BERT-INT: A BERT-based Interaction Model For Knowledge Graph Alignment

Xiaobin Tang[1,2] , **Jing Zhang**[1,2*] , **Bo Chen**[1,2] , **Yang Yang**[3] , **Hong Chen**[1,2] and **Cuiping Li**[1,2]
[1]Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, Renmin University of China
[2]Information School, Renmin University of China
[3]Zhejiang University
{txb, zhang-jing, bochen, chong, licuiping}@ruc.edu.cn, yangya@zju.edu.cn

Abstract

Knowledge graph alignment aims to link equivalent entities across different knowledge graphs. To utilize both the graph structures and the side information such as name, description and attributes, most of the works propagate the side information especially names through linked entities by graph neural networks. However, due to the heterogeneity of different knowledge graphs, the alignment accuracy will be suffered from aggregating different neighbors. This work presents an interaction model to only leverage the side information. Instead of aggregating neighbors, we compute the interactions between neighbors which can capture fine-grained matches of neighbors. Similarly, the interactions of attributes are also modeled. Experimental results show that our model significantly outperforms the best state-of-the-art methods by 1.9-9.7% in terms of HitRatio@1 on the dataset DBP15K.

1 Introduction

DBpedia, Freebase, YAGO and so on have been published as noteworthy large and freely available knowledge graphs (KGs), which can benefit many applications such as question answering and recommendation [Tong *et al.*, 2019]. However, a single KG is far from complete to support such applications with sufficient facts, which demands an effective way to align entities across KGs. To solve the problem, much attention has been paid to leveraging the graph structures to align entities. The specific technique has evolved from the traditional KG embedding models such as MTransE [Chen *et al.*, 2017] and IPTransE [Zhu *et al.*, 2017] to recent emergent graph neural networks such as attention-based GCN [Xu *et al.*, 2019], highway GCN [Wu *et al.*, 2019b], relation-aware GCN [Wu *et al.*, 2019a] and VR-GCN [Ye *et al.*, 2019].

Despite much effort taken on graph structures, the side information of entities such as name, description and attributes may play a more important role on many tasks on KGs, because: (1) the KGs are usually sparse, with a large number of long-tail entities whose structural embeddings have low expressiveness [Guo *et al.*, 2019]. For example, in DBpedia-ZH[1], the long-tail entities that appear less than 5 times occupy 74.1%; and (2) existing works have demonstrated the effect of side information on several tasks. For example, HMAN [Yang *et al.*, 2019] presents that on the task of entity alignment, when ignoring entity descriptions, HitRatio@1 will drop at least 30%.

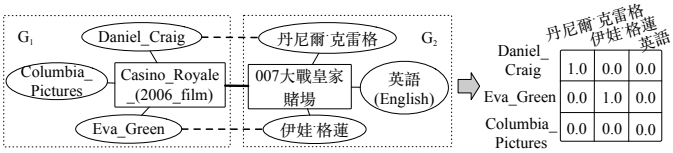


Figure 1: A motivating example (Refer to Yang et al., 2019)

To leverage the side information to align KGs, the most popular way is to initialize the embedding of each node by its side information and apply variant GCN models to update a node embedding by aggregating all neighbors' embeddings [Wang *et al.*, 2018; Wu *et al.*, 2019a; Xu *et al.*, 2019]. However, since different KGs are highly heterogeneous, it is not always the case that equivalent entities share similar neighbors. Taking an example from HMAN in Figure 1, the nodes in rectangles are entities to be aligned and the nodes in ovals are the neighbors. We can see that for the neighbor "English" in G_2, no counterpart can be found from the neighbors in G_1. In this case, propagating the dissimilar neighbors in GCNs, especially the hub entities such as the neighbor "English" with 832 relations, may introduce noises and thus harm performance [Yang *et al.*, 2019]. Although some works distinguish the influence from different neighbors [Cao *et al.*, 2019; Wu *et al.*, 2019a; Xu *et al.*, 2019], essentially, the GCN-like models still mix the side information of all the neighbors to represent an entity. HMAN has been aware of the issue, thus it thoroughly separates the modeling process of side information and graph structures. However, it discards the side information of neighbors. Moreover, similar to aggregating neighbors, it aggregates all the attributes together to represent an entity like most of the works did [Sun *et al.*, 2017; Zhang *et al.*, 2019], which also results in noisy matches between entities.

