Dual Attention Network for Cross-lingual Entity Alignment

Jian Sun[1,2], **Yu Zhou** [1,2,3] and **Chengqing Zong**[1,2]
[1] School of Artificial Intelligence, University of Chinese Academy of Sciences,
Beijing 100049, P.R.China
[2] National Laboratory of Pattern Recognition, Institute of Automation, CAS
[3]Beijing Fanyu Technology Co., Ltd, Beijing, China
{jian.sun, yzhou, cqzong}@nlpr.ia.ac.cn

Abstract

Cross-lingual Entity alignment is an essential part of building a knowledge graph, which can help integrate knowledge among different language knowledge graphs. In the real KGs, there exists an imbalance among the information in the same hierarchy of corresponding entities, which results in the heterogeneity of neighborhood structure, making this task challenging. To tackle this problem, we propose a dual attention network for cross-lingual entity alignment (DAEA). Specifically, our dual attention consists of relation-aware graph attention and hierarchical attention. The relation-aware graph attention aims at selectively aggregating multi-hierarchy neighborhood information to alleviate the difference of heterogeneity among counterpart entities. The hierarchical attention adaptively aggregates the low-hierarchy and the high-hierarchy information, which is beneficial to balance the neighborhood information of counterpart entities and distinguish non-counterpart entities with similar structures. Finally, we treat cross-lingual entity alignment as a process of linking prediction. Experimental results on three real-world cross-lingual entity alignment datasets have shown the effectiveness of DAEA.

1 Introduction

In recent years, many large-scale knowledge graphs (Bordes et al., 2013; Mahdisoltani et al., 2015; Auer et al., 2007) are built to represent and organize the explosive information over the Internet. They have been widely used in many fields, such as dialogue system (Wang et al., 2019), machine translation (Zhao et al., 2020) and medicine (Yan et al., 2020). However, the KGs are usually incomplete, so it is necessary to manually track emerging concepts and dynamically update the knowledge bases, making the whole process quite expensive. Fortunately, KGs of different languages are often complementary, which means that many components can be shared. To integrate this complementary knowledge, researches begin to pay attention to cross-lingual entity alignment.

Cross-lingual entity alignment aims at finding entities with the same semantics in KGs of different languages. Various methods have been explored for cross-lingual entity alignment. Traditional approaches rely on machine translation or feature engineering (Chen et al., 2013; Mahdisoltani et al., 2015; Otani et al., 2018; Feng et al., 2016). The effectiveness of these methods depends largely on the quality of the translation and the nature of the definition. Recently, many embedding approaches based on graph neural network (GNN) are proposed for cross-lingual entity alignment (Wang et al., 2018; Li et al., 2019; Sun et al., 2020; Wu et al., 2019a). These methods first represent the entities and relations in low dimensional spaces and then utilize the powerful encoding ability of graph neural network to learn vector representations for entity or relation. Finally, a mapping function is used to align the entities from the source knowledge graph to the target one.

However, due to the incompleteness of knowledge graphs and the diversity of knowledge, the structure of KGs in different languages is usually quite different. In the entity alignment task, this difference is mainly reflected in two aspects: the non-isomorphism among the neighborhood structures of

Proceedings of the 28th International Conference on Computational Linguistics, pages 3190–3201
Barcelona, Spain (Online), December 8-13, 2020

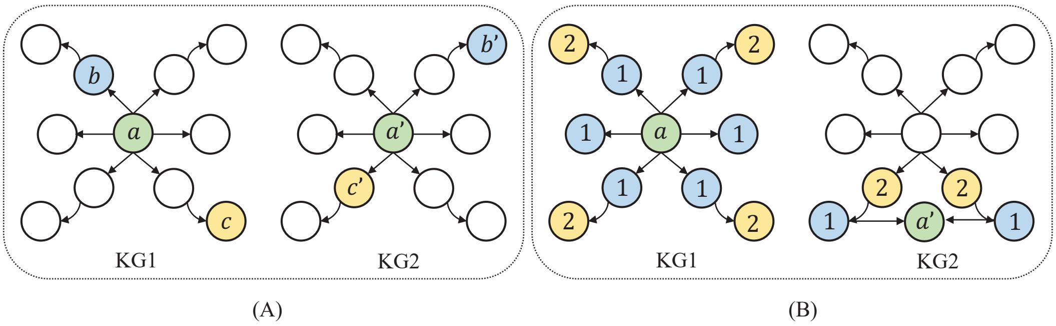


Figure 1: An illustration of the heterogeneous structures of KGs.

