Neighborhood Matching Network for Entity Alignment

Yuting Wu[1], **Xiao Liu**[1], **Yansong Feng**[1,2]*, **Zheng Wang**[3] and **Dongyan Zhao**[1,2]

[1]Wangxuan Institute of Computer Technology, Peking University, China
[2]The MOE Key Laboratory of Computational Linguistics, Peking University, China
[3]School of Computing, University of Leeds, U.K.
{wyting,lxlisa,fengyansong,zhaodongyan}@pku.edu.cn
z.wang5@leeds.ac.uk

Abstract

Structural heterogeneity between knowledge graphs is an outstanding challenge for entity alignment. This paper presents Neighborhood Matching Network (NMN), a novel entity alignment framework for tackling the structural heterogeneity challenge. NMN estimates the similarities between entities to capture both the topological structure and the neighborhood difference. It provides two innovative components for better learning representations for entity alignment. It first uses a novel graph sampling method to distill a discriminative neighborhood for each entity. It then adopts a cross-graph neighborhood matching module to jointly encode the neighborhood difference for a given entity pair. Such strategies allow NMN to effectively construct matching-oriented entity representations while ignoring noisy neighbors that have a negative impact on the alignment task. Extensive experiments performed on three entity alignment datasets show that NMN can well estimate the neighborhood similarity in more tough cases and significantly outperforms 12 previous state-of-the-art methods.

1 Introduction

By aligning entities from different knowledge graphs (KGs) to the same real-world identity, *entity alignment* is a powerful technique for knowledge integration. Unfortunately, entity alignment is non-trivial because real-life KGs are often incomplete and different KGs typically have heterogeneous schemas. Consequently, equivalent entities from two KGs could have distinct surface forms or dissimilar neighborhood structures.

In recent years, embedding-based methods have become the dominated approach for entity alignment (Zhu et al., 2017; Pei et al., 2019a; Cao et al., 2019; Xu et al., 2019; Li et al., 2019a; Sun et al.,
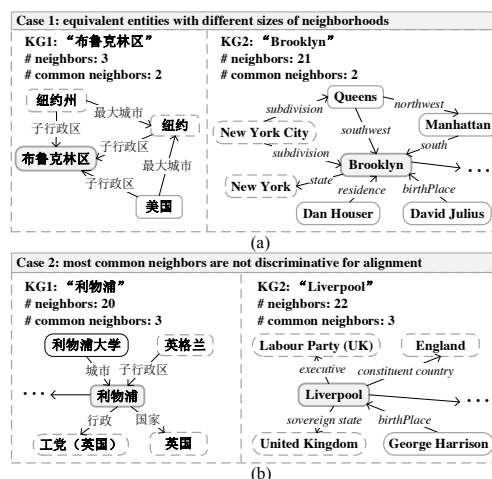

Figure 1: Illustrative examples: two tough cases for entity alignment. Dashed rectangles denote the common neighbors between different KGs.

2020). Such approaches have the advantage of not relying on manually constructed features or rules (Mahdisoltani et al., 2015). Using a set of seed alignments, an embedding-based method models the KG structures to automatically learn how to map the equivalent entities among different KGs into a unified vector space where entity alignment can be performed by measuring the distance between the embeddings of two entities.

The vast majority of prior works in this direction build upon an important assumption - entities and their counterparts from other KGs have similar neighborhood structures, and therefore, similar embeddings will be generated for equivalent entities. Unfortunately, the assumption does not always hold for real-life scenarios due to the incompleteness and heterogeneities of KGs. As an example, consider Figure 1 (a), which shows two equivalent entities from the Chinese and English versions of Wikipedia. Here, both central entities refer to the same real-world identity, *Brooklyn*, a borough of New York City. However, the two entities have different sizes of neighborhoods and

---

* Corresponding author.

distinct topological structures. The problem of dissimilar neighborhoods between equivalent entities is ubiquitous. Sun et al. (2020) reports that the majority of equivalent entity pairs have different neighbors in the benchmark datasets DBP15K, and the proportions of such entity pairs are over 86% (up to 90%) in different language versions of DBP15K. Particularly, we find that the alignment accuracy of existing embedding-based methods decreases significantly as the gap of equivalent entities' neighborhood sizes increases. For instance, RDGCN (Wu et al., 2019a), a state-of-the-art, delivers an accuracy of 59% on the Hits@1 score on entity pairs whose number of neighbors differs by no more than 10 on DBP15K$_{ZH-EN}$. However, its performance drops to 42% when the difference for the number of neighbors increases to 20 and to 35% when the difference increases to be above 30. The disparity of the neighborhood size and topological structures pose a significant challenge for entity alignment methods.

Even if we were able to set aside the difference in the neighborhood size, we still have another issue. Since most of the common neighbors would be popular entities, they will be neighbors of many other entities. As a result, it is still challenging to align such entities. To elaborate on this point, let us now consider Figure 1 (b). Here, the two central entities (both indicate the city *Liverpool*) have similar sizes of neighborhoods and three common neighbors. However, the three common neighbors (indicate *United Kingdom*, *England* and *Labour Party (UK)*, respectively) are not discriminative enough. This is because there are many *city* entities for England which also have the three entities in their neighborhoods – e.g., the entity *Birmingham*. For such entity pairs, in addition to common neighbors, other informative neighbors – like those closely contextually related to the central entities – must be considered. Because existing embedding-based methods are unable to choose the right neighbors, we need a better approach.

