Information Processing and Management 59 (2022) 102790

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Multi-heterogeneous neighborhood-aware for Knowledge Graphs alignment

Weishan Cai [a,b], Yizhao Wang [a], Shun Mao [a], Jieyu Zhan [a], Yuncheng Jiang [a,*]

[a] School of Computer Science, South China Normal University, Guangzhou 510631, China
[b] School of Computer and Information Engineering, Hanshan Normal University, Chaozhou 521041, China

ARTICLE INFO

Keywords:
Entity alignment
Knowledge Graphs
Attribute structure
Attention mechanism
Heterogeneous graph attention

ABSTRACT

Entity alignment is an important task for the Knowledge Graph (KG) completion, which aims to identify the same entities in different KGs. Most of previous works only utilize the relation structures of KGs, but ignore the heterogeneity of relations and attributes of KGs. However, these information can provide more feature information and improve the accuracy of entity alignment. In this paper, we propose a novel Multi-Heterogeneous Neighborhood-Aware model (MHNA) for KGs alignment. MHNA aggregates multi-heterogeneous information of aligned entities, including the entity name, relations, attributes and attribute values. An important contribution is to design a variant attention mechanism, which adds the feature information of relations and attributes to the calculation of attention coefficients. Extensive experiments on three well-known benchmark datasets show that MHNA significantly outperforms 12 state-of-the-art approaches, demonstrating that our approach has good scalability and superiority in both cross-language and monolingual KGs. An ablation study further supports the effectiveness of our variant attention mechanism.

1. Introduction

Knowledge Graphs (KGs) contain various entities and relations that exist in the real world, with the purpose of establishing relations between entities and helping users quickly find the information they need. In the past few decades, many KGs (*e.g.,* DBpedia Lehmann et al., 2015, YAGO Suchanek, Kasneci, & Weikum, 2007, Freebase Bollacker, Evans, Paritosh, Sturge, & Taylor, 2008, and CN-Probase Xu et al., 2017) have been constructed for different application requirements. However, these KGs are still far from complete because different KGs focus on collections of different knowledge. For example, for the same entity, some KGs may focus on the description of certain aspects of the entity, while others may focus on describing the relations between the entity and other entities. It is of great significance to construct a unified large-scale KG from different KGs, thereby improving the effectiveness of various knowledge-driven applications, *e.g.,* knowledge reasoning (Chen, Jia and Xiang, 2020), machine translation (Zhao, Zhang, Zhou, & Zong, 2020), and question answering (Qiao & Hu, 2020; Shin, Jin, Jung, & Lee, 2019). An effective way to solve this problem is **Entity Alignment** (also known as entity matching Lu, Wang, Ma, Xu, & Chen, 2020), which aims to identify the same entities in different KGs. Since pre-aligned entities between KGs are difficult and expensive to obtain, achieving entity alignment between KGs is a challenging task.

Most early approaches for entity alignment (Chen, Tian, Chang, Skiena, & Zaniolo, 2018; Chen, Tian, Yang, & Zaniolo, 2017; Sun, Hu, & Li, 2017; Sun, Hu, Zhang, & Qu, 2018; Trisedya, Qi, & Zhang, 2019; Zhang et al., 2019; Zhu, Xie, Liu, & Sun, 2017)

* Corresponding author.
E-mail addresses: caiws@m.scnu.edu.cn (W. Cai), wyz0714@m.scnu.edu.cn (Y. Wang), shunm@m.scnu.edu.cn (S. Mao), zhanjieyu@scnu.edu.cn (J. Zhan), ycjiang@scnu.edu.cn (Y. Jiang).

https://doi.org/10.1016/j.ipm.2021.102790
Received 8 April 2021; Received in revised form 14 August 2021; Accepted 4 October 2021
Available online 29 October 2021
0306-4573/

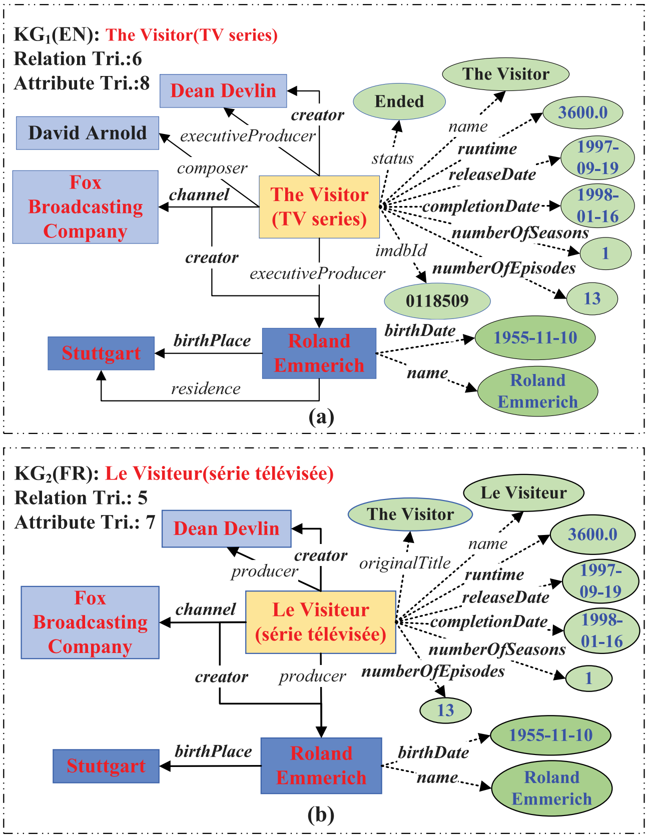

Fig. 1. An example for the similarity of relation and attribute structures between $KG_1(EN)$ in (a) and $KG_2(FR)$ in (b). The rectangles and ellipses indicate entities and attribute values respectively, while the solid and dashed line edges indicate relations and attributes respectively. The rectangles in red font and the ellipses in blue font indicate similar entities and attribute values between two KGs, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

rely on the translation-based models (*e.g.,* TransE Bordes, Usunier, García-Durán, Weston, & Yakhnenko, 2013). Some additional information of KGs are utilized by some approaches to obtain more relevant knowledge, such as attributes (Sun et al., 2017; Trisedya et al., 2019), attribute values (Trisedya et al., 2019; Zhang, Sun et al., 2019), topology features (Zhu et al., 2017), entity description information (Chen et al., 2018), *etc.* However, these translation-based models ignore the similarity of neighborhood structures of entities between two KGs. Recent works mainly utilize Graph Neural Networks (GNNs) (*e.g.,* Graph Convolutional Networks (GCNs) Kipf & Welling, 2017 and Graph Attention Networks (GATs) Veličković et al., 2018) to embed entities into the same low-dimensional vector space and aggregates the adjacent features of each entity. Since the outstanding performance of GNNs in employing deep neural networks to aggregate feature information of neighboring nodes, these GNNs-based models make the aggregated embeddings more powerful (Zhang, Song, Huang, Swami and Chawla, 2019). Even the above approaches have achieved promising performance, there are still some limitations.

Firstly, most GCN-based and GAT-based models (*e.g.,* HGCN Wu, Liu, Feng, Wang, Yan and Zhao, 2019, RDGCN Wu et al., 2019, and NMN Wu, Liu, Feng, Wang, & Zhao, 2020) do not consider the attribute structures of KGs, but only obtain the similarity of relation structures. However, the common KG has two structures, *i.e.,* the relation structure and attribute structure. For instance, the left half of Fig. 1 is the relation structure between entities and entities, and the right half is the attribute structure between entities and attribute values. The relations (*i.e.,* the solid line edges) are used to describe the connection between entities, and the attributes (*i.e.,* the dashed line edges) are the inherent characteristic of entities.

Secondly, most of GNNs-based models (*e.g.,* RDGCN Wu, Liu, Feng, Wang, Yan et al., 2019, NMN Wu et al., 2020, NAEA Zhu, Zhou, Wu, Tan, & Guo, 2019, and KECG Li et al., 2019) use traditional GNNs (*e.g.,* GCNs and GATs) to aggregate the neighborhood information of entities, and make similarity judgments of aligned entities on this basis. Nevertheless, these models ignore the heterogeneity of KG, since traditional GNNs run on the undirected and unlabeled graphs. The heterogeneity is an intrinsic property of KG. As Fig. 1 shown, a KG generally contains various types of relations and attributes, which is different from the homogeneous graphs with only one kind of edges and nodes. Due to the complexity of heterogeneous graph, traditional GNNs cannot be directly applied to heterogeneous graph.