counterpart entities and information imbalance among the same hierarchy of counterpart entities. The non-isomorphism among the neighborhood structures refers to the neighborhood of the two entities is inconsistent with each other, especially containing different sets of neighboring entities. Figure 1 (A) gives a toy example. In (A), entity pairs (a, a'), (b, b') and (c, c') denote three pairs of pre-aligned entities. In KG1, entity b is in the one-hop neighbor set of entity a. However, in KG2, entity b' is in two-hop neighbors set of entity a'. It suggests that the semantically-related entities can appear at different neighborhood hierarchy, which can easily lead to encoding the same pair of entities into different representations when encoding entities with GNN. The information imbalance among the hierarchy of entities means that the degrees are different at the same hierarchy of the corresponding entity. Figure 1 (B) gives a toy example of the degree imbalance. In KG1, the one-hop neighborhood of entity a contains five entities, and the two-hop neighborhood contains four entities. However, in KG2, the one-hop neighborhood and the two-hop neighborhood of entity a' each contain only two entities. The reason for this phenomenon may be that the two KGs focus on different contents when they are constructed or due to the lack of some knowledge. It may lead to some problems when encoding entities with GNN. For example, many entities at the center of the knowledge graphs have rich neighbor information, like a, they can integrate the information of the whole graph after limited updates, which will lead to little difference in the final representation of these central entities. In entity alignment, this will make it easier for entities with similar neighborhood structures to match together (Pei et al., 2019).

To address the issues above, in this paper, we propose a dual attention network for entity alignment (DAEA). The dual attention consists of the relation-aware graph attention (R-GAT) and hierarchical attention. In R-GAT, we extend the graph attention network (Veličković et al., 2018) by modeling relation in graph attention network and incorporate the translational assumption into the self-attention mechanism that we can model the relationship among head, tail, and relation in the graph attention network. By selectively aggregating multi-level neighborhood information, R-GAT can alleviate the non-isomorphic difference between corresponding entities' neighborhood structures. Inspired by jumping knowledge network (Xu et al., 2018), in the hierarchical attention module, we use an LSTM network to learn attention coefficients. It identifies the most useful neighborhood ranges for each entity and adaptively fuses the information of different hierarchy to balance the neighborhood information of peer entities and distinguish entities with similar structures. In entity alignment methods, we design a new entity alignment method, which uses link prediction directly to accomplish entity alignment. We perform thorough experiments with detailed ablation studies and analyses on three entity alignment datasets, demonstrating the effectiveness of DAEA.

2 Related Work

2.1 Knowledge Embedding

Various embedding based models have been developed to knowledge representation learning. The models project the entities and relations into a low-dimensional vector space and define a score function to measure the plausibility of each triple. The total likelihood of triples is finally maximized to learn the embeddings of entities and relations. TransE (Bordes et al., 2013) regarded each relation as a translation vector between the head entity and the tail entity. Based on the idea of TransE, more advanced models such as TransH (Wang et al., 2014), RotatE (Sun et al., 2019)have been proposed. There are other embeddings based models like RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2014) and ComplEx (Trouillon et al., 2016), which optimized a bilinear product scoring function between vector embedding for each head and tail entity and a full rank matrix for each relation.

2.2 Graph Neural Networks

Graph neural networks (GNNs), a connectionist model that captures the dependence of graphs via message passing between the nodes of graphs, are useful tools on non-Euclidean structures. There are various methods proposed trying to improve the model's capability. Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) , an extension of GNNs, are neural networks operating on unlabeled graphs and inducing features of nodes based on the structures of their neighborhoods. The attention mechanism has also been successfully used in GNNs. Graph Attention network (GAT) (Veličković et al., 2018) incorporates the attention mechanism into the propagation step and computes the hidden states of each node by attending over its neighbors, following a self-attention strategy. Furthermore, R-GCN (Schlichtkrull et al., 2018) has recently been proposed to model relational data and have been successfully exploited in link prediction and entity classification.

2.3 Entity Alignment

The earliest entity alignment approaches rely on machine translation or feature engineering. They are time-consuming, labor expensive and poor in adaptability, and scalability. Recently, embedding based methods have been applied to entity alignment. MTransE (Chen et al., 2017), IPTransE (Zhu et al., 2017), AlignE (Sun et al., 2018), JAPE (Sun et al., 2017) and NAEA (Zhu et al., 2019), rely on TransE model to learn entity embedding, define some kinds of transformation and learn a linear mapping or minimize the distance between the embedding of pre-aligned entities. Some works use GNNs for entity alignment (Wang et al., 2018; Li et al., 2019; Sun et al., 2020; Wu et al., 2019a; Wu et al., 2019b; Zhu et al., 2019). GCN-Align (Wang et al., 2018) takes advantage of GCN to propagate information from neighbors, and align entity embeddings enhanced by structural knowledge. It only considers the connectivity between entities and ignores the relation features in KGs. KECG (Li et al., 2019) introduces GAT with projection constraint to robustly encode graphs, and employ the nearest neighbor sampling strategy for KG representation learning towards one-to-one mapping. AliNet (Sun et al., 2020) introduces distant neighbors to expand the overlap between their neighborhood structures. It employs an attention mechanism to highlight helpful distant neighbors and reduce noises. AliNet considers the phenomenon of non-isomorphic neighborhood structures. However, they all ignore information imbalance among the same hierarchy of entities.

