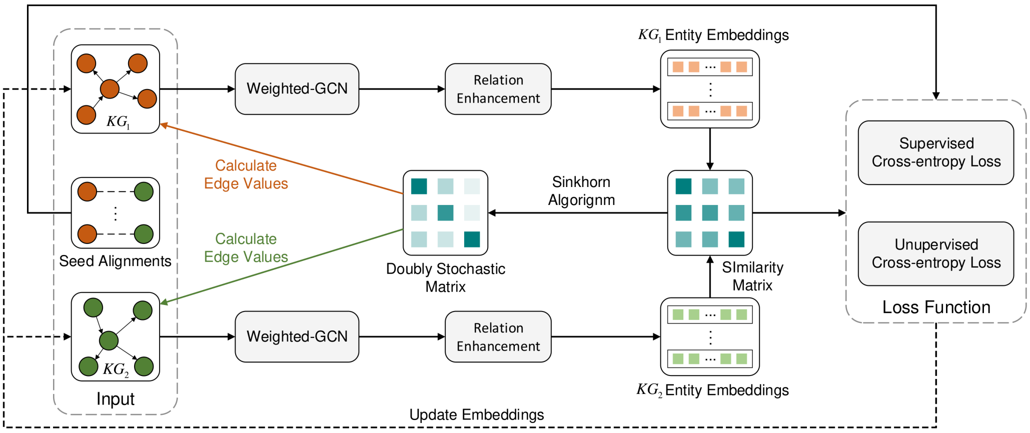


Figure 3: Overall network architecture of AGEA.

## 4 The Proposed Approach

For high efficiency and alleviation of overfitting, our framework AGEA contains a lightweight network with no training parameters except input entity embeddings. Figure 3 depicts the overall network architecture of AGEA. The input consists of relation triples of two KGs, seed alignments between two KGs, and initial entity embeddings. First, for entities in each KG, we adopt a basic single-KG encoder, including a weighted GCN and a relation enhancement module to obtain entity embeddings. Then, a entity similarity matrix is constructed by cosine similarity. For every $N_A$ epoch, we update adaptive weights of relation triples by calculating adaptive edge weights. During the training, we adopt the cross-entropy function to calculate both the supervised and unsupervised loss, and the embeddings of input entities are updated via the backpropagation algorithm.

### 4.1 Basic Single-KG Encoder

**Weighted GCN.** We utilize an $L$-layer weighted GCN to encode entities in each KG with structure information explicitly. The input of the $l$-th GCN layer is $\boldsymbol{X}^{(l)} = \left\{\boldsymbol{x}_1^{(l)}, \boldsymbol{x}_2^{(l)}, \cdots, \boldsymbol{x}_n^{(l)}\right\}$, where $n$ is the number of entities and $\boldsymbol{x}_i^{(l)}$ is the input vector of entity $e_i$ for the $l$-th layer. Unlike node classification tasks, EA pays more attention to the equivalence information brought by neighbor entities. In AGEA, edge weights (weights of relation triples) are already responsible for filtering non-equivalence information. Thus, we use the weight normalization matrix instead of the laplacian normalization matrix to utilize the equivalence information from neighbor entities. Since EA is a task that is bound to overfit, we remove the trainable weight matrix in the vanilla GCN to alleviate overfitting. Furthermore, we put the entities on the hypersphere of their vector space to adapt their distances to cosine similarity. The output of the $l$-th layer is obtained by convolution computation:

$$\boldsymbol{X}^{(l+1)} = \mathrm{Norm}\left(\mathrm{ReLU}\left(\bar{W}^{-1}\bar{A}\boldsymbol{X}^{(l)}\right)\right), \quad (1)$$

where $\bar{A} = A + I$, $A$ is the adjacency matrix of the corresponding KG, $I$ is an identity matrix, $\bar{W}$ is the diagonal adjacent edge weight summation matrix of entities, and $\mathrm{Norm}$ means L2 normalization. It is worth noting that GCN considers bidirectional edges of relation triples. Before training, all edge weights are set to 1. The final output of GCN is the concatenation of the input embeddings and the output of each GCN layer:

$$\boldsymbol{X}^{\mathrm{GCN}} = \left[\boldsymbol{X}^{(0)} \| \boldsymbol{X}^{(1)} \| \ldots \| \boldsymbol{X}^{(L)}\right]. \quad (2)$$