We present Neighborhood Matching Network (NMN), a novel sampling-based entity alignment framework. NMN aims to capture the most informative neighbors and accurately estimate the similarities of neighborhoods between entities in different KGs. NMN achieves these by leveraging the recent development in Graph Neural Networks (GNNs). It first utilizes the Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017)

to model the topological connection information, and then selectively samples each entity's neighborhood, aiming at retaining the most informative neighbors towards entity alignment. One of the key challenges here is how to accurately estimate the similarity of any two entities' sampled neighborhood. NMN addresses this challenge by designing a discriminative neighbor matching module to jointly compute the neighbor differences between the sampled subgraph pairs through a cross-graph attention mechanism. Note that we mainly focus on the neighbor relevance in the neighborhood sampling and matching modules, while the neighbor connections are modeled by GCNs. We show that, by integrating the neighbor connection information and the neighbor relevance information, NMN can effectively align entities from real-world KGs with neighborhood heterogeneity.

We evaluate NMN by applying it to benchmark datasets DBP15K (Sun et al., 2017) and DWY100K (Sun et al., 2018), and a sparse variant of DBP15K. Experimental results show that NMN achieves the best and more robust performance over state-of-the-arts. This paper makes the following technical contributions. It is the first to:

- employ a new graph sampling strategy for identifying the most informative neighbors towards entity alignment (Sec. 3.3).

- exploit a cross-graph attention-based matching mechanism to jointly compare discriminative subgraphs of two entities for robust entity alignment (Sec. 3.4).

2 Related Work

Embedding-based entity alignment. In recent years, embedding-based methods have emerged as viable means for entity alignment. Early works in the area utilize TransE (Bordes et al., 2013) to embed KG structures, including MTransE (Chen et al., 2017), JAPE (Sun et al., 2017), IPTransE (Zhu et al., 2017), BootEA (Sun et al., 2018), NAEA (Zhu et al., 2019) and OTEA (Pei et al., 2019b). Some more recent studies use GNNs to model the structures of KGs, including GCN-Align (Wang et al., 2018), GMNN (Xu et al., 2019), RDGCN (Wu et al., 2019a), AVR-GCN (Ye et al., 2019), and HGCN-JE (Wu et al., 2019b). Besides the structural information, some recent methods like KDCoE (Chen et al., 2018), AttrE (Trisedya et al., 2019), MultiKE (Zhang et al., 2019) and HMAN

(Yang et al., 2019) also utilize additional information like Wikipedia entity descriptions and attributes to improve entity representations.

However, all the aforementioned methods ignore the neighborhood heterogeneity of KGs. MuGNN (Cao et al., 2019) and AliNet (Sun et al., 2020) are two most recent efforts for addressing this issue. While promising, both models still have drawbacks. MuGNN requires both pre-aligned entities and relations as training data, which can have expensive overhead for training data labeling. AliNet considers all one-hop neighbors of an entity to be equally important when aggregating information. However, not all one-hop neighbors contribute positively to characterizing the target entity. Thus, considering all of them without careful selection can introduce noise and degrade the performance. NMN avoids these pitfalls. With only a small set of pre-aligned entities as training data, NMN chooses the most informative neighbors for entity alignment.

Graph neural networks. GNNs have recently been employed for various NLP tasks like semantic role labeling (Marcheggiani and Titov, 2017) and machine translation (Bastings et al., 2017). GNNs learn node representations by recursively aggregating the representations of neighboring nodes. There are a range of GNN variants, including the Graph Convolutional Network (GCN) (Kipf and Welling, 2017), the Relational Graph Convolutional Network (Schlichtkrull et al., 2018), the Graph Attention Network (Veličković et al., 2018). Giving the powerful capability for modeling graph structures, we also leverage GNNs to encode the structural information of KGs (Sec. 3.2).

Graph matching. The similarity of two graphs can be measured by exact matching (graph isomorphism) (Yan et al., 2004) or through structural information like the graph editing distance (Raymond et al., 2002). Most recently, the Graph Matching Network (GMN) (Li et al., 2019b) computes a similarity score between two graphs by jointly reasoning on the graph pair through cross-graph attention-based matching. Inspired by GMN, we design a cross-graph neighborhood matching module (Sec. 3.4) to capture the neighbor differences between two entities' neighborhoods.

Graph sampling. This technique samples a subset of vertices or edges from the original graph. Some of the popular sampling approaches include vertex-, edge- and traversal-based sampling (Hu

and Lau, 2013). In our entity alignment framework, we propose a vertex sampling method to select informative neighbors and to construct a neighborhood subgraph for each entity.

3 Our Approach

Formally, we represent a KG as $G = (E, R, T)$, where E, R, T denote the sets of entities, relations and triples respectively. Without loss of generality, we consider the task of entity alignment between two KGs, G_1 and G_2, based on a set of pre-aligned equivalent entities. The goal is to find pairs of equivalent entities between G_1 and G_2.

3.1 Overview of NMN

As highlighted in Sec. 1, the neighborhood heterogeneity and noisy common neighbors of real-world KGs make it difficult to capture useful information for entity alignment. To tackle these challenges, NMN first leverages GCNs to model the neighborhood topology information. Next, it employs neighborhood sampling to select the more informative neighbors. Then, it utilizes a cross-graph matching module to capture neighbor differences.