To address the problems of the two aspects, we propose a **M**ulti-**H**eterogeneous **N**eighborhood-**A**ware model (MHNA) for entity alignment in this paper, which simultaneously aggregates multi-source information from the relation and attribute structures of

KGs. We notice that attribute triples account for a large part of KGs, and aligned entities often have similar entity names and neighborhood structures, including the relations, attributes and attribute values, not just similar relation structures. Example of two KGs [1] are illustrated in Fig. 1. It can be seen that the movies *The Visitor*(*TV series*) in $KG_1(EN)$ and *Le Visiteur*(*série télévisée*) in $KG_2(FR)$ have not only two similar relations and three associated entities, but also five similar attributes and five attribute values, so is the entity *Roland Emmerich*. Besides, the names of aligned entities in Fig. 1 are also highly similar. We believe that comprehensive consideration of the relation structures, attribute structures and heterogeneous feature information of KGs can make the aggregation of entity embeddings more powerful and improve the accuracy of entity alignment. Our main contributions are summarized as follows:

- Propose a multi-heterogeneous neighborhood-aware model (MHNA) for EA, which simultaneously captures and aggregates multi-source information of aligned entities and their heterogeneous neighbors, including the entity name, relations, attributes, and attribute values.
- Design a variant attention mechanism of heterogeneous graphs, which adds the feature information of relations and attributes to the calculation of attention mechanism coefficients. Benefiting from such attention mechanism, MHNA can take the different importance of relations and attributes to entities into consideration.
- Evaluate MHNA on three well-known benchmark datasets, which are rarely verified simultaneously in previous works. Experimental results show MHNA significantly outperforms 12 state-of-the-art approaches, and achieves the best alignment performance with less training data.

The rest of the paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 introduces our proposed approach MHNA. Section 4 reports the experimental results compared with 12 state-of-the-art alignment approaches. Finally, Section 5 summarizes this paper and discusses future research direction.

2. Related work

2.1. Embedding-based entity alignment

In the past few years, there are many works on entity alignment (Sun, 2020; Zeng, Li, Hou, Li, & Feng, 2021). Early approaches (*e.g.,* MTransE Chen et al., 2017, JAPE Sun et al., 2017, IPTransE Zhu et al., 2017, BootEA Sun et al., 2018, KDCoE Chen et al., 2018, MultiKE Zhang, Sun et al., 2019, and AttrE Trisedya et al., 2019) basically utilize TransE (Bordes et al., 2013) to obtain the embeddings of entities, relations, and attributes in KGs. TransE is a translation-based model, which maps entities to low-dimensional vectors to obtain entity semantics. The goal of TransE is to interpret the relation vector as the translation from the head entity vector to the tail entity vector, *i.e.,* $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, if (h, r, t) is a relation triple in KG. For example, MTransE first uses two TransE models to train the network structures of two KGs respectively and obtains two vector spaces of the KGs, then uses some pre-aligned entities to train a linear transformation to align the two vector spaces. Therefore, these approaches rely on large number of pre-aligned entities that are difficult and expensive to obtain.

Some other approaches (*e.g.,* IPTransE Zhu et al., 2017, BootEA Sun et al., 2018, and AHAB Chen, Gu, Tian, & Chen, 2019) focus on iteratively discovering more new aligned entities. IPTransE (Zhu et al., 2017) takes an iterative approach to discover those highly self-confident aligned entities. BootEA (Sun et al., 2018) exploits a bootstrapping process to seek the maximize alignment likelihood on all labeled and unlabeled entities. Nonetheless, incorrect aligned entities produced by iteration processes may affect the accuracy of subsequent iterations. Some approaches even assume that the pre-aligned entities and pre-aligned relations can be easily found in advance.

Current state-of-the-art approaches for entity alignment tend to get the similar structure information between two KGs by using GCNs or GATs. The GCN-based models (*e.g.,* GCN-Align Wang, Lv, Lan, & Zhang, 2018, HMAN Yang et al., 2019, VR-GCN Ye, Li, Fang, Zang, & Wang, 2019, and GMNN Xu et al., 2019) use convolutional operators to aggregate the embeddings of entity neighbors and finally make the embeddings of aligned entity pair close to each other. Instead, the GAT-based models (*e.g.,* NAEA Zhu et al., 2019 and KECG Li, Cao et al., 2019) employ self-attention mechanism to aggregate the neighboring features of entities. Some other approaches (*e.g.,* Wu, Liu, Feng, Wang, Yan, Zhao, 2019, RDGCN Wu, Liu, Feng, Wang, Yan et al., 2019, NMN Wu et al., 2020, and MuGNN Cao et al., 2019) are based on both GCNs and GATs. However, most of these approaches do not consider the attribute structures of KGs, but only consider the relation structures.

To obtain more knowledge, the additional information of KGs is utilized in some approaches, including topology features, attributes, attribute values, entity name and description information. JAPE (Sun et al., 2017) is the first model to consider both relation and attribute structures, after that AttrE (Trisedya et al., 2019), MultiKE (Zhang, Sun et al., 2019), GCN-Align (Wang et al., 2018), HMAN (Yang et al., 2019), AttrGNN (Liu et al., 2020), COTSAE (Yang, Liu, Zhao, Wang, & Xie, 2020), and JarKA (Chen, Zhang, Tang, Chen and Li, 2020) do the same. Entity descriptions are considered in KDCoE (Chen et al., 2018) and HMAN (Yang et al., 2019), and topology features are considered in IPTransE (Zhu et al., 2017) and HMAN (Yang et al., 2019). Entity names are also used as the input features for entities in RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019), HGCN (Wu, Liu, Feng, Wang, Yan, Zhao, 2019), MultiKE (Zhang, Sun et al., 2019), CEA (Zeng, Zhao, Tang, & Lin, 2020), and AttrGNN (Liu et al., 2020).

[1] The data is extracted from EN-FR(V2), which is one version of our experimental datasets.

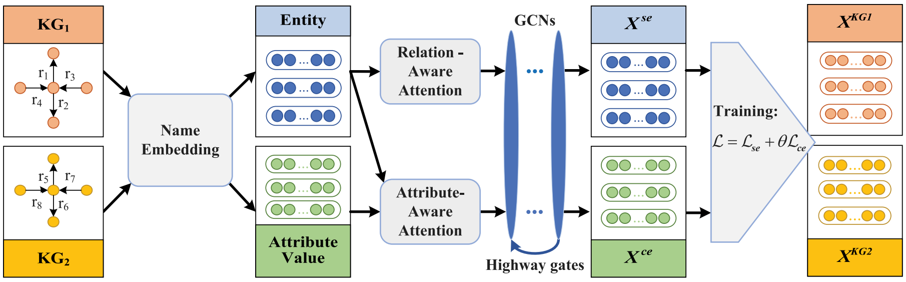

Fig. 2. The overall architecture of MHNA.

Since different additional information describes various aspects of the entities, the fusion of various additional information is also an important research direction of this problem. Therefore, we aggregates multi-source information from the relation and attribute structures of KGs. We especially design a variant attention mechanism to consider the heterogeneity of relations and attributes of KGs. As we will show later in this paper, the representation learning of entities based on the fusion of multi-heterogeneous neighborhood information can improve the accuracy of entity alignment.

2.2. Heterogeneous graph embedding

Compared with homogeneous graphs, heterogeneous graphs are a kind of network with different types of nodes or edges. Traditional GNNs just consider one type of edges, but ignore the labels of edges and nodes. Although GATs use attention mechanism to add weighted features of neighboring nodes, they still do not consider the heterogeneity of edge labels or node labels. Heterogeneous graph embedding models can mine more meaningful information by assigning different weights to different types of nodes and edges. Therefore, heterogeneous graph embedding has attracted a lot of widespread attention.

RGCNs (Schlichtkrull et al., 2018) model heterogeneous graphs by employing a weight matrix for each relation, but they are difficult to train due to the need for an excessive set of parameters. HAN (Wang et al., 2019) proposes a hierarchical attention mechanism, which includes both of node-level and semantic-level attentions. HAN can take the importance of nodes and meta-paths into consideration simultaneously, through learning the weights from the node-level and semantic-level attentions. HetGNN (Zhang, Song et al., 2019) also proposes a hierarchical attention mechanism, but selects LSTM as node-level aggregation. These are also some other heterogeneous graph embedding models, including GEM (Liu et al., 2018), HERec (Shi, Hu, Zhao, & Philip, 2018), GTN (Yun, Jeong, Kim, Kang, & Kim, 2019), HGAT (Linmei, Yang, Shi, Ji, & Li, 2019), MEIRec (Fan et al., 2019), and GAS (Li, Qin, Liu, Yang and Li, 2019), *etc.* However, these models can only be applied to these graphs with fewer types of nodes (or edges). Because they use different graph embedding layers for each type of neighbors, then aggregate the outputs of all graph embedding layers as the final representation of the node. When there are a large number of types of nodes or edges, their training will be prohibitively expensive. Unfortunately, the types of relations and attributes in common KGs reach hundreds or even thousands, so these models cannot be directly used for heterogeneous graph embedding of KGs.