**Relation Enhancement.** The weighted GCN only considers the adjacency of entities, ignoring the relation types between entities. Following (Wu et al., 2019b), we incorporate relation types into entity representations to take full advantage of the structure information KGs. Specifically, we construct relation-aware entity embeddings but eliminate trainable network parameters. For each relation $r$, its representation $\boldsymbol{r}$ is calculated by the average input embeddings of all head entities and tail entities of $r$. Then, the relation-aware entity embeddings $\boldsymbol{X}^{\mathrm{R}}$ are calculated by the average representations of corresponding neighbor relations. Finally, the output of relation enhancement module $\tilde{\boldsymbol{X}}$ is obtained by concatenating of $\boldsymbol{X}^{\mathrm{GCN}}$ and normalized $\boldsymbol{X}^{\mathrm{R}}$:

$$\tilde{\boldsymbol{X}} = \left[\boldsymbol{X}^{\mathrm{GCN}} \| \mathrm{Norm}\left(\boldsymbol{X}^{\mathrm{R}}\right)\right]. \quad (3)$$

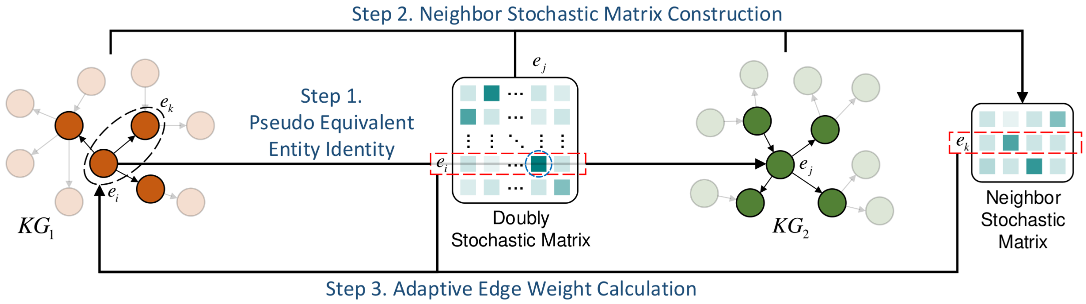


Figure 4: The calculation of adaptive edge weight $w_{ik}$ from $e_i$ to $e_k$ in $KG_1$.

### 4.2 Adaptive Edge Weight Calculation

After the basic single-KG encoder, we obtain entity embeddings $\tilde{\boldsymbol{X}}$. Then a similarity matrix $S$ for entities between two KGs could be constructed by cosine similarity. Before calculating adaptive edge weights, we first introduce a reliability measure for pseudo equivalent entities, which is used by the adaptive edge weights calculation and the unsupervised Loss. Inspired by (Mao et al., 2021), we apply the Sinkhorn algorithm (Cuturi, 2013) to $S$ to generate a doubly stochastic matrix $\bar{S}$:

$$\begin{aligned}\bar{S}^{(0)} &= \exp(TS),\\ \bar{S}^{(m)} &= \mathrm{Norm}_c\left(\mathrm{Norm}_r\left(\bar{S}^{(m-1)}\right)\right),\\ \bar{S} &= \lim_{m\to\infty} S^{(m)},\end{aligned} \tag{4}$$

where $T$ is a temperature coefficient, $\mathrm{Norm}_c$ and $\mathrm{Norm}_r$ mean L1 normalization with columns and rows, respectively. In practice, a certain number of operation rounds can approach $\bar{S}$. The time complexity of the Sinkhorn algorithm is $O(N_S n^2)$, where $N_S$ is the number of iterations. The Sinkhorn algorithm comprehensively considers the alignment problem in both directions, and the sum of the elements in each column and row of $\bar{S}$ is 1. In SEU (Mao et al., 2021), authors apply $\bar{S}$ directly to making alignment. However, we regard $\bar{S}_{ij}$ as the global similarity between entity $e_i$ from $KG_1$ and entity $e_j$ from $KG_2$.

For each entity $e_i$ in $KG_1$, we take the entity corresponding to the maximum value of $\bar{S}_{i:}$ as its pseudo equivalent entity. Besides, we define the reliability of this pair of pseudo equivalent entities as the maximum value minus the second largest value of $\bar{S}_{i:}$, denoted as $c(\bar{S}_{i:})$. Similarly, for each entity $e_j$ in $KG_2$, the reliability of the corresponding pair of pseudo equivalent entities is $c(\bar{S}_{:j})$.