As depicted in Figure 2, NMN takes as input two KGs, G_1 and G_2, and produces embeddings for each candidate pair of entities, e_1 and e_2, so that entity alignment can be performed by measuring the distance, $d(e_1, e_2)$, of the learned embeddings. It follows a four-stage processing pipeline: (1) KG structure embedding, (2) neighborhood sampling, (3) neighborhood matching, and (4) neighborhood aggregation for generating embeddings.

3.2 KG Structure Embedding

To learn the KG structure embeddings, NMN utilizes multi-layered GCNs to aggregate higher degree neighboring structural information for entities.

NMNs uses pre-trained word embeddings to initialize the GCN. This strategy is shown to be effective in encoding the semantic information of entity names in prior work (Xu et al., 2019; Wu et al., 2019a). Formally, let $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$ be two KGs to be aligned, we put G_1 and G_2 together as one big input graph to NMN. Each GCN layer takes a set of node features as input and updates the node representations as:

$$\boldsymbol{h}_i^{(l)} = \text{ReLU}(\sum_{j \in N_i \cup \{i\}} \frac{1}{\epsilon_i} \mathbf{W}^{(l)} \boldsymbol{h}_j^{(l-1)}) \quad (1)$$
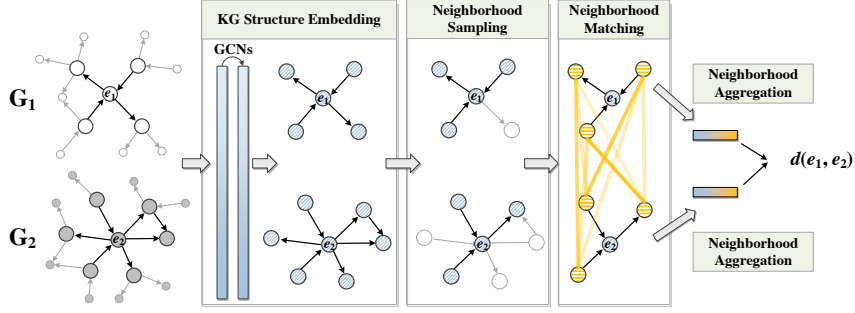
Figure 2: Overall architecture and processing pipeline of Neighborhood Matching Network (NMN).

where $\{\boldsymbol{h}_1^{(l)}, \boldsymbol{h}_2^{(l)}, ..., \boldsymbol{h}_n^{(l)} | \boldsymbol{h}_i^{(l)} \in \mathbb{R}^{d^{(l)}}\}$ is the output node (entity) features of l-th GCN layer, ϵ_i is the normalization constant, N_i is the set of neighbor indices of entity i, and $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ is a layer-specific trainable weight matrix.

To control the accumulated noise, we also introduce highway networks (Srivastava et al., 2015) to GCN layers, which can effectively control the noise propagation across GCN layers (Rahimi et al., 2018; Wu et al., 2019b).

3.3 Neighborhood Sampling

The one-hop neighbors of an entity are key to determine whether the entity should be aligned with other entities. However, as we have discussed in Sec. 1, not all one-hop neighbors contribute positively for entity alignment. To choose the right neighbors, we apply a down-sampling process to select the most informative entities towards the central target entity from its one-hop neighbors.

Recall that we use pre-trained word embeddings of entity names to initialize the input node features of GCNs. As a result, the entity embeddings learned by GCNs contain rich contextual information for both the neighboring structures and the entity semantics. NMN exploits such information to sample informative neighbors, i.e., neighbors that are more contextually related to the central entity are more likely to be sampled. Our key insight is that the more often a neighbor and the central (or target) entity appear in the same context, the more representative and informative the neighbor is towards the central entity. Since the contexts of two equivalent entities in real-world corpora are usually similar, the stronger a neighbor is *contextually related* to the target entity, the more alignment clues the neighbor is likely to offer. Experimental results in Sec. 5.3 confirm this observation.

Formally, given an entity e_i, the probability to sample its one-hop neighbor e_{i_j} is determined by:

$$p(\boldsymbol{h}_{i\_j}|\boldsymbol{h}_i) = \text{softmax}(\boldsymbol{h}_i \mathbf{W}_s \boldsymbol{h}_{i\_j}^T)$$
$$= \frac{exp(\boldsymbol{h}_i \mathbf{W}_s \boldsymbol{h}_{i\_j}^T)}{\sum_{k \in N_i} exp(\boldsymbol{h}_i \mathbf{W}_s \boldsymbol{h}_{i\_k}^T)} \quad (2)$$

where N_i is the one-hop neighbor index of central entity e_i, $\boldsymbol{h}_i$ and $\boldsymbol{h}_{i\_j}$ are learned embeddings for entities e_i and e_{i_j} respectively, and $\mathbf{W}_s$ is a shared weight matrix.

By selectively sampling one-hop neighbors, NMN essentially constructs a discriminative subgraph of neighborhood for each entity, which can enable more accurate alignment through neighborhood matching.