3 Problem Formulation

Formally, knowledge graphs consist of a set of entities and a set of relations. Knowledge facts are stored as a collection of triples $(head, relation, tail)$. We define a KG as $G = (E, R, T)$, where E and R represent the sets of entities and relations respectively, T is the set of relational triples. Let $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$ be two different KGs. $P = \{(e_{i1}, e_{i2})|e_{i1} \in E_1, e_{i2} \in E_2\}$ is the set of pre-aligned entity pairs between G_1 and G_2. The task of cross-lingual entity alignment aims to find new aligned entity pairs based on the pre-aligned seeds.

4 Our Approaches

The model DAEA proposed for entity alignment combines knowledge embedding and graph neural networks. It consists of three modules: the relation-aware graph attention network module, the LSTM-based hierarchical attention module, and the knowledge embedding module. Our model framework is shown in Figure 2.

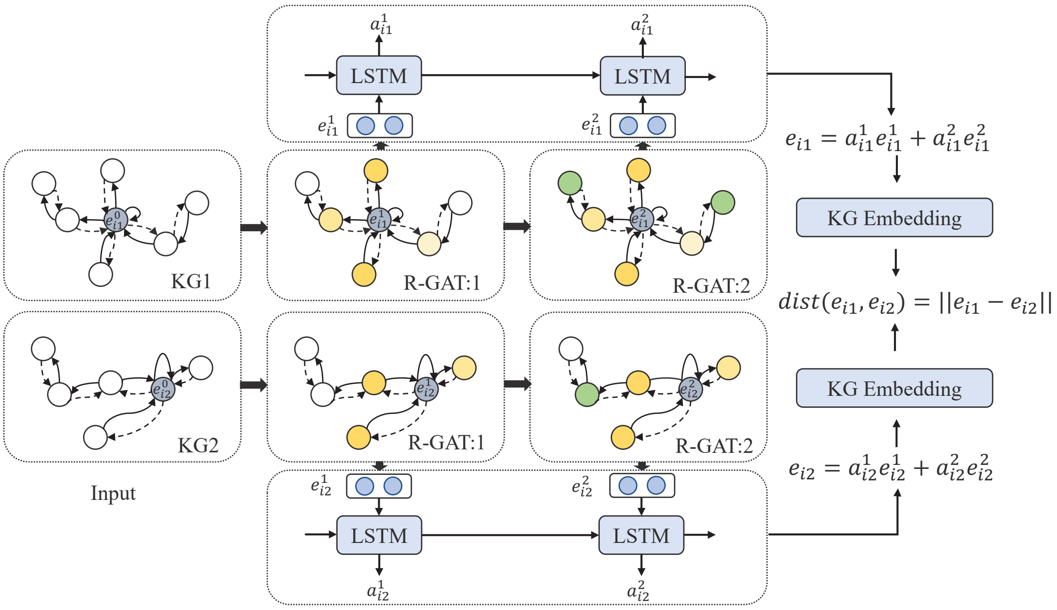


Figure 2: The framework of DAEA. (e_{i1}^m, e_{i2}^m) is pre-aligned entity pair. In e_{i1}^m, i, 1 and m represent entity i and knowledge graphs 1 and m-th update of entity representation, respectively. a_{i1}^m is the weight of the different update layers.

4.1 Relation-aware Graph Attention Module

This module is designed to mitigate the difference of non-isomorphism among counterpart entities by aggregating multi-hierarchy neighborhood information with different weights. Compared to GAT ignoring relation information, in R-GAT, we fuse relation-level information, which is vital to representation learning of entity to graph attention module. Meanwhile, to speed up the flow of information, we add inverse triples to KGs, which have been proved to be effective in knowledge graphs completion (Bordes et al., 2013; Neelakantan et al., 2015). The module R-GAT takes as input an tail entity e_t with its neighborhood set $N_{e_t} = \{(e_1, r_1), (e_2, r_2), ..., (e_n, r_n)\}$, where $e_t \in \{E_1, E_2\}$. In NAEA, the author only considers a fixed number of entities, which can lead to information loss. Here, just like the original GAT, we consider all neighbor entities.