To deal with the noisy matches caused by aggregat-

*Contact Author

[1]https://wiki.dbpedia.org/downloads-2016-10

ing neighbors or attributes, we propose BERT-INT—i.e., a BERT-based INTeraction model to only leverage the side information, especially the names/descriptions of current entities and neighbors, and attributes of current entities in a unified way. Generally, we mimic the cognitive process of comparing two entities by humans, who usually first compare current entities and then continue to check whether there are similar neighbors. Following this, based on the embedding of any piece of name, description and attribute, we compare each pair of neighbors or attributes, referred to as interactions henceforth, instead of aggregating them. In this way, we can capture the fine-grained exact/semantic matches between neighbors and get rid of the negative influence from the dissimilar neighbors, as presented on the right of Figure 1.

Although we leverage the side information of neighbors, unlike the variant GCN models, the structural characteristics of KGs are totally ignored in our model, because we never propagate the information of neighbors in graphs. From this point of view, we claim that BERT-INT only leverages side information. Just because of this, BERT-INT is capable of inductive learning, i.e., we can train BERT-INT on two aligned KGs and apply it to predict the matches between unseen entities in other KGs. However, if adopting the variant GCN models, the test entities should be included in the training KGs. In summary, we propose an inductive learning-based model to comprehensively leverage the name/description-view interaction, neighbor-view interactions and attribute-view interactions to align entities. By experiments, we demonstrate that BERT-INT significantly outperforms the state-of-the-art models by 1.9-9.7% in HitRatio@1.

2 Problem Definition

Definition 1. ***Knowledge Graph****: We denote a KG as $G = (E, R, A, V)$, where each $e \in E$, $r \in R$, $a \in A$ and $v \in V$ denotes an entity, a relation, an attribute and a value respectively. $\mathcal{N}^{\tau}(e) = \{(r_i, e_i)\}_{i=1}^{|\mathcal{N}^{\tau}(e)|}$ denotes the set of all the τ-hop neighbors of entity e with τ as the number of hops, where each of the i-th neighbor contains a neighboring relation r_i and the corresponding entity e_i. $\mathcal{A}(e) = \{(a_i, v_i)\}_{i=1}^{|\mathcal{A}(e)|}$ denotes the set of the attributes of e, where each of the i-th attribute contains a name a_i and the corresponding value v_i. $\mathcal{N}(e)$ without τ indicates all the neighbors of e. $|\mathcal{N}^{\tau}(e)|$ and $|\mathcal{A}(e)|$ is the number of $\mathcal{N}^{\tau}(e)$ and $\mathcal{A}(e)$ respectively.*

Problem 1. ***Knowledge Graph Alignment****: Given two KGs G and G' and a set of already aligned entity pairs $I = \{(e \sim e')\}$, we aim at learning a ranking function $f : E \times E' \to \mathbb{R}$ to calculate a similarity score between two entities, based on which we rank the correctly aligned entity e' to any queried entity e as high as possible among all the entities in E'.*

3 BERT-INT Model

This section introduces the proposed BERT-INT, which consists of a BERT model that is used as a basic representation unit to embed the name, description, attribute and value of an entity, and an interaction model built upon the BERT embeddings to compute the interactions between these embeddings. The interactions are further divided into the name/description-view interaction, the neighbor-view interactions, and the attribute-view interactions. A unified dual aggregation function is applied to extract features from the neighbor-view and attribute-view interactions to estimate the matching score of entities. Besides, to build comprehensive neighbor-view interactions, the interactions between the neighboring entities, the corresponding neighboring relations, and the multi-hop neighbors are also considered. Figure 2 shows the whole framework.

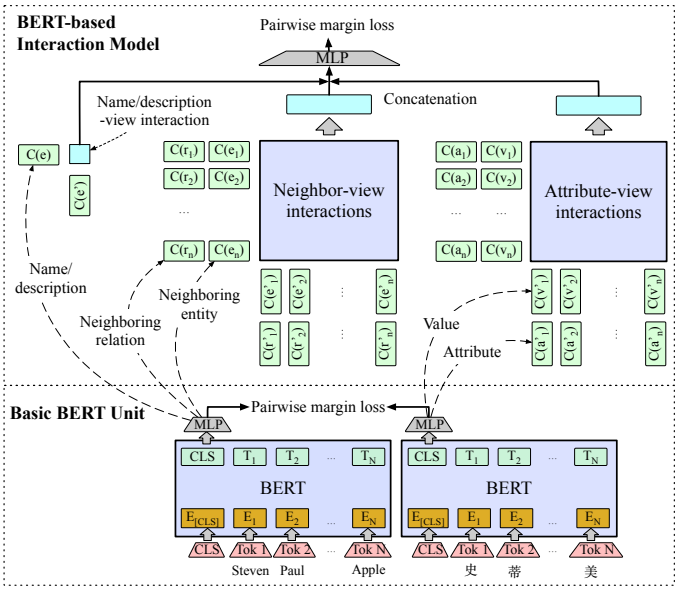


Figure 2: The framework of BERT-INT. We use BERT as a basic representation unit to embed the name/description, attribute and value of an entity, and build neighbor-view/attribute-view interaction models to compute the interactions between these embeddings.