Inspired by the above work, there are some works utilize heterogeneous graph embedding to achieve entity alignment. RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019) incorporates relation information via attentive interactions between the KGs and their dual relation counterpart, but does not consider the attribute structure of KGs, leaving much room for improvement. CG-MuAlign (Zhu et al., 2020) uses the hierarchical attention mechanism to learn the embedding of aligned entities from the node-level and edge-level attentions, and applies different linear transformations to each relation. As mentioned above, CG-MuAlign cannot be used for graph embedding of common KGs with many types of relations and attributes. Therefore, we propose a variant attention mechanism in this paper, which adds the heterogeneity of relations and attributes to the calculation of attention mechanism coefficients. Our approach considers the heterogeneity of relations and attributes when aggregating the heterogeneous neighborhood information of entities, and keeps the training complexity within a controllable range.

3. Proposed approach

In this section, we will explain how MHNA aggregates the multi-heterogeneous neighborhoods information to achieve entity alignment. Fig. 2 shows the overall architecture of MHNA, which includes the following hierarchical structures: (1) **Name Embedding.** Entity names and attribute values are pre-embedded by pre-trained word vectors. (2) **Relation-Aware and Attribute-Aware Attention Embeddings.** We design a variant attention mechanism to aggregate heterogeneous neighborhood features of entities on the relation-level and attribute-level of KGs, respectively. (3) **Incorporating Structural Embedding.** GCNs and highway gates are used to further aggregate the information of neighborhood structures, and output two entity embeddings based on relation and attribute structures, respectively. (4) **Entity Alignment and Training.** The above two aware attention embeddings of entities are used to measure the similar distances of candidate-entities. Our objective is to make two kinds of embeddings of the aligned entity pair as close as possible at the same time.

Table 1
Notations and descriptions.

Notation	Description
$\mathbf{X}^{e\_init}$	Initial name embedding matrix of entities.
$\mathbf{X}^{se}$	Output embedding matrix based relation-aware attention.
$\mathbf{X}^{ce}$	Output embedding matrix based attribute-aware attention.
$\mathbf{X}^{r}$	Embedding matrix of relations.
$\mathbf{X}^{m}$	Embedding matrix of attributes.
$\mathbf{X}^{v\_init}$	Initial embedding matrix of attribute values.
$\mathbb{R}^{d}$	d-dimensional real-valued space.
$\odot$	Element-wise multiplication operation.
$\Vert$	Vector concatenation operation.
σ	ReLU activation function.
η	LeakyReLU nonlinearity function.
ψ	Sigmoid function.

3.1. Problem statement

For notations, we represent a KG as a 6-tuple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{M}, \mathcal{V}, \mathcal{T}, \mathcal{K})$, where $\mathcal{E}, \mathcal{R}, \mathcal{M}$ and $\mathcal{V}$ respectively denote the sets of entities, relations, attributes and attribute values; $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ denotes the set of relation triples, and $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{M} \times \mathcal{V}$ denotes the set of attribute triples. Let $\mathcal{G}_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{M}_1, \mathcal{V}_1, \mathcal{T}_1, \mathcal{K}_1)$, $\mathcal{G}_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{M}_2, \mathcal{V}_2, \mathcal{T}_2, \mathcal{K}_2)$ denote two KGs to be aligned. We put $\mathcal{G}_1$ and $\mathcal{G}_2$ together as a primal graph $\mathcal{G}_a = (\mathcal{E}_a, \mathcal{R}_a, \mathcal{M}_a, \mathcal{V}_a, \mathcal{T}_a, \mathcal{K}_a)$, where $\mathcal{E}_a = \mathcal{E}_1 \cup \mathcal{E}_2$ is the union of all entities in $\mathcal{G}_1$ and $\mathcal{G}_2$, and others are similar. So our goal is to find a set of aligned entities, $\mathbb{L} = \{(e_i, e_j) \in \mathcal{E}_1 \times \mathcal{E}_2 | e_i \equiv e_j\}$, where $\equiv$ denotes the entity alignment.

Formally, we use bold lowercase letter for embedding vector and bold uppercase letter for embedding matrix. In particular, $\mathbf{x}_i^t$ represents the embedding of a type of an object, and $\mathbf{X}^t$ represents the embedding matrix of a type of objects, where the t denotes t-th type of objects and the i denotes i-th object. Specific notations and their descriptions are listed in Table 1.

3.2. Name embedding

As known, it is necessary to embed objects before training the embedding-based model. Since all names are composed of word or character sequences, we use word embedding to pre-embed the entities and attribute values, just like the literal embedding in MultiKE (Zhang, Sun et al., 2019). Fortunately, there are some pre-trained word embedding results in the field of natural language processing.

$$EM\left(w_j\right) = \begin{cases} word\_embed(w_j) & \text{if } w_j \text{ has a word embedding} \\ char\_embed(w_j) & \text{otherwise,} \end{cases} \tag{1}$$

where $EM(w_i)$ is defined as an embedding function to obtain the word embedding; $word\_embed(\cdot)$ returns the corresponding word embedding that comes from pre-trained English word embedding; $char\_embed(\cdot)$ returns the mean vector of character embeddings that are pre-trained by Skip-Gram (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013).

Let $N_{e_i} = (w_1, w_2, \ldots, w_n)$ denote the entity name of $e_i \in \mathcal{E}_a$, consisting of n words or characters, we construct its name embedding $\mathbf{x}_i^e \in \mathbf{X}^{e\_init} \in \mathbb{R}^{|\mathcal{E}_a| \times d}$ as follows:

$$\mathbf{x}_i^e = com(\sum_{w_j \in N_{e_i}} EM(w_j)), \tag{2}$$

where $com(\cdot)$ is a SUM compositional function, which also can be LSTM or N-gram (Trisedya et al., 2019). The initial embeddings of attribute values $\mathbf{X}^{v\_init} \in \mathbb{R}^{|\mathcal{V}_a| \times d}$ are also constructed in this way.

3.3. Relation-aware attention embedding

To better capture the heterogeneity of relations, we do not directly construct the relation embedding from its name. Let $\mathbf{X}^r \in \mathbb{R}^{|\mathcal{R}_a| \times 2d}$ denote the embedding matrix of relations, the embedding of relation r_t is approximated by averaging the embeddings of its head entities H_t and tail entities T_t from the relation triples $\mathcal{T}_a$ as follows:

$$\mathbf{x}_t^r = \sigma\left[\frac{\sum_{e_i \in H_t} b_i^1 \mathbf{x}_i^e}{|H_t|} \Vert \frac{\sum_{e_j \in T_t} b_j^2 \mathbf{x}_j^e}{|T_t|}\right], \tag{3}$$

where $\mathbf{x}_i^e, \mathbf{x}_j^e \in \mathbf{X}^{e\_init}$ are the name embeddings of entity e^i and e^j; $|H_t|$ and $|T_t|$ respectively denote the set sizes of head entities and tail entities in $\mathcal{T}_a$ connected by the relation r_t; $b^1 \in \mathbb{R}^{|\mathcal{E}_a|}$ denotes weight vector parameter of entities, and is used to emphasize the different contributions of associated head entities to the relation r_t, as well as b^2 for the associated tail entities.

Before aggregating the information from the neighbors of entity, we should notice that different relations play different roles and show different importance in learning entity embeddings. So we design a variant attention mechanism of heterogeneous graphs to learn the weights of neighbors with different relations to the associated entities. Different from the attention mechanism in GATs that

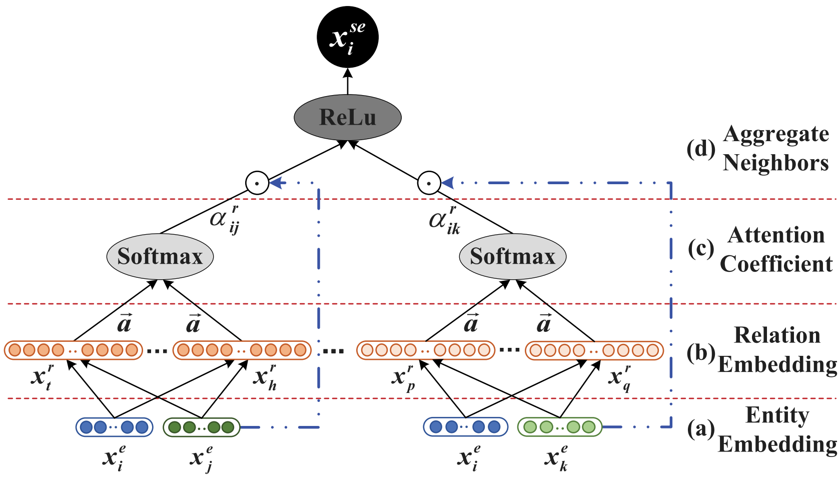

Fig. 3. An illustration of the variant attention mechanism.

only considers the neighboring nodes, we add inter-entity relations to the calculation of attention coefficients. Given a head-to-tail entity pair (e_i, e_j), that may be connected via different relations, the coefficient e_{ij}^r indicates the importance of e_j to e_i:

$$e_{ij}^r = att_{relation}(e_i, e_j; G_{ij}^r) , \tag{4}$$

where $att_{relation}(\cdot)$ is a function to perform relation-aware attention; $G_{ij}^r \subseteq \mathcal{T}_a$ is the set of relation triples with e_i as head entity and e_j as tail entity.