In R-GAT, we first calculate attention coefficients between entity e_t and neighborhood information. We use two weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ to transform the entity and relation, respectively. Here we merge the translational assumption from TransE into attention coefficients. For each triple, the self-attention coefficients can be calculated as follows:

$$c_{ij}^m = \mathbf{a}^T[\mathbf{W}_1^m(\mathbf{e}_j^m + \mathbf{r}_{ij}^m), \mathbf{W}_2^m\mathbf{e}_i^m] \tag{1}$$

where $\mathbf{a}^t \in R^{2d}$ is the learnable parameter, $\mathbf{e}_i^m$ and $\mathbf{e}_j^m$ are the hidden states of tail entity e_i and head entity e_j, $\mathbf{r}_{ij}^m$ is the embedding of r_{ij} which is between e_i and e_j. m is the number of layers. [,] indicate the concatenation operation. Note that in Figure 2, the neighborhood set of entity e_{i1} contains the self-loop, so we need to add a relation vector to relation set to represent self-loop relation. Then we calculate the weight by applying a non-linear activation function.

$$w_{ij} = \frac{\exp(\mathrm{LeakyReLU}(c_{ij}^m))}{\sum_{e_k \in N_{e_i} \cup \{e_i\}} \exp(\mathrm{LeakyReLU}(c_{ik}^m))} \tag{2}$$

where LeakyReLU is a nonlinear function (Veličković et al., 2018). To obtain the final output of the entity, we calculate a linear combination with different weights. Here we employ multi-head attention in the same way as GAT.

$$\mathbf{e}_i^{m+1} = \mathrm{ReLU}(\frac{1}{K}\sum_{k=1}^{\mathrm{K}}\sum_{j=1}^{n} w_{ij}^k(\mathbf{e}_j^m + \mathbf{r}_{ij}^m)) \tag{3}$$

where K is the number of head, n is the number of neighbors of entity e_i, w_{ij}^k are the attention coefficients computed by the k-th attention head. In the first iteration, $\mathbf{e}_i^1$ aggregates the information of $\mathbf{e}_j^0$. Meanwhile, $\mathbf{e}_j^1$ also aggregates the information of $\mathbf{e}_k^0$, where $\mathbf{e}_k$ is the neighborhood head entity of $\mathbf{e}_j$. In the second iteration, $\mathbf{e}_i^2$ aggregates the information of $\mathbf{e}_j^1$, here, $\mathbf{e}_j^1$ already contains the information of $\mathbf{e}_k^0$. Therefore, $\mathbf{e}_i^2$ contains two-hop information, $\mathbf{e}_j^1$ and $\mathbf{e}_k^0$. After m iterations, entity e can aggregate all neighborhood information between one-hop and m-hop.

Compared with AliNet which explicitly fuses the information of multiple hop neighborhood, R-GAT can merge the information of i-hop selectively into the current entity representation during the update of i-th layer, without explicit modeling, which is an essential feature of the graph neural networks.

4.2 LSTM-based Hierarchical Attention Module

Given the relation-aware graph attention module, now we discuss the LSTM-based hierarchical attention module. Like the example in Figure 1 (B), dense entities tend to be more consistently represented. To learn the low-level and the high-level information better adaptively and balance the neighborhood information of counterpart entities, we design the LSTM-based hierarchical attention module.

The input of the module is $(\mathbf{e}_i^1, \mathbf{e}_i^2, ..., \mathbf{e}_i^m)$ which is the output for each iteration in R-GAT. The LSTM-based hierarchical attention module can be described as follows:

$$\mathbf{h}_i^j, \mathbf{c}_i^j = \mathrm{LSTM}(\mathbf{e}_i^j, [\mathbf{h}_i^{j-1}, \mathbf{c}_i^{j-1}]) \tag{4}$$

$$a_i^j = \mathrm{softmax}(\mathbf{W}_3 \mathbf{h}_i^j) \tag{5}$$

$$\mathbf{e}_i^{out} = \sum_{j=1}^{m} a_i^j \mathbf{e}_i^j \tag{6}$$

where $\mathbf{h}_i^{j-1}$ and $\mathbf{c}_i^{j-1}$ is the hidden state and cell state, respectively. $\mathbf{W}_3 \in R^{d \times 1}$ are parameters to be learned. The LSTM-based hierarchical attention can learn the weight of different levels of entities, which can indirectly balance the information of the same level of counterpart entities through supervised learning.

4.3 Knowledge Embedding Module

Previous works have shown that knowledge embedding, which models inner-graph relationships, making entities more distinguishable, is effective for entity alignment (Li et al., 2019). Another reason to leverage knowledge embedding here is that it is necessary for the novel entity alignment method. It will be described later. Here we use TransE, which regards relation as a translation vector between entities, as our knowledge embedding module.

Formally, for a triple (e_h, r, e_t), we define the score function $f(e_h, r, e_t) = ||\mathbf{e}_h + r - \mathbf{e}_t||$ to measure the possibility of the triple. we use a margin-based ranking loss function as the training the knowledge embedding model. the loss function is defined as follows:

$$L_k = \sum_{(e_h,r,e_t)\in T} \sum_{(e_h^{'},r,e_t^{'})\in T'} [f(e_h, r, e_t) + \gamma_1 - f(e_h^{'}, r, e_t^{'})]_+ \tag{7}$$

where $[\cdot]_+ = max(0, \cdot)$, T is the set of the total relations in two KGs and $T^{'}$ is the set of negative samples generated by corrupting T. γ_1 is a positive margin hyper-parameter separating positive and negative triplets.