3.4 Neighborhood Matching

The neighborhood subgraph, produced by the sampling process, determines which neighbors of the target entity should be considered in the later stages. In other words, later stages of the NMN processing pipeline will only operate on neighbors within the subgraph. In the neighborhood matching stage, we wish to find out, for each candidate entity in the counterpart KG, which neighbors of that entity are closely related to a neighboring node within the subgraph of the target entity. Such information is essential for deciding whether two entities (from two KGs) should be aligned.

As discussed in Sec. 3.3, equivalent entities tend to have similar contexts in real-world corpora; therefore, their neighborhoods sampled by NMN should be more likely to be similar. NMN exploits this observation to estimate the similarities of the sampled neighborhoods.

Candidate selection. Intuitively, for an entity e_i in E_1, we need to compare its sampled neighborhood subgraph with the subgraph of each candidate entity in E_2 to select an optimal alignment entity. Exhaustively trying all possible entities of E_2 would be prohibitively expensive for large

real-world KGs. To reduce the matching overhead, NMN takes a low-cost approximate approach. To that end, NMN first samples an alignment candidate set $\mathcal{C}_i = \{c_{i_1}, c_{i_2}, ..., c_{i_t} | c_{i_k} \in E_2\}$ for e_i in E_1, and then calculates the subgraph similarities between e_i and these candidates. This is based on an observation that the entities in E_2 which are closer to e_i in the embedding space are more likely to be aligned with e_i. Thus, for an entity e_j in E_2, the probability that it is sampled as a candidate for e_i can be calculated as:

$$p(\boldsymbol{h}_j|\boldsymbol{h}_i) = \frac{exp(\|\boldsymbol{h}_i - \boldsymbol{h}_j\|_{L_1})}{\sum_{k \in E_2} exp(\|\boldsymbol{h}_i - \boldsymbol{h}_k\|_{L_1})} \quad (3)$$

Cross-graph neighborhood matching. Inspired by recent works in graph matching (Li et al., 2019b), our neighbor matching module takes a pair of subgraphs as input, and computes a cross-graph matching vector for each neighbor, which measures how well this neighbor can be matched to any neighbor node in the counterpart. Formally, let (e_i, c_{i_k}) be an entity pair to be measured, where $e_i \in E_1$ and $c_{i_k} \in E_2$ is one of the candidates of e_i, p and q are two neighbors of e_i and c_{i_k}, respectively. The cross-graph matching vector for neighbor p can be computed as:

$$a_{pq} = \frac{\exp(\boldsymbol{h}_p \cdot \boldsymbol{h}_q)}{\sum_{q' \in N_{i_k}^s} \exp(\boldsymbol{h}_p \cdot \boldsymbol{h}_{q'})} \quad (4)$$

$$\boldsymbol{m}_p = \sum_{q \in N_{i_k}^s} a_{pq}(\boldsymbol{h}_p - \boldsymbol{h}_q) \quad (5)$$

where a_{pq} are the attention weights, $\boldsymbol{m}_p$ is the matching vector for p, and it measures the difference between $\boldsymbol{h}_p$ and its closest neighbor in the other subgraph, $N_{i_k}^s$ is the sampled neighbor set of c_{i_k}, $\boldsymbol{h}_p$ and $\boldsymbol{h}_q$ are the GCN-output embeddings for p and q respectively.

Then, we concatenate neighbor p's GCN-output embeddings with weighted matching vector $\boldsymbol{m}_p$:

$$\hat{\boldsymbol{h}}_p = [\boldsymbol{h}_p \| \beta * \boldsymbol{m}_p] \quad (6)$$

For each target neighbor in a neighborhood subgraph, the attention mechanism in the matching module can accurately detect which of the neighbors in the subgraph of another KG is most likely to match the target neighbor. Intuitively, the matching vector $\boldsymbol{m}_p$ captures the difference between the two closest neighbors. When the representations of the two neighbors are similar, the matching vector

tends to be a zero vector so that their representations stay similar. When the neighbor representations differ, the matching vector will be amplified through propagation. We find this matching strategy works well for our problem settings.

3.5 Neighborhood Aggregation

In the neighborhood aggregation stage, we combine the neighborhood connection information (learned at the KG structure embedding stage) as well as the output of the matching stage (Sec. 3.4) to generate the final embeddings used for alignment.

Specifically, for entity e_i, we first aggregate its sampled neighbor representations $\{\hat{\boldsymbol{h}}_p\}$. Inspired by the aggregation method in (Li et al., 2016), we compute a neighborhood representation for e_i as:

$$\boldsymbol{g}_i = (\sum_{p \in N_i^s} \sigma(\hat{\boldsymbol{h}}_p \mathbf{W}_{gate}) \cdot \hat{\boldsymbol{h}}_p)\mathbf{W}_N \quad (7)$$

Then, we concatenate the central entity e_i's GCN-output representation $\boldsymbol{h}_i$ with its neighborhood representation to construct the matching oriented representation for e_i:

$$\boldsymbol{h}_i^{match} = [\boldsymbol{g}_i \| \boldsymbol{h}_i] \quad (8)$$

3.6 Entity Alignment and Training

Pre-training. As discussed in Sec. 3.3, our neighborhood sampling is based on the GCN-output entity embeddings. Therefore, we first pre-train the GCN-based KG embedding model to produce quality entity representations. Specifically, we measure the distance between two entities to determine whether they should be aligned:

$$\tilde{d}(e_1, e_2) = \|\boldsymbol{h}_{e_1} - \boldsymbol{h}_{e_2}\|_{L_1} \quad (9)$$

The objective of the pre-trained model is:

$$\tilde{L} = \sum_{(i,j) \in \mathbb{L}} \sum_{(i',j') \in \mathbb{L}'} \max\{0, \tilde{d}(i, j) - \tilde{d}(i', j') + \gamma\} \quad (10)$$

where $\gamma > 0$ is a margin hyper-parameter; $\mathbb{L}$ is our alignment seeds and $\mathbb{L}'$ is the set of negative aligned entity pairs generated by nearest neighbor sampling (Kotnis and Nastase, 2017).