As shown in Eq. (4), the attention coefficient depends on not just the features of entities, but also the features of relations between entities. Moreover, e_{ij}^r is asymmetric, *i.e.*, the importance of e_i to e_j and the importance of e_j to e_i can be quite different. It shows relation-aware attention can preserve this asymmetry, which is a critical property of heterogeneous graphs. Then, the definition of attention coefficient e_{ij}^r can be refined and normalized by the Softmax function:

$$\alpha_{ij}^r = \frac{\exp\left(\eta\left(\sum_{(e_i,r_t,e_j)\in G_{ij}^r} \mathbf{a}_1^T([\mathbf{x}_i^e \parallel \mathbf{x}_j^e] \odot \mathbf{x}_t^r)\right)\right)}{\sum_{e_k\in N_i^r} \exp\left(\eta\left(\sum_{(e_i,r_t,e_k)\in G_{ik}^r} \mathbf{a}_1^T([\mathbf{x}_i^e \parallel \mathbf{x}_k^e] \odot \mathbf{x}_t^r)\right)\right)} , \tag{5}$$

where $\mathbf{a}_1 \in \mathbb{R}^{2d\times 1}$ is the attention parameter; $N_i^r \subseteq \mathcal{E}_a$ denotes the set of relation neighbors of e_i (including itself).

After that, the new embedding of entity e^i based on relation structure can be constructed by aggregating the projection features of its relation neighbors:

$$\mathbf{x}_i^{se} = \sigma\left(\sum_{e_j\in N_i^r} \alpha_{ij}^r \mathbf{x}_j^e\right). \tag{6}$$

To better explain the variant attention mechanism of heterogeneous graphs, we also provide a brief explanation in Fig. 3. The (b)-level is newly added to the attention mechanism, which adds relation embeddings to the calculation of attention coefficient. So the attention coefficient α_{ij}^r can capture the semantic information of different relations between entity e_i and its neighbor e_j.

As we analyzed above, the initial name embeddings of entities also can provide important alignment information to entity alignment. Therefore, we get the final output by mixing the name embeddings and relation-aware attention embeddings of entities. *i.e.*, the final output is the balance between $\mathbf{X}^{e\_init}$ and $\mathbf{X}^{se}$ with a tradeoff-parameter β_1:

$$\mathbf{X}^{se} = \beta_1 \mathbf{X}^{se} + \mathbf{X}^{e\_init}. \tag{7}$$

3.4. Attribute-aware attention embedding

We construct the attribute embeddings according to attribute structures in the same way as relation structures. Let $\mathbf{X}^m \in \mathbb{R}^{|\mathcal{M}_a|\times 2d}$ represent the embedding matrix of attributes. For the attribute x_g, its embedding $\mathbf{x}_g^m$ is constructed by approximately averaging the embeddings of all its associated head entities and tail attribute values, as shown below:

$$\mathbf{x}_g^m = \sigma\left[\frac{\sum_{e_h\in H_g} c_h^1 \mathbf{x}_h^e}{|H_g|} \parallel \frac{\sum_{v_k\in V_g} c_k^2 \mathbf{x}_k^v}{|V_g|}\right], \tag{8}$$

where $\mathbf{x}_h^e$ and $\mathbf{x}_k^v$ denote the embeddings of head entity e^h and tail attribute value v^k; $|H_g|$ and $|V_g|$ denote the set sizes of associated head entities and tail attribute values of x_g, respectively; c^1 and c^2 denote the weight vector parameters, used to emphasize the different contributions of associated entities and attribute values to x_g, respectively.

For a given attribute triple $(e_p, m_g, v_q) \in \mathcal{K}_a$, composing of entity e_p, attribute m_g, and attribute value v_q, we use the variant attention mechanism to calculate the coefficient e_{pq}^m of attribute-aware attention that represents the weight of v_q to e_p, and normalize it by the Softmax function:

$$\alpha_{pq}^m = \frac{\exp(\eta(\sum_{(e_p,m_g,v_q)\in G_{pq}^m} \mathbf{a}_2^T([\mathbf{x}_p^e \parallel \mathbf{x}_q^v] \odot \mathbf{x}_g^m)))}{\sum_{v_k \in N_p^m} \exp(\eta(\sum_{(e_p,m_g,v_k)\in G_{pk}^m} \mathbf{a}_2^T([\mathbf{x}_p^e \parallel \mathbf{x}_k^v] \odot \mathbf{x}_g^m)))} , \tag{9}$$

where $\mathbf{x}_p^e, \mathbf{x}_g^m$ and $\mathbf{x}_q^v$ denote the embeddings of e_p, m_g and v_q; $\mathbf{a}_2 \in \mathbb{R}^{2d\times 1}$ denotes the attention parameter; $G_{pq}^m \subseteq \mathcal{K}_a$ denotes the set of attribute triples with e_p as head and v_q as tail; $N_p^m \subseteq \mathcal{V}_a$ denotes the set of associated attribute values of entity e_p.

Combining the attention coefficients and name embeddings of entities, we can get new embedding of entity e_p based on the attribute structure as follows:

$$\mathbf{x}_p^{ce} = \sigma(\sum_{v_q \in N_p^m} \alpha_{pq}{}^m \mathbf{x}_q^v) . \tag{10}$$

We get the final attribute-aware attention embeddings of entities in the same way as relation-aware attention embeddings.

$$\mathbf{X}^{ce} = \beta_2 \mathbf{X}^{ce} + \mathbf{X}^{e\_init} , \tag{11}$$

where β_2 also is a tradeoff-parameter to balance weight between $\mathbf{X}^{ce}$ and $\mathbf{X}^{e\_init}$.

3.5. Incorporating structural embedding

After the processing of relation-aware attention and attribute-aware attention, we get the entity embeddings $\mathbf{X}^{se}$ and $\mathbf{X}^{ce}$ that respectively contain the relation and attribute structures information of two KGs. Inspired by previous studies (Wu, Liu, Feng, Wang, Yan et al., 2019; Yang et al., 2019), two-layers of GCNs with highway gates are adopted on the entity embeddings $\mathbf{X}^{se}$ and $\mathbf{X}^{ce}$ respectively, to collect more evidences about the similarity of relation and attribute structures. The purpose of adding highway networks is to control the noises of neighborhood information transmitted to nodes in multilayer neural network (Rahimi, Cohn, & Baldwin, 2018).

$$X^{(f,l)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^{(f,l-1)} W^{(f,l)}) , \tag{12}$$

$$T(X^{(f,l)}) = \psi(X^{(f,l)} W_T^{(f,l)} + b_T^{(f,l)}) , \tag{13}$$

$$X^{(f,l+1)} = T(X^{(f,l)}) \odot X^{(f,l)} + (1 - T(X^{(f,l)})) \odot X^{(f,l-1)} , \tag{14}$$

where $f = \{se, ce\}$ denotes the relation-aware or attribute-aware embedding; $\tilde{A} = A + I$ is the adjacency matrix of relation structures with added self-connections, in which I is an identity matrix; $\tilde{D}$ is the diagonal node degree matrix of $\tilde{A}$; $W^{(f,l)}$ denotes the learnable parameters in the l-th layer; $X^{(f,l)}$ denotes the output of l-th layer; $W_T^{(f,l)}, b_T^{(f,l)}$ respectively denote the weight matrix and bias vector. Please note that we use different parameters on $\mathbf{X}^{se}$ and $\mathbf{X}^{ce}$ to better obtain the different features of the relation and attribute structures of two KGs.