To weaken the propagation of neighbor noises, we leverage the reliabilities of two adjacent entities with their respective pseudo equivalent entities for adaptive edge weights. Figure 4 illustrates an example of the adaptive edge weight calculation. For entity $e_i$ in $KG_1$, we first identity its pseudo equivalent entity $e_j$ in $KG_2$. Then a neighbor stochastic matrix $\bar{S}^{\mathcal{N}_{ij}}$ is constructed by extracting the row and column elements corresponding to neighbors of $e_i$ and $e_j$ from the doubly stochastic matrix $\bar{S}$. For the edge from $e_i$ to $e_k$, we define the $e_i$'s weight $w_i$ as the reliability $c\left(\bar{S}_{i:}\right)$ and $e_k$'s weight $w_k$ as the reliability $c\left(\bar{S}^{\mathcal{N}_{ij}}_{k:}\right)$. Finally, the corresponding adaptive edge weight $w_{ik}$ is calculated as follows:

$$w_{ik} = \max\left(w_i, \lambda\right) \cdot \max\left(w_k, \lambda\right), \tag{5}$$

where $\lambda$ is an hyperparameter that controls the minimum value of entity weights. In general, the adaptive edge weight $w_{ik}$ and $w_{ki}$ between entity $e_i$ and $e_j$ are asymmetric. We take the maximum value of $w_{ik}$ and $w_{ki}$ as the weight of the bidirectional edge. The time complexity of calculating whole adaptive edge weights is $O(n^2 d)$, where $d$ is the average degree of entities.

### 4.3 Training and Alignment

**Supervised Loss.** In the supervised setting, seed alignments $\mathcal{A}$ are provided. We adopt the multi-class cross entropy loss function. For each alignment $(e_i, e_j)$ in $\mathcal{A}$, we regard $e_j$ as the positive class and the entities corresponding to the largest $k$ elements other than $e_j$ in $S_{i:}$ as the set of negative classes $\mathcal{N}_i^k$. Then we use the values from $S$ as input logits and the supervised loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = -\frac{1}{|\mathcal{A}|}\sum_{(e_i,e_j)\in\mathcal{A}} \log\frac{\exp\left(S_{ij}\right)}{\sum_{e_{j'}\in\mathcal{N}_i^k\cup\{e_j\}} \cdot \exp\left(S_{ij'}\right)}. \tag{6}$$

**Unsupervised Loss.** In the unsupervised setting, no seed alignments are available. We use the reliability calculated by each entity in $KG_1$ to generate a set of pseudo alignments $\mathcal{A}'$. Specially, for each entity $e_i$, if $c\left(\bar{S}_{i:}\right) > \mu$ holds, then $e_i$ and its corresponding pseudo equivalent entity form a pseudo alignment. We replace $\mathcal{A}$ in Eq. 6 with $\mathcal{A}'$ to calculate the unsupervised loss.

**Alignment.** After the training, on the hypersphere of entity vector space, entities with similar neighbors are pulled closer, otherwise they are pushed away. Thus, a final similarity matrix for entities between two KGs could be obtained. Like SEU, we then apply the Sinkhorn algorithm to make alighments.

## 5 Experimental Setup

### 5.1 Datasets

We evaluate the proposed framework on two sets of frequently utilized datasets, including five pairs of KGs: (1) DBP15K (Sun et al., 2017). It contains three pairs of cross-lingual KGs: ZH-EN (Chinese to English), JA-EN (Japanese to English), and FR-EN (French to English). Each dataset includes 15,000 alignment entity pairs. (2) SRPRS (Guo et al., 2019). The datasets in it are sparser, which means their degree distributions are closer to the real world. We choose two pairs of cross-lingual KGs (EN-FR and EN-DE) for evaluation. Also, each dataset includes 15,000 alignment entity pairs. Table 1 show the statistics of datasets.

Table 1: Statistics data of datasets.

| Dataset | | #Entities | #Relations | #Rel Triples | #Alignments |
|---|---|---|---|---|---|
| DBP15K$_{\text{ZH-EN}}$ | Chinese | 19,388 | 1,701 | 70,414 | 15,000 |
| | English | 19,572 | 1,323 | 95,142 | |
| DBP15K$_{\text{JA-EN}}$ | Japanese | 19,814 | 1,299 | 77,214 | 15,000 |
| | English | 19,780 | 1,153 | 93,484 | |
| DBP15K$_{\text{FR-EN}}$ | French | 19,661 | 903 | 105,998 | 15,000 |
| | English | 19,993 | 1,208 | 93,484 | |
| SRPRS$_{\text{EN-FR}}$ | English | 15,000 | 221 | 36,508 | 15,000 |
| | French | 15,000 | 177 | 33,532 | |
| SRPRS$_{\text{EN-DE}}$ | English | 15,000 | 222 | 38,363 | 15,000 |
| | German | 15,000 | 120 | 37,377 | |