4.4 Training and Inference

We use a margin-based loss function as the training objective to let the embedding of aligned entities have a very small distance while those of unaligned entities have a large distance:

$$L_c = \sum_{(e_{i1},e_{i2})\in P} \sum_{\left(e_{i1}',e_{i2}'\right)\in P'} \left[\mathrm{dist}\left(e_{i1}, e_{i2}\right) + \gamma_2 - \mathrm{dist}\left(e_{i1}', e_{i2}'\right)\right]_+ \tag{8}$$

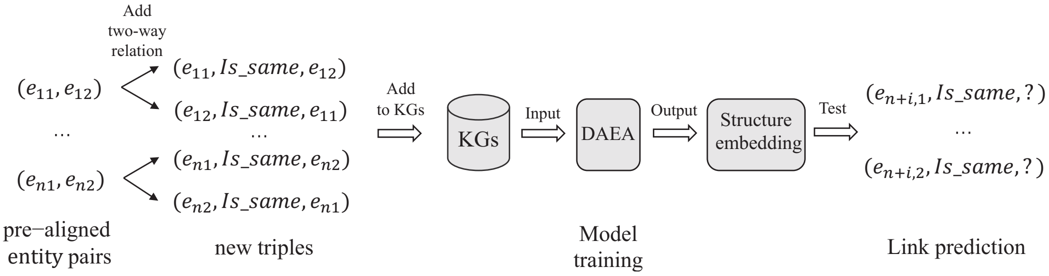


Figure 3: The framework of the completion-based method.

where $\mathrm{dist}(e_{i1}, e_{i2}) = ||\mathbf{e}_{i1} - \mathbf{e}_{i2}||_2$ is the L_2 distance, γ_2 is a positive margin hyper-parameter. P is the set of pre-aligned entity pair (e_{i1}, e_{i2}), and $P^{'}$ is the set of negative samples generated by nearest neighbor negative sampling.

We define the loss function of DAEA corresponding to the above two parts (Li et al., 2019; Sun et al., 2020):

$$L = L_c + L_k \tag{9}$$

It is worth noting that we can set different weights for L_c and L_k. Here, we set the same weight for them to treat both kinds of loss functions equally in experiments (Li et al., 2019).

In inference, we adopt two methods to align entity in this paper, distance-based method and completion-based method.

4.4.1 Distance-based Method

In distance-based methods, we can align the entity based on the nearest neighbor search among entity embedding by simply calculating the distance between two entities between different KGs.

$$dist(e_1, e_2) = ||\mathbf{e}_1 - \mathbf{e}_2||_2 \tag{10}$$

4.4.2 Completion-based Method

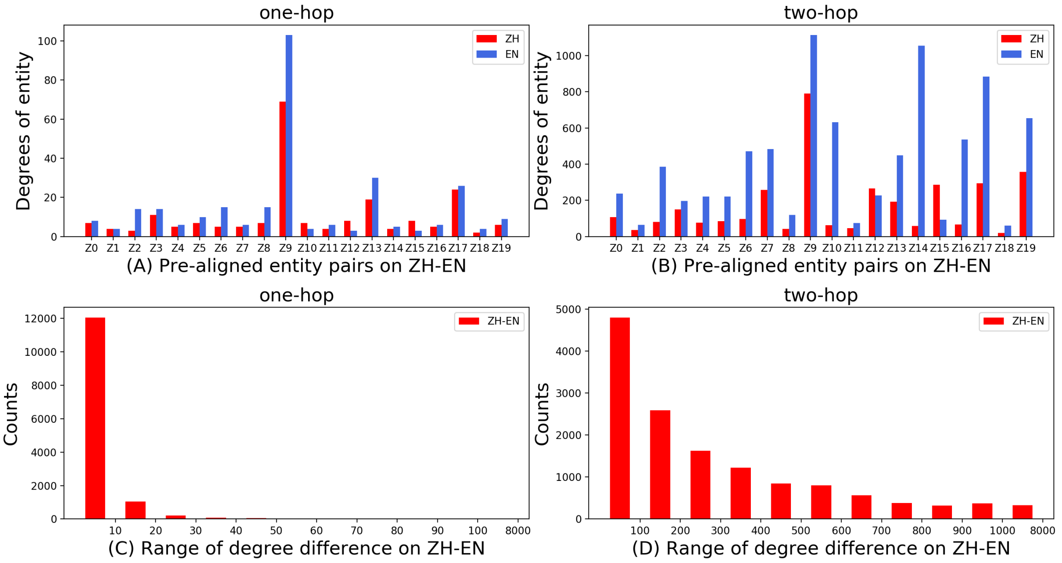
In this part, we design a novel entity alignment method based on link prediction following the knowledge graph completion task. Before training the model, we need to add a relation type to KGs in order to establish links between pre-trained entity pairs. For each entity pair $(e_{i1}, e_{i2}) \in P$, we add the symmetry relation 'Is_same' between entity e_{i1} and e_{i2} , so we obtain triples $(e_{i1}, Is\_same, e_{i2})$ and $(e_{i2}, Is\_same, e_{i1})$. Then we add them to KGs for training. In the testing, we align the entities directly according to the link prediction method. The framework of the completion-based method is shown in Figure 3.