3.1 Basic BERT Unit

We treat entity alignment as the downstream objective to fine-tune a pre-trained BERT model. Specifically, we first construct the training data $\mathcal{D} = \{(e, e'^{+}, e'^{-})\}$, where each triplet $(e, e'^{+}, e'^{-}) \in \mathcal{D}$ contains a queried entity $e \in E$, the rightly aligned counterpart $e'^{+} \in E'$ and a negative counterpart e'^{-} randomly sampled from E'. For each entity e in the dataset, we apply a pre-trained multi-lingual BERT[2] to accept its name/description as the input, filter the CLS embedding of BERT by a MLP layer to obtain

$$C(e) = \text{MLP}(\text{CLS}(e)), \tag{1}$$

and use a pairwise margin loss to fine-tune BERT:

$$\mathcal{L} = \sum_{(e,e'^{+},e'^{-}) \in \mathcal{D}} \max\{0, g(e, e'^{+}) - g(e, e'^{-}) + m)\}, \tag{2}$$

where m is a margin enforced between the positive pairs and negative pairs, and $g(e, e')$ is instantiated as l_1 distance to measure the similarity between $C(e)$ and $C(e')$. Negative

[2]https://github.com/google-research/bert

pairs are sampled according to the cosine similarity of two entities [Sun *et al.*, 2018].

For the input of BERT, we give priority to the description as it contains richer information. We use the name when the description is missing. Different from HMAN that directly uses the embeddings of the descriptions to align entities [Yang *et al.*, 2019], we use these embeddings as basic units to compose the following interaction model. Note we can also connect the basic BERT unit with the following interaction model as an end-to-end model to fine-tune BERT and train the interaction model simultaneously. In practice, considering the GPU memory and the running efficiency, we fine-tune the basic BERT unit beforehand and freeze its parameters in the following interaction model.

3.2 BERT-based Interaction Model

Based on the basic BERT unit, we build the interaction model which consists of the name/description-view interaction, the neighbor-view and the attribute-view interactions.

Name/Description-view Interaction

We apply the basic BERT unit on the names/descriptions of e and e' to obtain $C(e)$ and $C(e')$, and then calculate their cosine similarity as the name/description-view interaction.

Neighbor-view Interactions

We build the interactions between the neighbors $\mathcal{N}(e)$ and $\mathcal{N}(e')$. The general idea is to compare the names/descriptions of each neighbor pair rather than learning a global representation for e or e' by aggregating the names/descriptions of all their neighbors as existing works did [Wu *et al.*, 2019a; Xu *et al.*, 2019]. This similar idea is widely used in information retrieval to capture the exact and soft matches between a query and a candidate document [Xiong *et al.*, 2017]. Specifically, we apply the basic BERT unit to obtain $\{C(e_i)\}_{i=1}^{|\mathcal{N}(e)|}$ and $\{C(e'_i)\}_{i=1}^{|\mathcal{N}(e')|}$ for e and e''s neighboring entities based on their names/descriptions, compute a similarity matrix between the two embedding sets and then apply a dual aggregation function to extract the similarity features from the matrix. We use $\mathbf{S}$ to represent the interactions between the neighbors of e and e', with each element s_{ij} standing for the interaction, i.e., the cosine similarity $s_{ij} = \frac{C(e_i)\cdot C(e'_j)}{\|C(e_i)\|\cdot\|C(e'_j)\|}$ between $C(e_i)$ and $C(e'_j)$, where $C(e_i)$ and $C(e'_j)$ are obtained by Eq.(1) for the i-th neighbor of e and the j-th neighbor of e' respectively.

We apply a dual aggregation function to extract the similarity features along both the rows and columns of $\mathbf{S}$. CNN and RNN are usually used as the aggregation function to extract the matching patterns from the similarity matrix between two sentences. Different from sentences, the neighbors are disordered and independent from each other. Thus we adopt a RBF kernel aggregation function [Xiong *et al.*, 2017] to extract features about the accumulation of similarities.