### 5.2 Evaluation Metrics

Following (Wang et al., 2020), we use H@k (Hits@K) and MRR (Mean Reciprocal Rank) to measure the performance of EA. In the supervised setting, most previous works use 30% of the alignments as training data and 70% for testing, reporting the best results on testing data. For both normality and fair comparison, we take 5% of the testing data for validating and also report the results in all 70% testing data. It is worth noting that our evaluation method will lead to lower results compared with the evaluation method in most previous works. In the unsupervised setting, we randomly take out 5% of the alignments for validating and report the alignment results in all data. All reported results of our approach are the mean of ten experiment results of different random seeds.

### 5.3 Compared Methods

To comprehensively evaluate our framework, we compare both supervised EA methods and unsupervised baselines. For a fair comparison, we try to avoid comparing with the methods that require additional information besides entity names. The compared supervised baselines include TransE-based methods: MTransE (Chen et al., 2017), JAPE (Sun et al., 2017), BootEA (Sun et al., 2018), and TransEdge (Sun et al., 2019); GCN-based metheds: GCN-Align (Wang et al., 2018), MRAEA (Mao et al., 2020a), HGCN (Wu et al., 2019b), DGMC (Fey et al., 2020), NMN (Wu et al., 2020), RAGA (Zhu et al., 2021a), RNM (Zhu et al., 2021b), EPEA (Wang et al., 2020), and SoTead (Luo et al., 2022). And all unsupervised baselines are GCN-based, including EVA (Liu et al., 2021a), SEU (Mao et al., 2021), SelfKG (Liu et al., 2021b), and SoTead (Luo et al., 2022).

In the above methods, BootEA, TransEdge, and MRAEA adopt iteration or bootstrapping strategies to perform semi-supervised EA. JAPE, EPEA, EVA, and SelfKG adopt attribute triples, char-level entity names, entity images, and entity descriptions, respectively. These strategies and additional information are not leveraged in our framework. Besides, we add a naive method NameE which only adopts our initial entity embeddings and cosine similarly.

### 5.4 Implementation Details

Following (Wu et al., 2019b), we translate non-English entity names to English and construct the initial entity embeddings[2]. by pre-trained Glove (Pennington et al., 2014). We utilize Pytorch to implement our framework AGEA. Moreover, we apply multi-processing to speed up the computation of adaptive edge weights and the neighbor-aware alignment. The experiments are conducted on a workstation with an Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz, 128 GB memory, and an NVIDIA GeForce RTX 2080Ti GPU.

[2] We adopt the exact initial entity names as (Mao et al., 2021) and use glove.6B.300d.txt from `https://nlp.stanford.edu/projects/glove`.

During the training, the number of negative classes for each alignment $k$ is 20, and the number of interval epochs to calculate adaptive edge weights $N_A$ is 15. When calculating adaptive edge weights, the temperature of the Sinkhorn algorithm $T$ is 50, the number of iterations for the Sinkhorn algorithm $N_S$ is 10, and the minimum of entity weight $\lambda$ is 0.2. In the unsupervised setting, the threshold of constructing pseudo seed alignments $\mu$ is 0.5.

# 6 Experiment Results

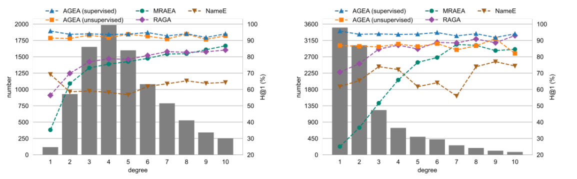
## 6.1 Main Results

Table 2 shows the overall results of all methods. Almost all comparable results are taken from their original papers. Some missing results come from a survey paper (Zhao et al., 2022). Since SEU provides alignment results that only use the same entity name information as us, we take these results from its ablation study.

**Comparisons with Supervised Methods.** Overall, GCN-based methods perform better than TransE-based models because they capture global information and widely use entity names and pre-trained language models to construct initial entity embeddings. In all baselines, RAGA, EPEA, and SoTead achieve the best results on some datasets, respectively. Compared with the above state-of-the-art methods, our framework AGEA achieves significant improvement on all datasets. Compared with NameE, AGEA effectively fuses the initial entity embeddings and the structure information of KGs through training and the neighbor-aware alignment, and finally improves H@1 results by 15.8%-33.6%.