5 Experiments and Results Analysis

5.1 Datasets

Dataset		#Ent	#Rel	#Tri
$DBP15K_{ZH-EN}$	ZH	66,469	2,830	153,929
	EN	98,125	2,317	237,674
$DBP15K_{JA-EN}$	JA	65,744	2,043	164,373
	EN	95,680	2,096	233,319
$DBP15K_{FR-EN}$	FR	66,858	1,379	192,191
	EN	105,889	2,209	278,590

Table 1: Statistics of datasets used in experiments. 'Ent' denotes the number of entities and 'Rel' denotes the number of all relations. 'Tri' denotes the number of triples.



(A) Pre-aligned entity pairs on ZH-EN

(B) Pre-aligned entity pairs on ZH-EN

(C) Range of degree difference on ZH-EN

(D) Range of degree difference on ZH-EN

Figure 4: Statistics of degrees on $DBP15K_{ZH-EN}$.

We evaluate DAEA on the $DBP15K$ datasets. $DBP15K$ contains three cross-lingual datasets built from $DBpedia$, denoted by $DBP15K_{ZH-EN}$(Chinese-to-English), $DBP15K_{JA-EN}$ (Japanese-to-English) and $DBP15K_{FR-EN}$ (French-to-English). Each dataset contains $15,000$ reference alignment links with popular entities from English to Chinese, Japanese and French respectively. The statistics of datasets are shown in Table 1.

Figure 4 shows the statistics of degrees of 20 randomly selected pre-aligned entities at different levels from $DBP15K_{ZH-EN}$. Here, the degree of an entity refers to the number of all triplets associated with that entity. From Figure 4(A) and (B), we can see that the degree imbalance of one-hop neighborhood exists only in a few entity pairs. However, the imbalance is obvious in the two-hop neighborhood set. For example, in entity pair Z14 , the degree difference between two-hop neighborhood entities is more than 1000. Figure 4 (D) shows that the distribution of degree difference in the two-hop neighborhood roughly follows the long tail distribution.

5.2 Experimental Setting

In DAEA, we set the dimension of entity (relation) embeddings to 128 and the learning rate to 0.001, For each positive triplet, we select 25 negative triples for graph model training and 2 negative triples for knowledge embedding model training. the margin γ_1 and γ_2 both set to 3 (Li et al., 2019). In R-GAT, we use a 2-layer R-GAT as the encoder. We used 30% of the gold standards as seed alignment while left the remaining as testing data. We compare DAEA with the following entity alignment methods: MTransE (Chen et al., 2017), IPTransE (Zhu et al., 2017), JAPE (Sun et al., 2017), AlignE (Sun et al., 2018), GCN-Align (Wang et al., 2018), SEA (Pei et al., 2019), MuGCN (Cao et al., 2019), KECG (Li et al., 2019) and AliNet (Sun et al., 2020). Some models, like GMNN (Xu et al., 2019) and RDGCN (Wu et al., 2019a), merge the literal information of the entities into their representations. Since our model relies only on structural information, we do not take these models into comparison. In order to verify the different alignment methods, we also select two KG embedding models TransH and RotatE which are usually evaluated on the task of link prediction as baselines. We report the Hits@1, Hits@10 and MRR results to evaluate entity alignment performance. Meanwhile, we conducted three ablation experiments: DAEA (w/o LSTM & rel.) means that does not LSTM-based attention and relation-aware. DAEA (w/o rel.) and (DAEA w/o LSTM) represent that does not relation-aware and LSTM-based attention, respectively.

Methods	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
MtansE (Chen et al., 2017)	30.83	61.41	0.364	27.86	57.45	0.349	24.41	55.55	0.335
IPTtransE (Zhu et al., 2017)	40.60	73.50	0.516	36.70	69.30	0.474	33.30	68.50	0.451
JAPE (Sun et al., 2017)	41.18	74.46	0.490	36.25	68.50	0.476	32.39	66.68	0.430
AlignE (Sun et al., 2018)	47.18	79.19	0.581	44.76	78.89	0.563	48.12	82.43	0.599
GCN-Align (Wang et al., 2018)	41.25	74.38	0.549	39.91	74.46	0.546	37.29	74.49	0.532
SEA (Pei et al., 2019)	42.40	79.60	0.548	38.5	78.30	0.518	40.00	79.70	0.533
KECG (Li et al., 2019)	47.77	83.50	0.598	48.97	84.40	0.610	48.64	85.06	0.610
MuGCN (Cao et al., 2019)	49.45	84.45	0.611	50.15	85.75	0.621	49.55	87.75	0.621
AliNet (Sun et al., 2020)	53.95	82.65	0.628	54.95	83.15	0.645	55.25	85.25	0.657
DAEA(w/o LSTM & rel.)	46.17	83.55	0.589	48.30	86.63	0.609	47.78	86.74	0.611
DAEA(w/o rel.)	48.40	83.31	0.604	50.40	85.36	0.621	50.61	86.96	0.630
DAEA(w/o LSTM)	53.50	87.46	0.654	54.82	88.11	0.664	53.49	89.24	0.660
DAEA	**56.76**	**88.30**	**0.677**	**57.59**	**89.23**	**0.683**	**58.04**	**91.16**	**0.695**

Table 2: The overall alignment performance for all models on the DBP15K datasets.