Before making use of the RBF kernel aggregations, we first apply a max pooling operation on each row $\mathbf{S}_i = \{s_{i0}, \cdots, s_{ij}, \cdots, s_{in}\}$ to get the maximal similarity s_i^{max}, i.e., we select the most possibly aligned counterpart from the neighbors of e' for the i-th neighbor of e. This is because, following the one-one mapping assumption, we only care about how similar the most possibly aligned counterpart is to a given neighbor of e. In another word, a neighbor of e is not required to be similar to all the neighbors of e', thus other similarities except the maximal one are discarded. Next, the maximal value s_i^{max} is transformed into a row-based feature vector $\mathbf{K}^r(\mathbf{S}_i)$, with each of the l-th element $K_l(s_i^{max})$ being converted by the l-th RBF kernel with mean μ_l and variance σ_l. Then $\mathbf{K}^r(\mathbf{S}_i)$ of all the rows are averaged into a row-based similarity embedding $\phi^r(\mathcal{N}(e), \mathcal{N}(e'))$:

$$
\begin{aligned}
s_i^{max} &= \max_{j=0}^{n}\{s_{i0}, \cdots, s_{ij}, \cdots, s_{in}\} \qquad (3)\\
K_l(s_i^{max}) &= \exp\left[-\frac{(s_i^{max}-\mu_l)^2}{2\sigma_l^2}\right]\\
\mathbf{K}^r(\mathbf{S}_i) &= [K_1(s_i^{max}), \cdots, K_l(s_i^{max}), \cdots, K_L(s_i^{max})]\\
\phi^r(\mathcal{N}(e), \mathcal{N}(e')) &= \frac{1}{|\mathcal{N}(e)|}\sum_{i=1}^{|\mathcal{N}(e)|}\log \mathbf{K}^r(\mathbf{S}_i),
\end{aligned}
$$

where n is the maximal number of neighbors over all the entities. $\mathbf{S}$ will be padded zero if the real number $|\mathcal{N}(e)|$ is less than n. The kernels are used to convert the one-dimension similarity into L-dimension similarities to enhance the discrimination ability, where the kernel with $\mu = 1$ and $\sigma \to 0$ only considers the exact matches between neighbors, and others capture the semantic matches between neighbors.

The above process aggregates the features along the rows, which reflects how similar each neighbor $e_i \in \mathcal{N}(e)$ is to the neighbors $\mathcal{N}(e')$ of e'. Similarly, we also aggregate the features along the columns to capture how similar each neighbor $e'_j \in \mathcal{N}(e')$ is to the neighbors $\mathcal{N}(e)$ of e. Finally, the row-aggregation vector ϕ^r and the column-aggregation vector ϕ^c are concatenated as the final similarity embedding:

$$
\phi(\mathcal{N}(e), \mathcal{N}(e')) = \phi^r(\mathcal{N}(e), \mathcal{N}(e')) \oplus \phi^c(\mathcal{N}(e), \mathcal{N}(e')), \qquad (4)
$$

where $\oplus$ indicates the concatenation operation. Figure 3 illustrates the neighbor-view interactions.

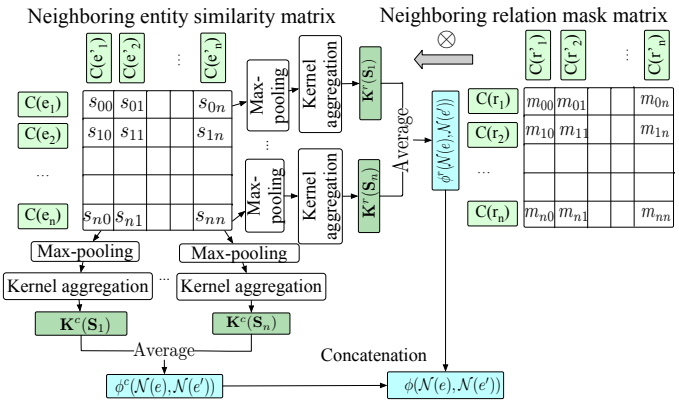


Figure 3: The neighbor-view interactions. We compute a similarity matrix between the neighbors $\mathcal{N}(e)$ and $\mathcal{N}(e')$ and apply a dual aggregation function to extract the features form the matrix.

Neighboring Relation Mask Matrix

A pair of neighbors are more convincing for supporting the alignment of two entities, if the corresponding relations are also similar. In another word, for a triplet (e, r_i, e_i) of e and a triplet (e', r'_j, e'_j) of e', if e_i is very similar to e'_j and r_i is also similar to r'_j, the two entities e and e' will be more likely to be aligned. According to this assumption, we not only compute the similarity matrix $\mathbf{S}$ between the neighboring entities but also compute the similarity matrix $\mathbf{M}$ between the corresponding neighboring relations. $\mathbf{M}$ is viewed as a mask matrix and is multiplied to $\mathbf{S}$, i.e., $S_{ij} = S_{ij} \otimes M_{ij}$, where $\otimes$ indicates elementwise product. To compute $\mathbf{M}$, we need to embed each neighboring relation. One popular way is to represent a relation approximately by its associated head-tail entity pairs based on the assumption that two relations are more similar if they associate to more similar head-tail pairs [Wu *et al.*, 2019a]. Specifically, we average $C(e)$ of all the associated head entities and also average those of all the tail entities, and concatenate them to represent the relation.