**Comparisons with Unsupervised Methods.** Compared with the strongest baseline SoTead on DBP15K, AGEA improves H@1 results by 1.4%-5.3%. Surprisingly, even compared to most supervised methods, our model still has an advantage, which demonstrates the effectiveness of our reliability measure and unsupervised loss. It is worth noting that SEU is a training-free method. Both SEU and AGEA leverage the Sinkhorn algorithm to achieve satisfactory results, revealing the superiority of the Sinkhorn algorithm.

**Comparisons of Entities with Different Degrees.** As shown in Figure 5, we conducted additional experiments to explore the performance of entities with different degrees. From the perspective of the dataset, compared with DBP15K$_{\text{ZH-EN}}$, a large number of low-degree entities on SRPRS$_{\text{EN-FR}}$ lead to the poor propagation of information and make the task harder. From the perspective of the model, although classic neural methods outperform naive NameE overall, they perform much worse on low-degree entities than high-degree entities. However, our framework AGEA performs even better on low-degree entities than high-degree entities. This is because adaptive edge weights make low-degree entities' backpropagation gradients purer.



(a) DBP15K$_{\text{ZH-EN}}$ (b) SRPRS$_{\text{EN-FR}}$

Figure 5: H@1 results of entities with different degrees.

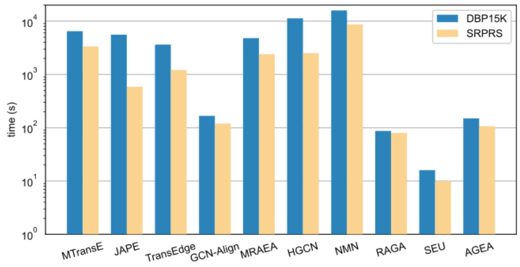


Figure 6: Overall time cost.

## 6.2 Time Efficiency

For the comprehensiveness of our evaluation, we run the source codes provided by the original papers and obtain the average running times on DBP15K and SRPRS, respectively. Figure 6 reported the average times of available methods, which include both training and testing times. It can be seen that the training-free method SEU is the fastest baseline, with a running time of approximately 16s on DBP15K and 10s on SRPRS. Thanks to the lightweight network and multi-process acceleration, our model achieves almost minimal time in training-based models. It takes about 120s on DBP15K and 100s on SRPRS.

## 6.3 Ablation Study

We conduct an extensive ablation study on for supervised AGEA. As shown in Table 3, we implement multiple variants of AGEA. In the table, Ada,

Table 2: Overall results. Methods marked with * adopt iterative or bootstrapping strategies. Methods marked with † use entity names and pre-trained Glove to get the initial entity embeddings. Methods marked with ‡ use additional information that not utilized in our framework AGEA. The best (second) results in supervised or unsupervised settings are marked in bold (underline).