5.3 Results and Discussion

5.3.1 Main Results

We present the entity alignment results in Table 2. Note that we adopt the completion-based method in DAEA. We can see that DAEA is significantly more effective than the baseline models. On $DBP15K_{FR-EN}$, DAEA improves Hit@1 by a margin of 0.171, Hit@10 by a margin of 0.039 and MRR by a margin of 0.119 against MuGNN. For $DBP15K_{ZH-EN}$, comparing KECG, our DAEA improves Hit@1 by a margin of 0.18, Hit@10 by a margin of 0.057 and MRR by a margin of 0.132. Comparing AliNet, DAEA achieves a gain of 0.048 by Hit@1, 0.073 by Hit@10 and 0.059 by MRR on $DBP15K_{JA-EN}$. These results can demonstrate the superiority of the DAEA.

It is not surprising that DAEA outperforms the most basic GNN-based alignment model, i.e. KECG, AliNet, and GCN-Align. By performing relation-aware graph attention over entity's neighbors, R-GAT not only extends the relation to GAT, but also integrates translation hypothesis into the self-attention mechanism, which can model the relationship among heads, tails and relations. Meanwhile, DAEA introduces an LSTM-based hierarchical attention module to identify the most useful neighborhood ranges for each entity and learn low-level information and high-level information adaptively.

5.3.2 Ablation Studies

In the ablation study, we can see that both the relation-aware graph attention and LSTM-based hierarchical attention play essential roles in our model. Among them, the module with the most significant gain is the relation-aware graph attention module. It can improve Hit@1 by a margin of 0.173 and Hit@10 by a margin of 0.06 on $DBP15_{ZH-EN}$. The LSTM-based hierarchical attention module also improves the performance of entity alignment based on the relation-aware graph attention module. The improvement of the LSTM-based attention module alone is not apparent. DAEA (w/o LSTM & rel.) is similar to KECG. Ablation studies show that the good performance of DAEA is largely attributed to the information fusion capability of R-GAT module, which incorporates the neighborhood entity and relation information of KGs with different weights for entity alignment.

5.4 Analysis

5.4.1 Effectiveness on alignment methods

Table 3 lists the results of different alignment methods on $DBP15K_{ZH-EN}$. Here we ignore the graph encoding part to quickly compare the performance of alignment methods. In TransE, the completion-based method is significantly more effective than the distance method. Note that the score function is $f_c = ||\mathbf{e}_{i1} + \mathbf{Is\_same} - \mathbf{e}_{i2}||$ in the completion method. In the distance method, the score function is $f_d = ||\mathbf{e}_{i1} - \mathbf{e}_{i2}||$ for entity pair (e_{i1}, e_{i2}). The embedding of Is_same is immutable during alignment. It indicates that the improvement of the completion-based module comes from

$DBP15K_{ZH-EN}$	TransE		RotatE		TransH		
	comp.	dist.	comp.	dist.	comp.	dist.	dist.(Is_same)
Hit@1	49.04	48.32	33.39	35.14	40.12	38.67	38.74
Hit@5	71.05	70.16	53.73	60.26	64.00	61.21	61.11
Hit@10	77.74	77.23	59.88	65.98	72.03	69.25	69.30
MRR	0.588	0.581	0.412	0.441	0.508	0.492	0.493

Table 3: Comparison on different alignment methods. 'comp.' denotes the completion-based method and 'dist.' denotes the distance-based method

Methods	DBP_{ZH-EN}			DBP_{JA-EN}			DBP_{FR-EN}		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
DAEA(concat)	41.50	77.4	0.530	42.91	78.43	0.543	43.17	80.69	0.561
DAEA(MaxPooling)	53.39	87.83	0.652	54.71	87.78	0.661	55.21	89.94	0.671
DAEA(mean)	**56.84**	88.16	**0.678**	57.31	88.95	0.682	**58.18**	91.04	**0.696**
DAEA(LSTM)	56.76	**88.30**	0.677	**57.59**	**89.23**	**0.683**	58.04	**91.16**	0.695

Table 4: Comparison on different hierarchical aggregate methods.

added triples $(e_{i1}, Is\_same, e_{i2})$. So in TransH, we add an experiment, dist.(Is_same), where triples $(e_{i1}, Is\_same, e_{i2})$ are added to the training data in the distance-based alignment method. After eliminating differences in data, we can see that the completion-based alignment method works better than the distance-based alignment.