Interactions between Multi-hop Neighbors

Intuitively, the one-hop neighbors take the most important role in aligning two entities. However, as claimed by [Sun *et al.*, 2020], the multi-hop neighbors also impact the alignment results in some situations, as the direct neighbors of one entity in G will probably appear as the distant neighbors of the counterpart in G' due to the heterogeneity of two KGs. Thus, we also consider the interactions between multi-hop neighbors. Specifically, given one-hop neighbors $\mathcal{N}^1(e)$ and $\mathcal{N}^1(e')$ and multi-hop neighbors $\mathcal{N}^m(e)$ and $\mathcal{N}^m(e')$, we build the interaction matrix between $\mathcal{N}^1(e)$ and $\mathcal{N}^1(e')$, $\mathcal{N}^1(e)$ and $\mathcal{N}^m(e')$, $\mathcal{N}^m(e)$ and $\mathcal{N}^1(e')$, and $\mathcal{N}^m(e)$ and $\mathcal{N}^m(e')$, use the same aggregation functions as Eq.(3) and Eq.(4) to extract the similarity vector respectively, and concatenate them as the final neighbor similarity embedding.

Attribute-view Interactions

We also build the interactions between the attributes $\mathcal{A}(e)$ and $\mathcal{A}(e')$. Different from a unique name or description, attributes are a set of attribute-value pairs, which is similar to the neighboring relation-entity pairs. Thus similar to the neighbor-view interactions, we also compare each attribute pair rather than learning a global representation for e or e' by aggregating all their attributes as existing works did [Sun *et al.*, 2017; Zhang *et al.*, 2019]. Specifically, in Figure 3, we change the similarity matrix between the neighboring entities into that between the embeddings $\{C(v_i)\}_{i=1}^{|\mathcal{A}(e)|}$ of e's values and the embeddings $\{C(v'_i)\}_{i=1}^{|\mathcal{A}(e')|}$ of e''s values, and change the mask matrix between the neighboring relations into that between the embeddings $\{C(a_i)\}_{i=1}^{|\mathcal{A}(e)|}$ of e's attributes and the embeddings $\{C(a'_i)\}_{i=1}^{|\mathcal{A}(e')|}$ of e''s attributes. The assumption is a pair of values is more convincing for supporting the alignment of two entities if the corresponding attributes are also similar. Then we use the same aggregation functions as Eq.(3) and Eq.(4) to produce an attribute similarity vector $\phi(\mathcal{A}(e), \mathcal{A}(e'))$. Note we ignore the attributes of neighbors, as on one hand, incorporating neighbors' attributes will result in nested interactions, which is inefficient. On the other hand, for aligning e and e', their own attributes are most important.

The Final Combination

Given the cosine similarity $\cos(C(e), C(e'))$ between the descriptions/names of two entities, the neighbor similarity vector $\phi(\mathcal{N}(e), \mathcal{N}(e'))$ and the attribute similarity vector $\phi(\mathcal{A}(e), \mathcal{A}(e'))$, we concatenate them together and apply a MLP layer to get the final similarity score between e and e':

$$\begin{aligned} \phi(e, e') &= [\phi(\mathcal{N}(e), \mathcal{N}(e')) \oplus \phi(\mathcal{A}(e, \mathcal{A}(e')) \oplus \cos(C(e), C(e'))], \\ g(e, e') &= \text{MLP}(\phi(e, e')). \end{aligned} \tag{5}$$

Finally, we inject $g(e, e')$ into the same pairwise loss function as Eq.(2) to optimize the parameters of MLP. Note that the parameters of BERT have been fine-tuned by the basic BERT unit in Section 3.1 and are frozen at this stage.

3.3 Entity Alignment

Given an entity e from G, we first quickly filter top-κ candidates from G' and then accurately infer the counterpart from the candidates. Specifically, we apply the basic BERT unit to obtain an embedding for each entity by Eq.(1), compute the cosine similarity between the embedding of e from G and the embedding of each entity from G', and return the top-κ similar entities as candidates of e. Then for e and each candidate, we apply the BERT-based interaction model to infer a matching score between them and rank all the candidates for evaluation. The candidate selection process can significantly improve the alignment efficiency by the interaction model.

4 Experiments

4.1 Datasets and Settings

We evaluate our model on the widely used cross-lingual dataset DBP15K and the mono-lingual dataset DWY100K and use HitRatio@K (K=1,10) and MRR to evaluate (Cf. [Sun *et al.*, 2018] for details). The dimension of the BERT CLS embedding is 768. We use a 300-dimension MLP in Eq.(1) and a 11 plus 1-dimension MLP in Eq.(5). The maximal number of neighbors and attributes are both set as 50. In Eq.(3), we use 20 semantic matching kernels, where μ is from 0.025 to 0.975 with interval 0.05 and all $\sigma = 0.1$, and use an exact matching kernel with $\mu = 1.0$ and $\sigma = 10^{-3}$. The number of the returned candidates by the basic BERT unit, i.e., κ is set as 50, as we find that 99% ground truth can be included in the top-50 candidates. The margin m in Eq.(2) for fine-tuning BERT is set as 3, and for training the interaction model is set as 1. Codes and datasets are online now[3].