| Method | $\text{DBP15K}_{\text{ZH-EN}}$ | | | $\text{DBP15K}_{\text{JA-EN}}$ | | | $\text{DBP15K}_{\text{FR-EN}}$ | | | $\text{SRPRS}_{\text{EN-FR}}$ | | | $\text{SRPRS}_{\text{EN-DE}}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR |
| | | | | | | | | Supervised | | | | | | | |
| MTransE | 30.8 | 61.4 | 0.364 | 27.9 | 57.5 | 0.349 | 24.4 | 55.6 | 0.335 | 25.1 | 55.1 | 0.350 | 31.2 | 58.6 | 0.400 |
| JAPE‡ | 41.2 | 74.5 | 0.490 | 36.3 | 68.5 | 0.476 | 32.4 | 66.7 | 0.430 | 25.6 | 55.1 | 0.350 | 32.0 | 59.9 | 0.410 |
| BootEA* | 62.9 | 84.7 | 0.703 | 62.2 | 85.4 | 0.701 | 65.3 | 87.4 | 0.731 | 36.5 | 64.9 | 0.460 | 50.3 | 73.2 | 0.580 |
| TransEdge* | 73.5 | 91.9 | 0.801 | 71.9 | 93.2 | 0.795 | 71.0 | 94.1 | 0.796 | 40.0 | 67.5 | 0.490 | 55.6 | 75.3 | 0.630 |
| GCN-Align | 41.2 | 74.5 | 0.549 | 36.3 | 68.5 | 0.476 | 32.4 | 66.7 | 0.430 | 15.5 | 34.5 | 0.220 | 25.3 | 46.4 | 0.330 |
| MRAEA* | 75.7 | 93.0 | 0.827 | 75.8 | 93.4 | 0.826 | 78.0 | 94.8 | 0.849 | 46.0 | 76.8 | 0.559 | 59.4 | 81.5 | 0.664 |
| NameE† | 60.7 | 72.6 | 0.653 | 67.0 | 78.2 | 0.714 | 83.2 | 90.4 | 0.859 | 65.6 | 73.3 | 0.686 | 75.4 | 84.0 | 0.783 |
| HGCN† | 72.0 | 85.7 | 0.760 | 76.6 | 89.7 | 0.810 | 93.3 | 96.0 | 0.910 | 67.0 | 77.0 | 0.710 | 76.3 | 86.3 | 0.801 |
| DGMC† | 80.1 | 87.4 | - | 84.8 | 89.7 | - | 93.3 | 96.0 | - | - | - | - | - | - | - |
| NMN† | 73.3 | 86.9 | - | 78.5 | 91.2 | - | 90.2 | 96.7 | - | - | - | - | - | - | - |
| RAGA† | 79.8 | 93.0 | 0.847 | 83.1 | 95.0 | 0.875 | 91.4 | 98.3 | 0.949 | 78.4 | 89.8 | 0.815 | 87.2 | 94.4 | 0.902 |
| RNM† | 84.0 | 91.9 | 0.870 | 87.2 | 94.4 | 0.899 | 93.8 | 98.1 | 0.954 | - | - | - | - | - | - |
| EPEA†‡ | 88.5 | 95.3 | 0.911 | 92.4 | 96.9 | 0.942 | 95.5 | 98.6 | 0.967 | - | - | - | - | - | - |
| SoTead† | 91.5 | - | - | 94.1 | - | - | 98.4 | - | - | - | - | - | - | - | |
| AGEA† | **94.3** | **98.6** | **0.959** | **96.2** | **99.4** | **0.975** | **99.0** | **99.9** | **0.993** | **87.6** | **95.1** | **0.902** | **95.2** | **98.6** | **0.964** |
| | | | | | | | | Unsupervised | | | | | | | |
| EVA‡ | 73.1 | 90.9 | 0.792 | 73.7 | 89.0 | 0.791 | 75.2 | 89.5 | 0.804 | - | - | - | - | - | - |
| NameE† | 59.2 | 71.3 | 0.635 | 65.4 | 77.2 | 0.697 | 82.2 | 90.0 | 0.850 | 64.5 | 72.7 | 0.675 | 74.3 | 83.2 | 0.775 |
| SEU† | 81.6 | 92.3 | 0.854 | 86.5 | 95.2 | 0.896 | 95.3 | 98.9 | 0.967 | 81.2 | 90.2 | 0.843 | 90.2 | 95.1 | 0.920 |
| SelfKG†‡ | 82.9 | 91.9 | - | 89.0 | 95.3 | - | 95.7 | 99.2 | - | - | - | - | - | - | - |
| SoTead† | 87.7 | - | - | 91.5 | - | - | 97.5 | - | - | - | - | - | - | - | - |
| AGEA† | **93.0** | **97.2** | **0.946** | **94.8** | **97.8** | **0.959** | **98.9** | **99.6** | **0.992** | **87.0** | **93.8** | **0.887** | **94.2** | **98.5** | **0.957** |

Sink, and Train indicate whether to apply the adaptive edge weight calculation, the Sinkhorn algorithm for making alignments, and the training process, respectively. And NameE only adopts intial entity embedding and cosine similarity. Overall, all modules contribute, where the Sinkhorn algorithm and the training process are the basis and contribute the most. Without training, compared with NameE, our framework improves H@1 results by 12.6%-23.4%. Combining only the trained weighted-GCN and the Sinkhorn algorithm, a H@1 result of 93.6% can be achieved on $\text{DBP15K}_{\text{ZH-EN}}$, which is even better than the performance of the best baseline SoTead in Table 2. Moreover, regardless of whether the Sinkhorn algorithm is used or not, after adding the adaptive edge weight calculation, there is a certain accuracy improvement. It indicates the effectiveness of the adaptive edge weight calculation module.