Overall training objective. The pre-training phase terminates once the entity alignment performance has converged to be stable. We find that after this stage, the entity representations given by the GCN are sufficient for supporting the neighborhood sampling and matching modules. Hence,
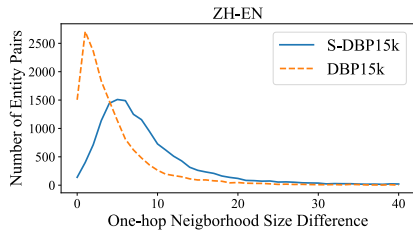
Figure 3: Distribution of difference in the size of neighborhoods of aligned entity pairs on DBP15K$_{ZH-EN}$.

we replace the loss function of NMN after the pre-training phase as:

$$L = \sum_{(r,t)\in\mathbb{L}} \sum_{(r',t')\in\mathbb{C}} \max\{0, d(r,t) - d(r',t') + \gamma\} \quad (11)$$

$$d(r,t) = \|\boldsymbol{h}_r^{match} - \boldsymbol{h}_t^{match}\|_{L_1} \quad (12)$$

where the negative alignments set $\mathbb{C} = \{(r',t')|(r' = r \wedge t' \in \mathcal{C}_r) \vee (t' = t \wedge r' \in \mathcal{C}_t)\}$ is made up of the alignment candidate sets of r and t, $\mathcal{C}_r$ and $\mathcal{C}_t$ are generated in the candidate selection stage described in Sec. 3.4.

Note that our sampling process is non-differentiable, which corrupts the training of weight matrix $\mathbf{W}_s$ in Eq. 2. To avoid this issue, when training $\mathbf{W}_s$, instead of direct sampling, we aggregate all the neighbor information by intuitive weighted summation:

$$\boldsymbol{g}_i^w = \Big(\sum_{p\in N_i} \alpha_{ip} \cdot \sigma(\hat{\boldsymbol{h}}_p \mathbf{W}_{gate}) \cdot \hat{\boldsymbol{h}}_p \Big) \mathbf{W}_N \quad (13)$$

where α_{ip} is the aggregation weight for neighbor p, and is the sampling probability $p(\boldsymbol{h}_p|\boldsymbol{h}_i)$ for p given by Eq. 2. Since the aim of training $\mathbf{W}_s$ is to let the learned neighborhood representations of aligned entities to be as similar as possible, the objective is:

$$L_w = \sum_{(r,t)\in\mathbb{L}} \|\boldsymbol{g}_r^w - \boldsymbol{g}_t^w\|_{L_1} \quad (14)$$

In general, our model is trained end-to-end after pre-training. During training, we use Eq. 11 as the main objective function, and, every 50 epochs, we tune $\mathbf{W}_s$ using Eq. 14 as the objective function.

4 Experimental Setup

Datasets. Follow the common practice of recent works (Sun et al., 2018; Cao et al., 2019; Sun et al., 2020), we evaluate our model on DBP15K (Sun et al., 2017) and DWY100K (Sun et al., 2018) datasets, and use the same split with previous works, 30% for training and 70% for testing. To

Datasets		Ent.	Rel.	Tri.	Tri. Remain in S.
ZH-EN	ZH	66,469	2,830	153,929	26%
	EN	98,125	2,317	237,674	100%
JA-EN	JA	65,744	2,043	164,373	41%
	EN	95,680	2,096	233,319	100%
FR-EN	FR	66,858	1,379	192,191	45%
	EN	105,889	2,209	278,590	100%

Table 1: Summary of DBP15K and S-DBP15k.

Datasets		Ent.	Rel.	Tri.
DBP-WD	DBpedia	100,000	330	463,294
	Wikidata	100,000	220	448,774
DBP-YG	DBpedia	100,000	302	428,952
	YAGO3	100,000	31	502,563

Table 2: Summary of DWY100K.

evaluate the performance of NMN in a more challenging setting, we also build a sparse dataset S-DBP15K based on DBP15K. Specifically, we randomly remove a certain proportion of triples in the non-English KG to increase the difference in neighborhood size for entities in different KGs. Table 1 gives the detailed statistics of DBP15K and S-DBP15K, and the information of DWY100K is exhibited in Table 2. Figure 3 shows the distribution of difference in the size of one-hop neighborhoods of aligned entity pairs. Our source code and datasets are freely available online.[1]

Comparison models. We compare NMN against 12 recently proposed embedding-based alignment methods: MTransE (Chen et al., 2017), JAPE (Sun et al., 2017), IPTransE (Zhu et al., 2017), GCN-Align (Wang et al., 2018), BootEA (Sun et al., 2018), SEA (Pei et al., 2019a), RSN (Guo et al., 2019), MuGNN (Cao et al., 2019), KECG (Li et al., 2019a), AliNet (Sun et al., 2020), GMNN (Xu et al., 2019) and RDGCN (Wu et al., 2019a). The last two models also utilize entity names for alignment.