4.2 Experimental Results

Overall Performance on DBP15K

We compare all the state-of-the-art models with available results or codes. Some methods such as [Chen *et al.*, 2018; Trsedya *et al.*, 2019] are not compared due to the code implementation issue. In principle, we divide them into the category that only utilizes graph structures and the one that uses additional side information. We further divide the former category into three fine-grained types: variant TransE, variant

[3]https://github.com/kosugi11037/bert-int

Model	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$		
	HR1	HR10	MRR	HR1	HR10	MRR	HR1	HR10	MRR
Only use graph structures by variant TransE									
MTransE [Chen *et al.*, 2017]	0.308	0.614	0.364	0.279	0.575	0.349	0.244	0.556	0.335
IPTransE [Zhu *et al.*, 2017]	0.406	0.735	0.516	0.367	0.693	0.474	0.333	0.685	0.451
BootEA [Sun *et al.*, 2018]	0.629	0.848	0.703	0.622	0.854	0.701	0.653	0.874	0.731
RSNs [Guo *et al.*, 2019]	0.508	0.745	0.591	0.507	0.737	0.590	0.516	0.768	0.605
TransEdge [Sun *et al.*, 2019]	0.735	0.919	0.801	0.719	0.932	0.795	0.710	0.941	0.796
MRPEA [Shi and Xiao, 2019]	0.681	0.867	0.748	0.655	0.859	0.727	0.677	0.890	0.755
Only use graph structures by variant TransE plus GCN									
MuGNN [Cao *et al.*, 2019]	0.494	0.844	0.611	0.501	0.857	0.621	0.495	0.870	0.621
NAEA [Zhu *et al.*, 2019]	0.650	0.867	0.720	0.641	0.873	0.718	0.673	0.894	0.752
KECG [Li *et al.*, 2019]	0.478	0.835	0.598	0.490	0.844	0.610	0.486	0.851	0.610
AliNet [Sun *et al.*, 2020]	0.539	0.826	0.628	0.549	0.831	0.645	0.552	0.852	0.657
Only use graph structures by variant TransE plus adversarial learning									
AKE [Lin *et al.*, 2019]	0.325	0.703	0.449	0.259	0.663	0.390	0.287	0.681	0.416
SEA [Pei *et al.*, 2019]	0.424	0.796	0.548	0.385	0.783	0.518	0.400	0.797	0.533
Combine graph structures and side information by variant GCN									
GCN-Align [Wang *et al.*, 2018]	0.413	0.744	0.549	0.399	0.745	0.546	0.373	0.745	0.532
GM-Align [Xu *et al.*, 2019]	0.679	0.785	-	0.740	0.872	-	0.894	0.952	-
RDGCN [Wu *et al.*, 2019a]	0.708	0.846	0.746	0.767	0.895	0.812	0.886	0.957	0.911
HGCN [Wu *et al.*, 2019b]	0.720	0.857	0.768	0.766	0.897	0.813	0.892	0.961	0.917
DGMC [Fey *et al.*, 2020]	0.772	0.897	-	0.774	0.907	-	0.891	0.967	-
Combine graph structures and side information by multi-view learning									
JAPE [Sun *et al.*, 2017]	0.412	0.745	0.490	0.363	0.685	0.476	0.324	0.667	0.430
MultiKE [Zhang *et al.*, 2019]	0.509	0.576	0.532	0.393	0.489	0.426	0.639	0.712	0.665
JarKA [Chen *et al.*, 2020]	0.706	0.878	0.766	0.646	0.855	0.708	0.704	0.888	0.768
HMAN [Yang *et al.*, 2019]	0.871	0.987	-	0.935	0.994	-	0.973	0.998	-
CEAFF [Zeng *et al.*, 2020]	0.795	-	-	0.860	-	-	0.964	-	-
BERT-INT	**0.968**	**0.990**	**0.977**	**0.964**	**0.991**	**0.975**	**0.992**	**0.998**	**0.995**

Table 1: Overall performance of entity alignment on DBP15K.

TransE plus GCN, and variant TransE plus adversarial learning, and divide the later one into two fine-grained types: variant GCN and multi-view learning. Table 1 shows the overall performance on DBP15K. Generally, we conclude that including side information can obtain better performance than only considering graph structures.