Table 3: H@1 results of different variants.

| Method | ZH-EN | JA-EN | FR-EN | EN-FR | EN-DE |
|---|---|---|---|---|---|
| AGEA | **94.3** | **96.2** | **99.0** | **87.6** | **95.2** |
| w/o Ada | 93.6 | 95.6 | 98.6 | 86.5 | 94.7 |
| w/o Sink | 88.2 | 90.1 | 96.3 | 81.7 | 91.2 |
| w/o Ada+Sink | 84.1 | 86.6 | 94.0 | 79.4 | 90.1 |
| w/o Train | 84.4 | 88.5 | 95.8 | 83.6 | 91.6 |
| w/o Train+Sink | 72.5 | 78.2 | 89.0 | 70.1 | 81.1 |
| NameE | 60.7 | 67.0 | 83.2 | 65.6 | 75.4 |

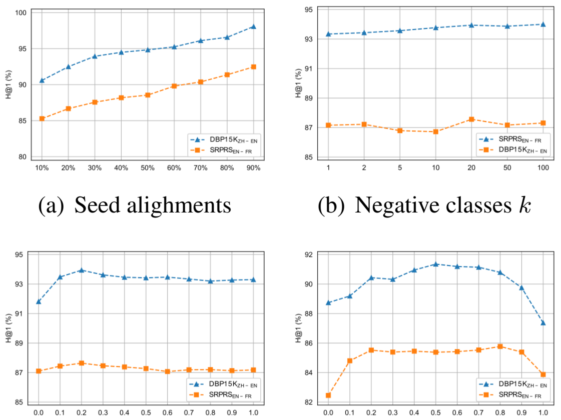
### 6.4 Hyperparameter Analyses

Figure 7 shows the hyperparameter analyses.

**Proportion of Seed Alighments.** More seed alignments provide more information to bridge different KGs. The accuracy of supervised AGEA steadily improves as the proportion of seed alignments increases.

**Number of Negative Classes $k$.** The number of negative classes $k$ balances equivalent and difference information of entity pairs. A small $k$ leads to insufficient difference information, while a large $k$ drowns out equivalent information. As $k$ increases, the performance of supervised AGEA rises slowly on DBP15K and fluctuates slightly on SRPRS.

**Minimum Value of Entity Weights $\lambda$.** When calculating adaptive edge weights, the minimum value of entity weights $\lambda$ implies a correction to the reliability for pseudo equivalent entities, which affects the weights of entities and edges in turn. When $\lambda$ equals 1, all edge weights are 1. Conversely, its value of 0 means that a large number of edge weights are approximately equal to 0, causing insufficient utilization of structure information. To make a trade-off, AGEA performs best in the case of $\lambda$ is about 0.2.



(a) Seed alighments (b) Negative classes $k$

(c) Minimum value $\lambda$ (d) Threshold $\mu$

Figure 7: Hyperparameter analyses.

**Threshold $\mu$.** The threshold $\mu$ controls the accuracy of the pseudo alignments in unsupervised EA. When $\mu$ is small, too few pseudo alignments are constructed, resulting in insufficient training. On the contrary, if $\mu$ is big, the accuracy of the pseudo alignments will drop, leading to inaccurate entity embeddings. In our practice, it is appropriate for $\mu$ to take 0.5.

### 6.5 Case Study

We present a case that visualizes the adaptive edge weights after training in Figure 8. The legends are the same as in Figure 1. We marked the weight values on the edge and used transparency of edges to show the strength of the weights. It can be seen that for the equivalent entity pair (*Comté_de_San_Mateo*, *San_Mateo_County*), their corresponding edge weights of the neighborhood equivalent entity pairs (*Ville_de_séquoia*, *Redwood_City*), (*Le_Parc_Menlo*, *Menlo_Park*), and (*Comté_de_Santa_Clara*, *Santa_Clara_County*) are large. Furthermore, since (*Ville_de_ séquoia*, *Redwood_City*) is a seed alignment, their corresponding edge weights are slightly larger than the other two equivalent entity pairs. The above phenomena reveal the feasibility of our adaptive edge weight calculation module in addressing the neighbor noise problem.

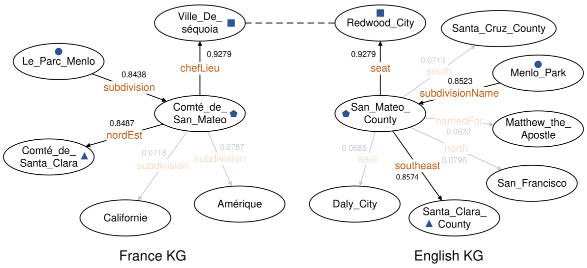


Figure 8: Visualization of adaptive edge weights.