Model variants. To evaluate different components of our model, we provide two implementation variants of NMN: (1) NMN (w/o nbr-m), where we replace the neighborhood matching part by taking the average of sampled neighbor representations as the neighborhood representation; and (2) NMN (w/o nbr-s), where we remove the sampling process and perform neighborhood matching on all one-hop neighbors.

Implementation details. The configuration we use in the DBP15K and DWY100k datasets is: $\beta = 0.1$, $\gamma = 1.0$, and we sample 5 neigh-

---

[1]https://github.com/StephanieWyt/NMN

bors for each entity in the neighborhood sampling stage (Sec. 3.3). For S-DBP15K, we set β to 1. We sample 3 neighbors for each entity in S-DBP15K$_{ZH-EN}$ and S-DBP15K$_{JA-EN}$, and 10 neighbors in S-DBP15K$_{FR-EN}$. NMN uses a 2-layer GCN. The dimension of hidden representations in GCN layers described in Sec. 3.2 is 300, and the dimension of neighborhood representation g_i described in Sec. 3.5 is 50. The size of the candidate set in Sec. 3.4 is 20 for each entity. The learning rate is set to 0.001.

To initialize entity names, for the DBP15K datasets, we first use Google Translate to translate all non-English entity names into English, and use pre-trained English word vectors *glove.840B.300d*[2] to construct the initial node features of KGs. For the DWY100K datasets, we directly use the pre-trained word vectors to initialize the nodes.

Metrics. Following convention, we use Hits@1 and Hits@10 as our evaluation metrics. A Hits@k score is computed by measuring the proportion of correctly aligned entities ranked in the top k list. A higher Hits@k score indicates better performance.

5 Experimental Results

5.1 Performance on DBP15K and DWY100K

Table 3 reports the entity alignment performance of all approaches on DBP15K and DWY100K datasets. It shows that the full implementation of NMN significantly outperforms all alternative approaches.

Structured-based methods. The top part of the table shows the performance of the state-of-the-art structure-based models which solely utilize structural information. Among them, BootEA delivers the best performance where it benefits from more training instances through a bootstrapping process. By considering the structural heterogeneity, MuGNN and AliNet outperform most of other structure-based counterparts, showing the importance of tackling structural heterogeneity.

Entity name initialization. The middle part of Table 3 gives the results of embedding-based models that use entity name information along with structural information. Using entity names to initialize node features, the GNN-based models, GMNN and RDGCN, show a clear improvement over structure-based models, suggesting that entity

names provide useful clues for entity alignment. In particular, GMNN achieves the highest Hits@10 on the DWY100K datasets, which are the only monolingual datasets (in English) in our experiments. We also note that, GMNN pre-screens a small candidate set for each entity based on the entity name similarity, and only traverses this candidate set during testing and calculating the Hits@k scores.

NMN vs. its variants. The bottom part of Table 3 shows the performance of NMN and its variants. Our full NMN implementation substantially outperforms all baselines across nearly all metrics and datasets by accurately modeling entity neighborhoods through neighborhood sampling and matching and using entity name information. Specifically, NMN achieves the best Hits@1 score on DBP15K$_{ZH-EN}$, with a gain of 2.5% compared with RDGCN, and 5.4% over GMNN. Although RDGCN employs a dual relation graph to model the complex relation information, it does not address the issue of neighborhood heterogeneity. While GMNN collects all one-hop neighbors to construct a topic entity graph for each entity, its strategy might introduce noises since not all one-hop neighbors are favorable for entity alignment.

When comparing NMN and NMN (w/o nbr-m), we can observe around a 2.5% drop in Hits@1 and a 0.6% drop in Hits@10 on average, after removing the neighborhood matching module. Specifically, the Hits@1 scores between NMN and NMN (w/o nbr-m) differ by 3.9% on DBP15K$_{FR-EN}$. These results confirm the effectiveness of our neighborhood matching module in identifying matching neighbors and estimating the neighborhood similarity.

Removing the neighbor sampling module from NMN, i.e., NMN (w/o nbr-s), leads to an average performance drop of 0.3% on Hits@1 and 1% on Hits@10 on all the datasets. This result shows the important role of our sampling module in filtering irrelevant neighbors.

When removing either the neighborhood matching module (NMN (w/o nbr-m)) or sampling module (NMN (w/o nbr-s)) from our main model, we see a substantially larger drop in both Hits@1 and Hits@10 on DBP15K than on DWY100K. One reason is that the heterogeneity problem in DBP15K is more severe than that in DWY100K. The average proportion of aligned entity pairs that have a different number of neighbors is 89% in DBP15K compared to 84% in DWY100K. These results show