Only using Graph Structures. We analyze the differences of the methods only using the graph structures. Among variant TransE methods, BootEA, TransEdge and MRPEA perform the best, where BootEA and TransEdge both bootstrap the labeled alignments iteratively, which is the key technique to improve the alignment performance. Strictly, MRPEA is a little far away from TransE, as it changes the subtraction between entities and relations into multiplications, which is more friendly to the multi-mapping relations. Several works combine the advantages of TransE and GCN by convolving all the neighboring information for an entity as GCN does and meanwhile keeping the translation relationship among head, relation and tail as TransE does. Among them, NAEA also bootstraps the labeled alignments. For other methods without bootstrapping, we cannot observe significant improvement by incorporating GCN. This may due to the heterogeneity of KGs. Two neighboring entities in KGs are more likely to be translated to each other by their relation, rather than similar to each other as assumed by GCN. Besides, we notice that based on TransE, incorporating adversarial learning cannot obtain expected performance. This is also caused by the high heterogeneity of KGs, which results in the difficulty of transferring from one KG to another by a similar linear mapping function as transferring between multi-lingual word spaces [Conneau *et al.*, 2018].

Combining Graph Structures and Side Information. We analyze the differences of the methods that also incorporate the side information. Among the methods that model the graph structures and the side information in a unified GCN model, GM-Align, RDGCN, and HGCN distinguish the effects of different neighbors, thus they perform better than the original GCN-Align model. Among the methods that model the graph structures and the side information separately and then combine them by multi-view learning, CEAFF that emphasizes the effect of name and HMAN that uses BERT to embed descriptions, significantly outperform others. DGMC and CEAFF both solve the global alignment inconsistency and obtain excellent performance. CEAFF further outperforms DGMC by 2.3-8.6% in HR1, which indicates that separating the two views can reduce noises and is more reasonable than mixing the two views in a unified GCN network.

Only using Side Information. Our model BERT-INT only uses the side information. Despite the missing of graph structures, BERT-INT outperforms the best baselines by 9.7%, 2.9% and 1.9% in HR1 on ZH-EN, JA-EN, and FR-EN respectively, which indicates that the semantics of the side information is more powerful than the structural characteristics. Compared with all the other methods, we directly compare neighbors or attributes through the interaction mechanism, which can improve the matching accuracy between entities.

Overall Performance on DWY100K

Since CEAFF [Zeng *et al.*, 2020] has already obtained 100% HR1 on the mono-lingual dataset DWY100K, we only compare our model with it. We use the name as the basic representation of an entity due to the missing descriptions in the dataset. On the two datasets of DWY100K, we obtain 99.2% and 99.9% HR1 respectively, which are comparable to CEAFF. However, we also observe that the names for most of the aligned entities are exactly the same, which demands more challenging mono-lingual datasets [Zeng *et al.*, 2020].

Ablation Study

We perform ablation study from three aspects and show the results in Table 2. All the variants are based on BERT-INT.

Remove Components from BERT-INT. When removing the max-pooling operations from the interaction model[4],

[4]We remove both the row-based and column-based max-pooling.

Model	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$		
	HR1	HR10	MRR	HR1	HR10	MRR	HR1	HR10	MRR
BERT-INT	**0.968**	**0.990**	**0.977**	**0.964**	**0.991**	**0.975**	**0.992**	**0.998**	**0.995**
Remove components									
-max pooling	0.962	0.989	0.973	0.959	0.991	0.973	0.992	0.998	0.995
-column aggregation	0.960	0.989	0.971	0.959	0.990	0.971	0.991	0.998	0.994
-neighbors	0.947	0.987	0.963	0.937	0.986	0.956	0.988	0.998	0.992
-attributes	0.919	0.984	0.945	0.938	0.987	0.957	0.983	0.998	0.990
-neighbors & attributes	0.830	0.970	0.883	0.848	0.974	0.897	0.965	0.995	0.978
Change the interaction component to variant GCN									
BERT-GCN	0.736	0.950	0.799	0.767	0.960	0.824	0.914	0.992	0.936
BERT-RDGCN	0.847	0.974	0.896	0.857	0.969	0.900	0.952	0.990	0.967
BERT-HMAN	0.911	0.993	0.943	0.937	0.994	0.960	0.982	0.999	0.989
Add components									
+relation mask	0.966	0.989	0.975	0.962	0.990	0.973	0.992	0.998	0.995
+attribute mask	0.942	0.986	0.959	0.950	0.990	0.966	0.989	0.998	0.993
+2-hop neighbors	0.965	0.990	0.975	0.964	0.991	0.975	0.992	0.998	0.995