3.6. Entity alignment and training

Alignment. After obtaining the final embeddings from incorporating structural embedding, we use the Manhattan distance (L_1) to measure the similarity of candidate–entity pair. A smaller distance indicates a higher probability of the candidate–entity pair being aligned. To capture the various aspect features of entities, the previous approaches (*e.g.,* HGCN Wu, Liu, Feng, Wang, Yan, Zhao, 2019, HMAN Yang et al., 2019, and NMN Wu et al., 2020) usually concatenate multiple embeddings of entities and then use the concatenated embeddings in the loss function. Since the relation and attribute information of entities may be quite diverse, the contributions of relation-aware and attribute-aware embeddings to entity alignment should also be different. Therefore, we calculate the distances of candidate-entities in $\mathbf{X}^{se}$ and $\mathbf{X}^{ce}$ respectively that are used in the following loss function:

$$d_f(e_1, e_2) = \left\| \mathbf{x}_{e_1}^f - \mathbf{x}_{e_2}^f \right\|_{L_1} , \tag{15}$$

where e_1 from $\mathcal{G}_1$ and e_2 from $\mathcal{G}_2$; $f = \{se, ce\}$; $\mathbf{x}_{e_1}^f$ and $\mathbf{x}_{e_2}^f$ respectively denote one type embeddings of entity e_1 and e_2.

Training. Our training goal is to keep the embedding distance of aligned entity pair (positive pair) as small as possible, while the embedding distance of unaligned entity pair (negative pair) as large as possible. We use the margin-based ranking loss function over the training data:

$$\mathcal{L} = \sum_{(p,q)\in\mathbb{L},(p',q')\in\mathbb{L}'_{se}} [d_{se}(p,q) - d_{se}(p',q') + \gamma_1]_+ + \theta \cdot \sum_{(p,q)\in\mathbb{L},(p'',q'')\in\mathbb{L}'_{ce}} [d_{ce}(p,q) - d_{ce}(p'',q'') + \gamma_2]_+ , \tag{16}$$

where $[\cdot]_+ = max\{0, \cdot\}$; $\mathbb{L}$ denotes the set of aligned entities, and $\mathbb{L}'_{se}$ and $\mathbb{L}'_{ce}$ represent the negative pairs of relation-aware embeddings and attribute-aware embeddings, respectively; $\gamma_1, \gamma_2 > 0$ are margin hyper-parameters for separating positive and negative pairs, and θ is a tradeoff-parameter to balance the two loss parts.

K-nearest neighbor sampling (Sun et al., 2018) is used to generate the $\mathbb{L}'_{se}$ and $\mathbb{L}'_{ce}$ based on relation-aware and attribute-aware embeddings, maximizing the distance between negative and positive pairs. Under the guidance of training data, the model is optimized through back propagation to learn the embeddings of entities. The training procedure of MHNA is shown in Algorithm 1.

Algorithm 1: Training Procedure of MHNA.

Input: (1) adjacency matrices of $\mathbb{G}_a$; (2) aligned entities $\mathbb{L}$; (3) weight parameters Θ and hyper-parameters $\beta_1, \beta_2, \gamma_1, \gamma_2, \theta$.
Output: entity embeddings $\mathbf{X}^{se}, \mathbf{X}^{ce}$.
Calculate the entity name and attribute value embedding matrices $\mathbf{X}^{e\_init}$ and $\mathbf{X}^{v\_init}$ using Eqs. (1) and (2);
while *not converged* **do**
    `// Relation-Aware Attention Embedding`
    Calculate the relation embedding matrix $\mathbf{X}^r$ using Eq. (3);
    Calculate the relation-aware attention embedding matrix $\mathbf{X}^{se}$ using Eqs. (5)~(7);
    Update $\mathbf{X}^{se}$ by GCNs and highway networks using Eqs. (12)~(14);
    `// Attribute-Aware Attention Embedding`
    Calculate the attribute embedding matrix $\mathbf{X}^m$ using Eq. (8);
    Calculate the attribute-aware attention embedding matrix $\mathbf{X}^{ce}$ using Eqs. (9)~(11);
    Update $\mathbf{X}^{ce}$ by GCNs and highway networks using Eqs. (12)~(14);
    `// Loss function`
    Construct the negative pairs $\mathbb{L}'_{se}$, $\mathbb{L}'_{ce}$ from $\mathbf{X}^{se}$, $\mathbf{X}^{ce}$ respectively by K-nearest neighbor sampling every 10 epochs;
    Calculate objective function $\mathcal{L}$ using Eqs. (15) and (16);
    Update model parameters Θ by $\frac{\partial \mathcal{L}}{\partial \Theta}$;
end

4. Experiments

In this section, we conduct a series of experiments on three well-known benchmark datasets to prove that our model is not only outperforms all existing methods but also is robust. The code is available on GitHub.[2]

4.1. Datasets and settings

Datasets. In this paper, we evaluate MHNA on three well-known benchmark datasets, including cross-lingual and monolingual datasets. Table 2 lists the statistics of all datasets.

- WK31-15K (Sun, 2020) contains two cross-lingual datasets collected from DBpedia: English–German and French–English, each version of which includes 15K aligned entity pairs. Moreover, two versions of each pair of cross-lingual KGs generates are generated: V1 is sparse and directly gained by using the IDS algorithm, while V2 is twice denser than V1. Therefore, this WK31-15K version used in this paper contains four subsets: EN-DE(V1), EN-DE(V2), EN-FR(V1), and EN-FR(V2).
- DBP-15K (Sun et al., 2017) also comes from cross-lingual DBpedia like WK31-15K and contains three subsets: Chinese–English(ZH-EN), Japanese–English(JA-EN), and French–English(FR-EN). This original DBP-15K only contains the attribute sets, but not complete attribute triples. Therefore we crawl the attribute values through a web crawler procedure to construct a complete set of attribute triples. Unfortunately, the attribute values of ZH-EN is unobtainable, because the attribute links have been abandoned. So only the JA-EN and FR-EN can be used in our model, excluding the ZH-EN. To distinguish from WK31-15K, we mark the two subsets as JA-EN(DBP) and FR-EN(DBP) respectively.
- DWY-100K (Sun et al., 2018; Zhang, Sun et al., 2019) contains two large-scale monolingual KGs: DBpedia and Wikidata (DBP-WD), DBpedia and YAGO3(DBP-YG), each of which has 100K aligned entity pairs.

Model Variants: To evaluate the different components of MHNA, we provide an implementation variant model for an ablation study. This model is a MHNA model without considering the attribute-aware attention embedding, marked as $\text{MHNA}_{(\text{w/o m})}$.

Metrics: Following convention, we use Hits@k and MRR (Mean Reciprocal Rank) as our experimental evaluation metrics. Hits@k is computed by measuring the proportion of correctly aligned entities ranked in the top-k candidates. MRR is the average of the reciprocal ranks. Higher Hits@k and MRR scores indicate better performance of EA.

4.2. Overall performance on DBP-15K and WK31-15K

Comparison Models: For DBP-15K and WK31-15K, we compare MHNA with 9 previous state-of-the-art alignment models (mentioned in Section 2.1): MTransE (Chen et al., 2017), IPTransE (Zhu et al., 2017), JAPE (Sun et al., 2017), BootEA (Sun et al., 2018), GCN-Align (Wang et al., 2018), AttrE (Trisedya et al., 2019), SEA (Pei, Yu, Hoehndorf, & Zhang, 2019), RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019), and NMN (Wu et al., 2020).

[2] https://github.com/cwswork/MHNA.

Table 2
Statistics of datasets.

Datasets			Entity	Rel.	Rel tr.	Attr	Value	Attr tr.
WK31-15K	EN-DE(V1)	EN	15,000	215	47,676	286	49,956	837,555
		DE	15,000	131	50,419	194	57,661	156,150
	EN-DE(V2)	EN	15,000	169	84,867	171	45,805	81,988
		DE	15,000	96	92,632	116	57,652	186,335
	EN-FR(V1)	EN	15,000	267	47,334	308	45,783	73,121
		FR	15,000	210	40,864	404	39,772	67,167
	EN-FR(V2)	EN	15,000	193	96,318	189	36,391	66,899
		FR	15,000	166	80,112	221	32,412	68,779
DBP-15K	JA-EN(DBP)	JA	19,814	1299	77,214	5311	84,814	216,841
		EN	19,780	1153	93,484	4791	71,948	199,917
	FR-EN(DBP)	FR	19,661	903	105,998	4136	93,680	212,609
		EN	19,993	1208	115,722	5844	88,202	259,496
DWY-100K	DBP-WD	DBpedia	100,000	330	463,294	356	310,728	622,331
		WikiPedia	100,000	220	448,774	730	765,610	990,517
	DBP-YG	DBpedia	100,000	302	428,952	341	365,617	760,062
		YAGO3	100,000	31	502,563	33	567,597	827,671

Implementation Details: For WK31-15K and DBP-15K, we follow (Sun, 2020) to use a 5-fold cross-validation to ensure an unbiased evaluation, which is picked one fold (20%) as training data and leaved the remaining for validation (10%) and testing (70%) for each running. The embedding size is =300, *i.e.,* the embedding dimensions of entities, relations, attributes and attribute values are 300, 600, 600 and 300, respectively. Just like previous approaches (Wu, Liu, Feng, Wang, Yan et al., 2019; Wu et al., 2020), we first use Google Translate to translate all Japanese words in JA-EN (DBP) to English, and use the pre-trained English word vectors of fastText (***wiki-news-300d-1M***)[3] to generate the initial name embeddings of entities and attribute values. We generate the 50-nearest candidates for each entity as negative samples every 10 epochs during the training. The other configurations are: learning-rate = 0.01, $\beta_1 = \beta_2 = 1$, $\gamma_1 = \gamma_2 = 5$, $\theta = 0.5$. We also use a 100-patience early stopping strategy, *i.e.,* the training stop if the verification loss does not decrease in 100 consecutive epochs.