Models	DBP$_{ZH-EN}$		DBP$_{JA-EN}$		DBP$_{FR-EN}$		DBP-WD		DBP-YG	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
MTransE (Chen et al., 2017)	30.8	61.4	27.9	57.5	24.4	55.6	28.1	52.0	25.2	49.3
JAPE (Sun et al., 2017)	41.2	74.5	36.3	68.5	32.4	66.7	31.8	58.9	23.6	48.4
IPTransE (Zhu et al., 2017)	40.6	73.5	36.7	69.3	33.3	68.5	34.9	63.8	29.7	55.8
GCN-Align (Wang et al., 2018)	41.3	74.4	39.9	74.5	37.3	74.5	50.6	77.2	59.7	83.8
SEA (Pei et al., 2019a)	42.4	79.6	38.5	78.3	40.0	79.7	51.8	80.2	51.6	73.6
RSN (Guo et al., 2019)	50.8	74.5	50.7	73.7	51.6	76.8	60.7	79.3	68.9	87.8
KECG (Li et al., 2019a)	47.8	83.5	49.0	84.4	48.6	85.1	63.2	90.0	72.8	91.5
MuGNN (Cao et al., 2019)	49.4	84.4	50.1	85.7	49.5	87.0	61.6	89.7	74.1	93.7
AliNet (Sun et al., 2020)	53.9	82.6	54.9	83.1	55.2	85.2	69.0	90.8	78.6	94.3
BootEA (Sun et al., 2018)	62.9	84.8	62.2	85.4	65.3	87.4	74.8	89.8	76.1	89.4
GMNN (Xu et al., 2019)	67.9	78.5	74.0	87.2	89.4	95.2	93.0	**99.6**	94.4	**99.8**
RDGCN (Wu et al., 2019a)	70.8	84.6	76.7	89.5	88.6	95.7	97.9	99.1	94.7	97.3
NMN	**73.3**	**86.9**	**78.5**	**91.2**	**90.2**	**96.7**	**98.1**	99.2	**96.0**	98.2
w/o nbr-m	71.1	86.7	75.4	90.4	86.3	95.8	96.0	98.4	95.0	97.8
w/o nbr-s	73.0	85.6	77.9	88.8	89.9	95.7	98.0	99.0	95.9	98.1

Table 3: Performance on DBP15K and DWY100K.

Models	ZH-EN		JA-EN		FR-EN	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
BootEA	12.2	27.5	27.8	52.6	32.7	53.2
GMNN	47.5	68.3	58.8	78.2	75.0	90.9
RDGCN	60.7	74.6	69.3	82.9	83.6	92.6
NMN	**62.0**	**75.1**	70.3	84.4	86.3	94.0
w/o nbr-m	52.0	71.1	62.1	82.7	80.0	92.0
w/o nbr-s	60.9	74.1	**70.7**	**84.5**	**86.5**	**94.2**

Table 4: Performance on S-DBP15K.

that our sampling and matching modules are particularly important, when the neighborhood sizes of equivalent entities greatly differ and especially there may be few common neighbors in their neighborhoods.

5.2 Performance on S-DBP15K

On the more sparse and challenging datasets S-DBP15K, we compare our NMN model with the strongest structure-based model, BootEA, and GNN-based models, GMNN and RDGCN, which also utilize the entity name initialization.

Baseline models. In Table 4, we can observe that all models suffer a performance drop, where BootEA endures the most significant drop. With the support of entity names, GMNN and RDGCN achieve better performances over BootEA. These results show when the alignment clues are sparse, structural information alone is not sufficient to support precise comparisons, and the entity name semantics are particularly useful for accurate alignment in such case.

NMN. Our NMN outperforms all three baselines on all sparse datasets, demonstrating the effectiveness and robustness of NMN. As discussed in Sec. 1, the performances of existing embedding-based methods decrease significantly as the gap of equivalent entities' neighborhood sizes increases.

Specifically, on DBP15K$_{ZH-EN}$, our NMN outperforms RDGCN, the best-performing baseline, by a large margin, achieving 65%, 53% and 48% on Hits@1 on the entity pairs whose number of neighbors differs by more than 10, 20 and 30, respectively.

Sampling and matching strategies. When we compare NMN and NMN (w/o nbr-m) on the S-DBP15K, we can see a larger average drop in Hits@1 than on the DBP15K (8.2% vs. 3.1%). The result indicates that our neighborhood matching module plays a more important role on the more sparse dataset. When the alignment clues are less obvious, our matching module can continuously amplify the neighborhood difference of an entity pair during the propagation process. In this way, the gap between the equivalent entity pair and the negative pairs becomes larger, leading to correct alignment.

Compared with NMN, removing sampling module does hurt NMN in both Hits@1 and Hits@10 on S-DBP15K$_{ZH-EN}$. But, it is surprising that NMN (w/o nbr-s) delivers slightly better results than NMN on S-DBP15K$_{JA-EN}$ and S-DBP15K$_{FR-EN}$. Since the average number of neighbors of entities in S-DBP15K is much less than that in the DBP15K datasets. When the number of neighbors is small, the role of sampling will be unstable. In addition, our sampling method is relatively simple. When the alignment clues are very sparse, our strategy may not be robust enough. We will explore more adaptive sampling method and scope in the future.