## 7 Conclusion

In this work, we deal with the neighbor noise problem in the EA task. A lightweight and efficient framework AGEA has been proposed, which mainly consists of an adaptive edge weight calculation module. Besides, the Sinkhorn algorithm is incorporated into our GCN-based EA framework and shows satisfactory performance. Experiments on five datasets indicate that AGEA outperforms the state-of-the-art methods in both supervised and unsupervised settings with high efficiency and interpretability.

## Limitations

In our framework AGEA, the adaptive edge weight calculation module, unsupervised loss, and final alignment are all based on the Sinkhorn algorithm. However, the premise that the Sinkhorn algorithm works in the EA task is that EA can be transformed into an assignment problem (Mao et al., 2021), which requires that for each entity there always exists a counterpart in the other KG. Our framework performs well on ideal datasets DBP15K and SRPRS, but may defeat to datasets where the two KGs have little overlap. This limitation is ignored by most previous works and leads to the new research area called entity alignment with dangling cases (Luo et al., 2022).

## Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments.

## References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Muhao Chen, Weijia Shi, Ben Zhou, and Dan Roth. 2021. Cross-lingual entity alignment with incidental supervision. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021,* *Online, April 19 - 23, 2021*, pages 645–658. Association for Computational Linguistics.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1511–1517. ijcai.org.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2292–2300.

Matthias Fey, Jan Eric Lenssen, Christopher Morris, Jonathan Masci, and Nils M. Kriege. 2020. Deep graph matching consensus. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2505–2514. PMLR.

Jiale Han, Bo Cheng, and Xu Wang. 2020. Open domain question answering based on text enhanced knowledge graph with hyperedge infusion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1475–1481, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021a. Visual pivoting for (unsupervised) entity alignment. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4257–4266. AAAI Press.

Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. 2021b. A self-supervised method for entity alignment. *CoRR*, abs/2106.09395.

Shengxuan Luo, Pengyu Cheng, and Sheng Yu. 2022. Semi-constraint optimal transport for entity alignment with dangling cases. *CoRR*, abs/2203.05744.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. From alignment to assignment: Frustratingly simple unsupervised entity alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2853, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020a. MRAEA: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 420–428. ACM.

Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020b. Relational reflection entity alignment. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 1095–1104. ACM.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Xiao Sha, Zhu Sun, and Jie Zhang. 2021. Hierarchical attentive knowledge graph embedding for personalized recommendation. *Electron. Commer. Res. Appl.*, 48:101071.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, volume 10587 of *Lecture Notes in Computer Science*, pages 628–644. Springer.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4396–4402. ijcai.org.

Zequn Sun, JiaCheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, volume 11778 of *Lecture Notes in Computer Science*, pages 612–629. Springer.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. BERT-INT: A bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3174–3180. ijcai.org.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 349–357, Brussels, Belgium. Association for Computational Linguistics.

Zhichun Wang, Jinjian Yang, and Xiaoju Ye. 2020. Knowledge graph alignment with entity-pair embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1672–1680, Online. Association for Computational Linguistics.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019a. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5278–5284. ijcai.org.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019b. Jointly learning entity and relation representations for entity alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 240–249, Hong Kong, China. Association for Computational Linguistics.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2020. Neighborhood matching network for entity alignment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6477–6487, Online. Association for Computational Linguistics.

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 *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3156–3161. Association for Computational Linguistics.

Yueji Yang, Divyakant Agrawal, H. V. Jagadish, Anthony K. H. Tung, and Shuang Wu. 2019. An efficient parallel keyword search engine on knowledge graphs. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 338–349. IEEE.

Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2020. Collective entity alignment via adaptive features. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, pages 1870–1873. IEEE.

Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian M. Suchanek. 2022. An experimental study of state-of-the-art entity alignment approaches. *IEEE Trans. Knowl. Data Eng.*, 34(6):2610–2625.

Renbo Zhu, Meng Ma, and Ping Wang. 2021a. RAGA: relation-aware graph attention networks for global entity alignment. In *Advances in Knowledge Discovery and Data Mining - 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11-14, 2021, Proceedings, Part I*, volume 12712 of *Lecture Notes in Computer Science*, pages 501–513. Springer.

Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2021b. Relation-aware neighborhood matching model for entity alignment. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4749–4756. AAAI Press.