Table 3 shows the comparison results of MHNA and all comparison models on cross-lingual datasets, each result of which is the average value of the 5-fold cross-validation datasets. The bottom part of Table 3 is the performance of MHNA and its variant $\text{MHNA}_{(\text{w/o m})}$. As shown in Table 3, MHNA achieves the best performance across all metrics and datasets. Specifically, the Hits@1 of MHNA significantly outperforms 9 previous state-of-the-art approaches, exceeding by 3.30–10.54% on average.

Compared with the previous models, $\text{MHNA}_{(\text{w/o m})}$ also achieves the better results, with an average increase of 1.66–8.26% on all datasets. This ablation study shows that the variant attention mechanism does improve the representation learning, thereby improving the accuracy of entity alignment.

In addition, it is comforting that the performance difference of MHNA becomes wider when it involves attribute structures, because the attribute structure can provide more information to embed entities, especially for the sparse datasets. Compared with $\text{MHNA}_{(\text{w/o m})}$, MHNA has improved performance on sparse-datasets (V1) and dense-datasets (V2), increasing by 3.72% and 2.21% on average, respectively. This performance improvements confirm that MHNA can achieve better results when entities have more neighborhood information, *i.e.,* MHNA can well mine the neighborhood information of entities in KGs.

However, the Hits@1 of MHNA is only about 1.72% higher than $\text{MHNA}_{(\text{w/o m})}$ on DBP-15K, although DBP-15K has significantly more attribute triples than WK31-15K. Therefore, the attribute structure information is not the more the better, but only plays a supporting role in MHNA. The supplementary effect of attribute structure is more obvious when the relation triples are relatively sparse, while it is less obvious when the relation triples are dense.

4.3. Overall performance on DWY-100K

Comparison Models: Since only a few models are evaluated on DWY-100K datasets, we compare MHNA with the following models: MultiKE (Zhang, Sun et al., 2019), RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019), NMN (Wu et al., 2020), COTSAE (Yang et al., 2020), and AliNet (Sun et al., 2020).

Implementation Details: For DWY-100K, we adopt the same train (30%)/test (70%) split as all comparison models and use the same initial name embedding of entities as NMN. To reduce the training complexity, we set embedding size $d = 100$, and learn a linear transformation to obtain the initial name embedding of entities and attribute values in 100-d: $\mathbf{X}^{e/v\_init} = \mathbf{W} \cdot \mathbf{X} + \mathbf{b}$. The learning-rate is 0.005, and other configurations are the same as above.

Table 4 shows the performance of all models on DWY-100K, in which MHNA consistently outperforms all comparison models in various scenarios. Both DBpedia and YAGO are derived from Wikipedia, resulting in 77.60% of the released equivalent entities have exactly the same names while the rest have very similar names (Liu et al., 2020). Therefore, most of Hits@1 in Table 4 reach more than 90%. $\text{MHNA}_{(\text{w/o m})}$ is the model only considers the relation structures of KGs, which is the same as RDGCN, NMN, and

[3] https://fasttext.cc/docs/en/english-vectors.html.

Table 3
Cross-validation results of all models on WK31-15K and DBP-15K datasets, which is divided into two parts. Means±$stds$ are shown. Other results of comparison models are produced using their source code. **Bold** numbers are the best results, and underline numbers are the second-ranked results. "*Improv. best*" represents the increase compared with the best performance of comparison models.

Datasets	EN-DE(V1)			EN-DE(V2)			EN-FR(V1)		
Models	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR
MTransE (Chen et al., 2017)	[a]30.7 ± .70	[a]51.8 ± .40	[a].407 ± .006	[a]19.3 ± 1.6	[a]35.2 ± 2.3	[a].274 ± .018	[a]24.7±.60	[a]46.7 ± .90	[a].351 ± .007
IPTransE (Zhu et al., 2017)	[a]35.0 ± .90	[a]51.5 ± 1.2	[a].430 ± .110	[a]47.6 ± 1.2	[a]67.8 ± 1.1	[a].571 ± .010	[a]16.9 ± 1.3	[a]32.0 ± 2.5	[a].243 ± .019
JAPE (Sun et al., 2017)	[a]28.8 ± 1.6	[a]51.2 ± 1.8	[a].394 ± .016	[a]16.7 ± 1.1	[a]32.9 ± 1.5	[a].250 ± .013	[a]26.2 ± .60	[a]49.7 ± 1.0	[a].372 ± .007
BootEA (Sun et al., 2018)	[a]67.5 ± .40	[a]82.0 ± .40	[a].740 ± .004	[a]83.3 ± 1.5	[a]91.2 ± .80	[a].869 ± .012	[a]50.7 ± 1.0	[a]71.8 ± 1.2	[a].603 ± .011
GCN-Align (Wang et al., 2018)	[a]48.1 ± .30	[a]67.9 ± .50	[a].571 ± .003	[a]53.4 ± .50	[a]71.7 ± .50	[a].618 ± .005	[a]33.8±.20	[a]58.9 ± .90	[a].451 ± .005
AttrE (Trisedya et al., 2019)	[a]51.7 ± 1.1	[a]68.7 ± 1.3	[a].597 ± .011	[a]65.0 ± 1.5	[a]81.6 ± .80	[a].726 ± .012	[a]48.1 ± 1.0	[a]67.1 ± .90	[a].569 ± .010
SEA (Pei et al., 2019)	[a]53.0 ± 2.7	[a]71.8 ± 2.6	[a].617 ± .025	[a]60.6 ± 2.4	[a]77.9 ± 1.8	[a].687 ± .020	[a]28.0 ± 1.5	[a]53.0 ± 2.6	[a].397 ± .019
RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019)	[a]83.0 ± .60	[a]89.5 ± .40	[a].859 ± .005	[a]83.3 ± .70	[a]89.1 ± .50	[a].860 ± .006	[a]75.5 ± .40	[a]85.4 ± .30	[a].800 ± .003
NMN (Wu et al., 2020)	85.57 ± .30	90.45 ± .11	.877 ± .002	85.18 ± .80	89.57 ± .71	.871 ± .007	85.12 ± .34	90.74 ± .55	.876 ± .004
MHNA$_{(w/o\ m)}$	91.14 ± .27	96.55 ± .32	.935 ± .003	93.44 ± .29	97.51 ± .06	.953 ± .002	88.43 ± .37	95.38 ± .45	.916 ± .004
MHNA	**94.11 ± .80**	**97.26 ± .25**	.955 ± .005	**95.72 ± .51**	**98.22 ± .42**	.969 ± .005	**92.90 ± .19**	**96.40 ± .25**	.945 ± .002
Improve-best	8.54	6.81	.078	10.54	8.65	.098	7.78	5.66	.069