5.3 Analysis

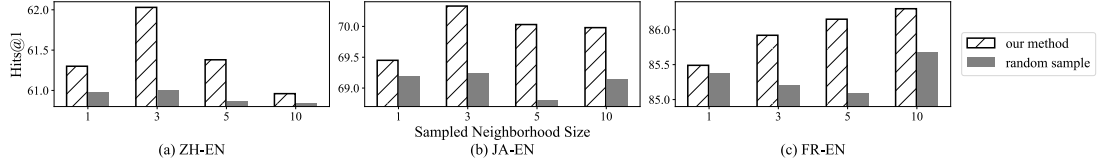**Impact of neighborhood sampling strategies.** To explore the impact of neighborhood sampling

Figure 4: Comparison between our neighborhood sampling strategy and random sampling on S-DBP15K.
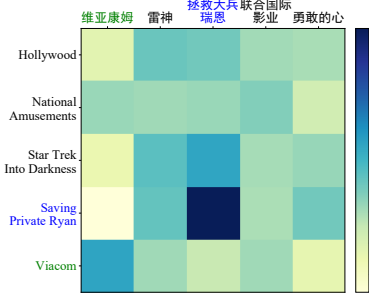

Figure 5: Visualization of attention weights in the neighborhood matching module for the example of *Paramount Pictures*. The green and blue words are two pairs of equivalent neighbors.

strategies, we compare our NMN with a variant that uses random sampling strategy on S-DBP15K datasets. Figure 4 illustrates the Hits@1 of NMN using our designed graph sampling method (Sec. 3.3) and a random-sampling-based variant when sampling different number of neighbors. Our NMN consistently delivers better results compared to the variant, showing that our sampling strategy can effectively select more informative neighbors.

Impact of neighborhood sampling size. From Figure 4, for S-DBP15K$_{ZH-EN}$, both models reach a performance plateau with a sampling size of 3, and using a bigger sampling size would lead to performance degradation. For S-DBP15K$_{JA-EN}$ and S-DBP15K$_{FR-EN}$, we observe that our NMN performs similarly when sampling different number of neighbors. From Table 1, we can see that S-DBP15K$_{ZH-EN}$ is more sparse than S-DBP15K$_{JA-EN}$ and S-DBP15K$_{FR-EN}$. All models deliver much lower performance on S-DBP15K$_{ZH-EN}$. Therefore, the neighbor quality of this dataset might be poor, and a larger sampling size will introduce more noise. On the other hand, the neighbors in JA-EN and FR-EN datasets might be more informative. Thus, NMN is not sensitive to the sampling size on these two datasets.

How does the neighborhood matching module work? In an attempt to understand how our neighborhood matching strategy helps alignment, we visualize the attention weights in the neighborhood matching module. Considering an equivalent entity pair in DBP15K$_{ZH-EN}$, both of which indicate an American film studio *Paramount Pictures*. From Figure 5, we can see that the five neighbors sampled by our sampling module for each central entity are very informative ones for aligning the two central entities, such as the famous movies released by *Paramount Pictures*, the parent company and subsidiary of *Paramount Pictures*. This demonstrates the effectiveness of our sampling strategy again. Among the sampled neighbors, there are also two pairs of common neighbors (indicate *Saving Private Ryan* and *Viacom*). We observe that for each pair of equivalent neighbors, one neighbor can be particularly attended by its counterpart (the corresponding square has a darker color). This example clearly demonstrates that our neighborhood matching module can accurately estimate the neighborhood similarity by accurately detecting the similar neighbors.

6 Conclusion

We have presented NMN, a novel embedded-based framework for entity alignment. NMN tackles the ubiquitous neighborhood heterogeneity in KGs. We achieve this by using a new sampling-based approach to choose the most informative neighbors for each entity. As a departure from prior works, NMN simultaneously estimates the similarity of two entities, by considering both topological structure and neighborhood similarity. We perform extensive experiments on real-world datasets and compare NMN against 12 recent embedded-based methods. Experimental results show that NMN achieves the best and more robust performance, consistently outperforming competitive methods across datasets and evaluation metrics.

References

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1452–1461, Florence, Italy. Association for Computational Linguistics.

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 *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3998–4004. ijcai.org.

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.

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.

Pili Hu and Wing Cheong Lau. 2013. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865*.

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.

Bhushan Kotnis and Vivi Nastase. 2017. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *arXiv preprint arXiv:1708.06816*.

Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019a. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. 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 2723–2732, Hong Kong, China. Association for Computational Linguistics.

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. 2019b. Graph matching networks for learning the similarity of graph structured objects. 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 3835–3845. PMLR.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. 2016. Gated graph sequence neural networks. In *Proceedings of ICLR'16*.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019a. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 3130–3136. ACM.

Shichao Pei, Lu Yu, and Xiangliang Zhang. 2019b. Improving cross-lingual entity alignment via optimal transport. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3231–3237. ijcai.org.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2018. Semi-supervised user geolocation via graph convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2009–2019, Melbourne, Australia. Association for Computational Linguistics.

John W. Raymond, Eleanor J. Gardiner, and Peter Willett. 2002. RASCAL: calculation of graph similarity using maximum common edge subgraphs. *Comput. J.*, 45(6):631–644.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

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, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*.

Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 297–304.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

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.

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.

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 Annual Meeting of the Association for Computational Linguistics*, pages 3156–3161, Florence, Italy. Association for Computational Linguistics.

Xifeng Yan, Philip S. Yu, and Jiawei Han. 2004. Graph indexing: A frequent structure-based approach. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 335–346. ACM.

Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. 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 4431–4441, Hong Kong, China. Association for Computational Linguistics.

Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A vectorized relational graph convolutional network for multi-relational network alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4135–4141. ijcai.org.

Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5429–5435. ijcai.org.

Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4258–4264. ijcai.org.

Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 1943–1949. ijcai.org.