However, in RotatE, the performance of the completion-based method is poorer than that of the distance method. We also utilize KGs embedding models ComplEx to alignment entity. However, the completion method based on ComplEx also has poor performance. It indicates that the scoring function affects the performance of the completion-based method. The scoring function based on the translation hypothesis contains distance metric, and it can also model the spatial transformation of different language entity pairs through the connection of the Is_same. In RotatE and ComplEx, distance supervision is not apparent, which requires a large amount of training data. However, in entity alignment, the prior alignment usually accounts for only a small proportion, which would prevent these approaches from learning accurate embeddings for entities. In general, the advantage of the distance-based method is fast convergence and high robustness. Nevertheless, it lacks the ability to model spatial transformation between entity pairs. The completion-based method can model this transformation; however, it is sensitive to the scoring function and relies on a large number of the prior alignment entity pairs.

5.4.2 Effectiveness on hierarchical aggregate methods

We design three additional hierarchical aggregation methods, the mean-based method, the concat-based method, and the MaxPooling-based method. In mean-based methods, we average all R-GAT layers $e^{out}_{i,mean} = \frac{1}{H}\sum_{j=1}^{m} \mathbf{e}^{j}_{i}$. In concat-based method, we concatenate all R-GAT layers and then transform the dimensions through the full connection layer $e^{out}_{i,concat} = g([\mathbf{e}^1_i, \mathbf{e}^2_i, ..., \mathbf{e}^m_i])$, where $g(\cdot)$ is a full connection layer. In MaxPooling-based method, we maximize all R-GAT layers $e^{out}_{i,MaxPooling} = \max(\mathbf{e}^1_i, \mathbf{e}^2_i, ..., \mathbf{e}^m_i)$, where $\max(\cdot)$ is element-wise. The results are shown in Table 4. We observe that nonlinear aggregation methods such as concat-based and MaxPooling-based methods do not show promising performance. Compared with nonlinear aggregation methods, the LSTM-based method is more effective than the others on Hit@10. On $DBP15K_{ZH-EN}$ and $DBP15K_{FR-EN}$, the mean-based method achieves the best performance on Hit@1 and MRR. Look carefully, the mean-based is a special form of the LSTM-based methods, which indicates that the linear aggregation methods are more effective for balancing the neighborhood information of corresponding entities.

6 Conclusion

In this paper, we introduce a dual attention mechanism entity alignment model DAEA, which contains the relation-aware graph attention module and LSTM-based hierarchical attention module. In the relation-aware graph attention module, we model relation into graph attention network and integrate translation hypothesis into the self-attention mechanism, which can mitigate the non-isomorphic difference among the neighborhood structures of counterpart entities. In the hierarchical attention module, we use the LSTM attention mechanism to learn the weight of each output layer of the relation-aware graph attention module, which can balance the neighborhood information of peer entities and distinguish non-counterpart entities with similar structures by learning low-level information and high-level information adaptively. We also design a new alignment method based on link prediction. Our experiments on three datasets demonstrate the effectiveness of DAEA. In future work, we will integrate the semantic information into the model and design the appropriate model based on the completion-based method.

7 Acknowledgments

The research work descried in this paper has been supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002103.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

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, July. Association for Computational Linguistics.

Yufeng Chen, Chengqing Zong, and Keh-Yih Su. 2013. A joint model to identify and align bilingual named entities. *Computational linguistics*, 39(2):229–266.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Xiaocheng Feng, Duyu Tang, Bing Qin, and Ting Liu. 2016. English-Chinese knowledge base translation with neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2935–2944, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 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)*, pages 2723–2732.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2015. Yago3: A knowledge base from multilingual wikipedias.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 156–166.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816.

Naoki Otani, Hirokazu Kiyomaru, Daisuke Kawahara, and Sadao Kurohashi. 2018. Cross-lingual knowledge projection using machine translation and target-side knowledge base completion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1508–1520.

Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference*, pages 3130–3136.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pages 628–644. Springer.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pages 4396–4402.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

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*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

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.

Weikang Wang, Jiajun Zhang, Qian Li, Chengqing Zong, and Zhifei Li. 2019. Are you for real? detecting identity fraud via dialogue interactions. 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 1762–1771, Hong Kong, China, November. 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-19*, pages 5278–5284.

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. Association for Computational Linguistics.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*.

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, July. Association for Computational Linguistics.

Jinghui Yan, Yining Wang, Lu Xiang, Yu Zhou, and Chengqing Zong. 2020. A knowledge-driven generative model for multi-implication chinese medical procedure entity normalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*.

Yang Zhao, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge graphs enhanced neural machine translation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4039–4045.

Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*, pages 4258–4264.

Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *See Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI*, pages 10–16.