Datasets	EN-FR(V2)			JA-EN(DBP)			FR-EN(DBP)		
Models	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR
MTransE (Chen et al., 2017)	[a]24.0 ± .50	[a]43.6 ± .70	[a].336 ± .005	20.41 ± .46	40.52 ± 1.1	.303 ± .007	19.74 ± .46	40.37 ± .38	.297 ± .003
IPTransE (Zhu et al., 2017)	[a]23.6 ± 1.2	[a]44.9 ± 2.1	[a].339 ± .016	27.92 ± .77	52.70 ± 1.2	.396 ± .010	31.22 ± 2.0	57.42 ± 1.9	.434 ± .019
JAPE (Sun et al., 2017)	[a]29.2 ± .90	[a]52.4 ± .60	[a].402 ± .007	23.86 ± 1.0	44.50 ± 1.8	.340 ± .012	22.98 ± .86	45.22 ± 1.4	.336 ± .012
BootEA (Sun et al., 2018)	[a]66.0 ± .60	[a]85.0 ± .50	[a].745 ± .005	52.71 ± 1.7	71.89 ± .72	.616 ± .013	57.61 ± .20	77.27 ± .26	.666 ± .002
GCN-Align (Wang et al., 2018)	[a]41.4 ± .50	[a]69.8 ± .70	[a].542 ± .005	33.71 ± 1.4	59.51 ± 1.0	.455 ± .013	32.90 ± .72	59.83 ± .84	.452 ± .009
AttrE (Trisedya et al., 2019)	[a]53.5 ± 1.5	[a]74.6 ± 1.4	[a].631 ± .014	35.96 ± .82	60.31 ± .74	.475 ± .007	40.21 ± .62	66.09 ± .34	.522 ± .004
SEA (Pei et al., 2019)	[a]36.0 ± 1.8	[a]65.1 ± 1.8	[a].494 ± .017	29.12 ± .48	55.64 ± .92	.416 ± .006	32.64 ± .37	60.28 ± 1.1	.455 ± .004
RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019)	[a]84.7 ± .60	[a]91.9 ± .40	[a].880 ± .005	81.22 ± .68	87.98 ± .83	.844 ± .007	80.88 ± .48	88.08 ± .43	.842 ± .004
NMN (Wu et al., 2020)	89.29 ± .34	94.28 ± .40	.915 ± .004	84.29 ± .76	90.47 ± .37	.870 ± .006	83.46 ± .49	90.10 ± .33	.864 ± .005
MHNA$_{(w/o\ m)}$	94.06 ± .29	97.82 ± .27	.957 ± .002	85.95 ± .75	92.83 ± .40	.891 ± .005	86.00 ± .39	93.07 ± .34	.892±.003
MHNA	**96.19 ± .27**	**98.41 ± .13**	.972 ± .002	**87.59 ± .27**	**93.42 ± .35**	.903 ± .003	**87.80 ± .57**	**93.70 ± .39**	.905 ± .005
Improve-best	6.90	4.13	.057	3.30	2.95	.033	4.34	3.60	.041

[a]Marks the results obtained from OpenEA (Sun, 2020).

Table 4
The overall performance of all models on DWY-100K datasets. The performances of RDGCN and NMN are obtained from Wu et al. (2020), and others are taken from their respective papers.

Datasets	DBP-WD			DBP-YG		
Models	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MultiKE (Zhang, Sun et al., 2019)	91.86	96.26	.935	88.03	95.32	.906
RDGCN (Wu, Liu, Feng, Wang, Yan et al., 2019)	97.90	99.10	–	94.70	97.30	–
NMN (Wu et al., 2020)	98.10	99.20	–	96.00	98.20	–
COTSAE (Yang et al., 2020)	92.68	97.86	.945	94.39	98.74	.961
AliNet (Sun et al., 2020)	69.00	90.80	.766	78.60	94.30	.841
MHNA$_{(w/o\ m)}$	99.31	99.87	.995	97.06	99.44	.979
MHNA	**99.49**	**99.87**	.996	**99.94**	**99.99**	**1.00**
Improv. best	1.39	0.67	.051	3.94	1.25	.039

AliNet. But MHNA$_{(w/o\ m)}$ shows better performance, demonstrating the effectiveness of our variant attention mechanism. Compared with MultiKE and COTSAE which consider both relation structures and attribute structures, MHNA achieves the better results. This result further supports our motivation to focus on the heterogeneous attention aggregation of the attribute structures. Since the size of DWY-100K is several times that of WK31-15K and DBP-15K, this experiments demonstrates that our model has good scalability and superiority in the larger real-world and monolingual KGs.

4.4. Analysis

More Detailed Evaluation Results. To better evaluate the effectiveness of MHNA, we test the Hits@[1, 3, 5, 10, 50, 100] on FR-EN (DBP). As Fig. 4 shown, MHNA and MHNA$_{(w/o\ m)}$ basically outperform other models in all evaluation metrics. Especially when $k \leq 10$, MHNA and MHNA$_{(w/o\ m)}$ are significantly better than the other two models. These results prove the effectiveness of the variant attention mechanism in identifying the similarity of heterogeneous neighbors of entities.

Sensitivity to Size of Training Datasets. To further analyze the effectiveness of alignment strategy of MHNA with different sizes of training dataset, we take EN-FR (V2) dataset as an example for a further evaluation. We vary the proportion of training dataset from 5% to 30% with step 5%, while the validation dataset remains at 10%. Fig. 5 illustrates the Hits@[1, 5] performance with different proportions of training dataset. Unsurprisingly, the performance of all models on EN-FR (V2) improved as the proportion

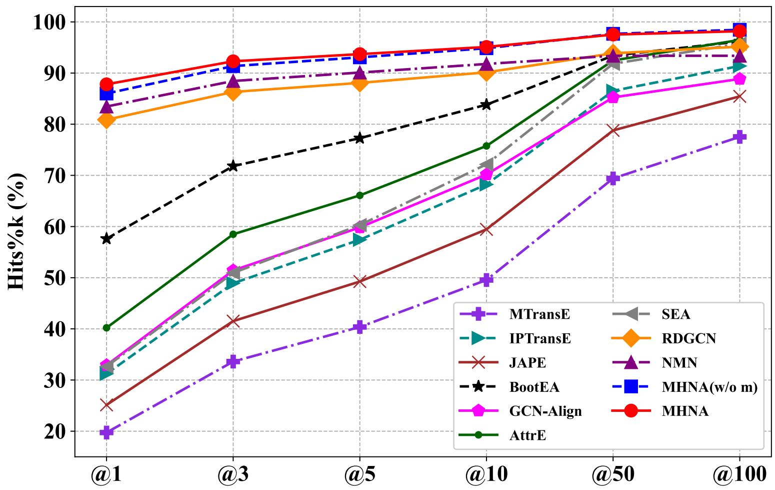

Fig. 4. The Hits@k performance of MHNA and all comparison models on FR-EN (DBP). The x-axis denotes top-k candidates, while y-axis denotes Hits@k.

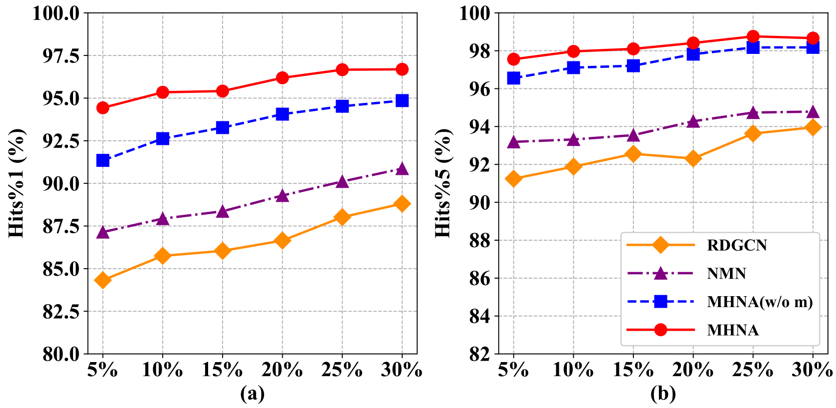

Fig. 5. The Hits@[1, 5] performance with different proportions of training dataset on EN-FR (V2). The y-axis of (a) is Hits@1 and the y-axis of (b) is Hits@5, while the x-axes are the proportions of training dataset.

increased, because more training data can provide more information to the align alignment. Obviously, MHNA maintains consistent performance and significantly outperforms the other two models. Particularly MHNA has strong robustness, because it still achieves satisfactory results with less training data. For example, the Hits@1 of RDGCN, NNN, $\text{MHNA}_{(\text{w/o m})}$ and MHNA are 84.32%, 87.15%, 91.35% , and 94.4% respectively, when only 5% of training dataset are used. Compared with other models and $\text{MHNA}_{(\text{w/o m})}$, MHNA consistently shows better results, indicating that our learning strategy of fusing multi-heterogeneous information can lead to more accurate alignment.

5. Conclusion and future work

We find that aligned entities have not only similar relation structures but also similar attribute structures. Therefore, we propose a Multi-Heterogeneous Neighborhood-Aware model (MHNA) in this paper. MHNA on the one hand takes into account the heterogeneity of relations and attributes in KGs, and on the other hand considers the similarity of aligned entities based on their names and their heterogeneous neighbors in terms of relations, attributes and attribute values. Moreover, we design a variant attention mechanism for further mining the heterogeneity of entity neighbors. Compared with the state-of-the-art approaches, MHNA achieves the best alignment results in various scenarios, and is 1.39–10.54% in Hits@1 higher than the best-performing baselines. Our approach also shows high performance on fewer pre-aligned seeds.

In the future, we will consider more information about entities, such as entity descriptions and topology features. We also are more interested in mining the complex semantics of KGs and try to design more efficient heterogeneous graph neural networks for entity alignment.

CRediT authorship contribution statement

Weishan Cai: Conceptualization, Methodology, Writing – original draft. **Yizhao Wang:** Data curation, Writing – original draft. **Shun Mao:** Data curation, Writing – original draft. **Jieyu Zhan:** Writing – review & editing. **Yuncheng Jiang:** Methodology, Supervision, Writing – review & editing.

Acknowledgments

The authors would like to thank the anonymous referees for their valuable comments as well as helpful suggestions from Professors Jim Jansen and Paolo Rosso which greatly improved the exposition of the paper. The work described in this paper is funded by the National Natural Science Foundation of China under Grant Nos. 61772210 and U1911201, Guangdong Province Universities Pearl River Scholar Funded Scheme (2018) and the Project of Science and Technology in Guangzhou in China under Grant Nos. 201807010043 and 202007040006.

References

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (pp. 1247–1250). New York, NY, USA: ACM.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 2787–2795). Lake Tahoe, USA: MIT Press.

Cao, Y., Liu, Z., Li, C., Liu, Z., Li, J., & Chua, T.-S. (2019). Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 1452–1461). Florence, Italy: ACL.

Chen, L., Gu, W., Tian, X., & Chen, G. (2019). AHAB: Aligning heterogeneous knowledge bases via iterative blocking. *Information Processing & Management*, *56*(1), 1–13.

Chen, X., Jia, S., & Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, *141*, Article 112948.

Chen, M., Tian, Y., Chang, K.-W., Skiena, S., & Zaniolo, C. (2018). Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 3998–4004). Stockholm, Sweden: Morgan Kaufmann.

Chen, M., Tian, Y., Yang, M., & Zaniolo, C. (2017). Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1511–1517). Melbourne, Australia: Morgan Kaufmann.

Chen, B., Zhang, J., Tang, X., Chen, H., & Li, C. (2020). JarKA: Modeling attribute interactions for cross-lingual knowledge alignment. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Vol. 12084* (pp. 845–856). Singapore, Singapore: Springer.

Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., et al. (2019). Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 2478–2486). Anchorage, AK, USA: ACM.

Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*,

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., et al. (2015). DBpedia – A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, *6*(2), 167–195.

Li, C., Cao, Y., Hou, L., Shi, J., Li, J., & Chua, T.-S. (2019). 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)* (pp. 2723–2732). Hong Kong, China: ACL.

Li, A., Qin, Z., Liu, R., Yang, Y., & Li, D. (2019). Spam review detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)* (pp. 2703–2711). Beijing, China: ACM.

Linmei, H., Yang, T., Shi, C., Ji, H., & Li, X. (2019). Heterogeneous graph attention networks for semi-supervised short text classification. 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)* (pp. 4820–4829). Hong Kong, China: ACL.

Liu, Z., Cao, Y., Pan, L., Li, J., Liu, Z., & Chua, T.-S. (2020). Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6355–6364). Online: ACL.

Liu, Z., Chen, C., Yang, X., Zhou, J., Li, X., & Song, L. (2018). Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)* (pp. 2077–2085). Torino, Italy: ACM.

Lu, W., Wang, P., Ma, X., Xu, W., & Chen, C. (2020). Enrich cross-lingual entity links for inline wikis via multi-modal semantic matching. *Information Processing & Management*, *57*(5), Article 102271.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS)* (pp. 3111–3119). Lake Tahoe, USA: MIT Press.

Pei, S., Yu, L., Hoehndorf, R., & Zhang, X. (2019). Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Proceedings of the 2019 World Wide Web Conference (WWW)* (pp. 3130–3136). San Francisco, CA, USA: ACM.

Qiao, C., & Hu, X. (2020). A neural knowledge graph evaluator: Combining structural and semantic evidence of knowledge graphs for predicting supportive knowledge in scientific QA. *Information Processing & Management*, *57*(6), Article 102309.

Rahimi, A., Cohn, T., & Baldwin, T. (2018). Semi-supervised user geolocation via graph convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 2009–2019). Melbourne, Australia: ACL.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)* (pp. 593–607). Iraclion, Greece: Springer.

Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. (2018). Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*(2), 357–370.

Shin, S., Jin, X., Jung, J., & Lee, K.-H. (2019). Predicate constraints based question answering over knowledge graph. *Information Processing & Management*, *56*(3), 445–462.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). YAGO: A core of semantic knowledge unifying WordNet and wikipedia. In *Proceedings of the 16th International Conference on World Wide Web (WWW)* (pp. 697–706). New York, NY, USA: ACM.

Sun, Z. (2020). A benchmarking study of embedding-based entity alignment for knowledge graphs. In *Proceedings of the VLDB Endowment (VLDB), Vol. 13* (pp. 2326–2340). Tokyo, Japan: Morgan Kaufmann.

Sun, Z., Hu, W., & Li, C. (2017). Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of the International Semantic Web Conference (ISWC)* (pp. 628–644). Vienna, Austria: IEEE.

Sun, Z., Hu, W., Zhang, Q., & Qu, Y. (2018). Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4396–4402). Stockholm, Sweden: Morgan Kaufmann.

Sun, Z., Wang, C., Hu, W., Chen, M., Dai, J., Zhang, W., et al. (2020). Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Vol. 34(1)* (pp. 222–229). New York, USA: AAAI.

Trisedya, B. D., Qi, J., & Zhang, R. (2019). Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence (AAAI), Vol. 33(01)* (pp. 297–304). Hawaii, USA: AAAI.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations(ICLR)*,

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., et al. (2019). Heterogeneous graph attention network. In *Proceedings of the 2019 World Wide Web Conference (WWW)* (pp. 2022–2032). San Francisco, USA: ACM.

Wang, Z., Lv, Q., Lan, X., & Zhang, Y. (2018). Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 349–357). Brussels, Belgium: ACL.

Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., & Zhao, D. (2019). Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 5278–5284). Macao, China: Morgan Kaufmann.

Wu, Y., Liu, X., Feng, Y., Wang, Z., & Zhao, D. (2019). 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)* (pp. 240–249). Hong Kong, China: ACL.

Wu, Y., Liu, X., Feng, Y., Wang, Z., & Zhao, D. (2020). Neighborhood matching network for entity alignment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 6477–6487). Online: ACL.

Xu, K., Wang, L., Yu, M., Feng, Y., Song, Y., Wang, Z., et al. (2019). Cross-lingual knowledge graph alignment via graph matching neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 3156–3161). Florence, Italy: ACL.

Xu, B., Xu, Y., Liang, J., Xie, C., Liang, B., Cui, W., et al. (2017). CN-dbpedia: A never-ending Chinese knowledge extraction system. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 428–438). Springer.

Yang, K., Liu, S., Zhao, J., Wang, Y., & Xie, B. (2020). COTSAE: CO-training of structure and attribute embeddings for entity alignment. In *Proceedings of 34th AAAI Conference on Artificial Intelligence (AAAI), Vol. 34(03)* (pp. 3025–3032). New York, USA: AAAI.

Yang, H.-W., Zou, Y., Shi, P., Lu, W., Lin, J., & Sun, X. (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)* (pp. 4430–4440). Hong Kong, China: ACL.

Ye, R., Li, X., Fang, Y., Zang, H., & Wang, M. (2019). A vectorized relational graph convolutional network for multi-relational network alignment. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4135–4141). Macao, China: Morgan Kaufmann.

Yun, S., Jeong, M., Kim, R., Kang, J., & Kim, H. J. (2019). Graph transformer networks. In *Proceedings of the 33th International Conference on Neural Information Processing Systems (NeurIPS), Vol. 32*. Vancouver, Canada: MIT Press.

Zeng, K., Li, C., Hou, L., Li, J., & Feng, L. (2021). A comprehensive survey of entity alignment for knowledge graphs. *AI Open*, *2*, 1–13.

Zeng, W., Zhao, X., Tang, J., & Lin, X. (2020). Collective entity alignment via adaptive features. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (pp. 1870–1873). Online.

Zhang, C., Song, D., Huang, C., Swami, A., & Chawla, N. V. (2019). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 793–803). Anchorage, AK, USA: ACM.

Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., & Qu, Y. (2019). Multi-view knowledge graph embedding for entity alignment. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 5429–5435). Macao, China: Morgan Kaufmann.

Zhao, Y., Zhang, J., Zhou, Y., & Zong, C. (2020). Knowledge graphs enhanced neural machine translation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence* (pp. 4039–4045). Yokohama, Japan: Morgan Kaufmann.

Zhu, Q., Wei, H., Sisman, B., Zheng, D., Faloutsos, C., Dong, X. L., et al. (2020). Collective multi-type entity alignment between knowledge graphs. In *Proceedings of the 2020 World Wide Web Conference (WWW)* (pp. 2241–2252). Taipei, Taiwan: ACM.

Zhu, H., Xie, R., Liu, Z., & Sun, M. (2017). Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4258–4264). Melbourne, Australia: Morgan Kaufmann.

Zhu, Q., Zhou, X., Wu, J., Tan, J., & Guo, L. (2019). Neighborhood-aware attentional representation for multilingual knowledge graphs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1943–1949). Macao, China: Morgan Kaufmann.