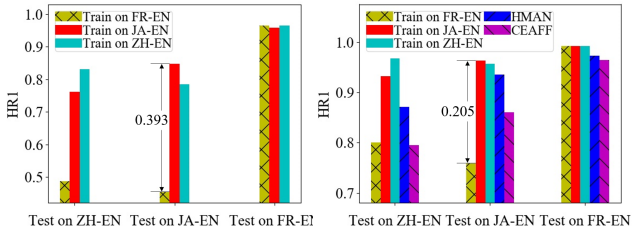
Table 2: Ablation study on DBP15K.

the performance is reduced by maximal 0.6%, which indicates it is more effective to only capture the most possibly aligned counterpart rather than taking the similarities to all the neighbors/attributes into consideration. When removing the column-aggregation[5], the performance is reduced by maximal 0.8%, which indicates that we need to consider the similarity of each neighbor to all the corresponding neighbors from both the directions. When removing the neighbors or the attributes, the performance is reduced by maximal 2.7% or 4.9% respectively, and significantly reduced by maximal 13.8% when removing both of them, i.e., only using the names/descriptions of entities. The results demonstrate modeling the interactions of neighbors and attributes is effective.

Change Interaction Model in BERT-INT to GCN. We initialize each node e by $C(e)$ in Eq.(1), and update node embeddings by the original GCN or RDGCN [Wu *et al.*, 2019a]. Compared with BERT-INT, HR1 is reduced by maximal 23.2%, and is even 1.3-9.4% worse than -neighbors & attributes, i.e., our model that only uses the names/descriptions of entities. The results indicate that when the side information is powerfully modeled by BERT, introducing them in GCNs may lose effect, and even introduce noises. For fair comparison, we change the BERT embeddings in HMAN by ours [Yang *et al.*, 2019]. The result of BERT-HMAN also under-performs BERT-INT by maximal 5.7% HR1, which reveals the effectiveness of the interaction part in BERT-INT.

Add Components into BERT-INT. BERT-INT does not include the relation/attribute mask matrix and the multi-hop neighbors. We add each of them to BERT-INT to validate the effect. Unfortunately, none of them present an improvement on BERT-INT. We analyze the reasons as follows. Following [Wu *et al.*, 2019a], we represent a relation by all its associated head-tail entities, which is too general to reflect the exact meaning of the relation. We also tried to represent a relation by its name but obtained the similar results, as names are too short to represent a relation completely. Attributes can be explained in the same way. To test the effect of multi-hop neighbors, we take 2-hop neighbors as an example. As some

[5]Removing row-aggregation obtains the similar results.



(a) The basic BERT unit (b) BERT-INT

Figure 4: Inductive learning on DBP15K. Different colors indicate the models are trained on the datasets in different languages. HMAN and CEAFF both train and test on the datasets in the same language.

entities have a large number of 2-hop neighbors, to reduce the noises, for each pair of entities, we select top-k similar 2-hop neighbor pairs based on their BERT embeddings[6]. As the BERT-based neighbor/attribute interactions are already powerful, it limits the effect of the 2-hop neighbors. Although AliNet [Sun *et al.*, 2020] shows the effectiveness of the multi-hop neighbors, it only uses the structure embeddings, which are not as powerful as BERT embeddings, thus it needs more structural information from the multi-hop neighbors.

Inductive Learning

We train the basic BERT unit and BERT-INT on one dataset of DBP15K, directly transfer and evaluate them on the other two datasets, and show the results in Figure 4(a) and Figure 4(b). When training BERT-INT on ZH-EN/JA-EN and testing on JA-EN/ZH-EN, the results are even better than HMAN and CEAFF, which reveals ZH and JA are easier to be transferred between each other. No matter training on any dataset, the test performance on FR-EN is comparable to the normal one (i.e., train and test on FR-EN), because FR is similar to EN, making the alignment avoids the affect of different languages. We also observe that the basic BERT unit also has the inductive capacity. However, on the basis of the normal performance on JA-EN, the reduced HR1 by the transferred BERT-INT is 18.8% less than that by the the transferred basic BERT unit, which reveals the inductive capacity is not only caused by multi-lingual BERT, but also caused by the proposed interaction model.

5 Conclusions

This paper solves knowledge graph alignment by building the interactions between neighbors or attributes based on their BERT embeddings, which can obtain fine-grained matches of neighbors or attributes. The proposed model can achieve the best performance among all the state-of-the-art models and can enable inductive learning compared with other models.

ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China (No.2018YFB1004401) and NSFC (No.61532021, 61772537, 61772536, 61702522)

[6]We vary k from 10 to 200 and find the best value as 50.

References

[Cao *et al.*, 2019] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. Multi-channel graph neural network for entity alignment. In *ACL'19*, pages 1452–1461, 2019.

[Chen *et al.*, 2017] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI'17*, pages 1511–1517, 2017.

[Chen *et al.*, 2018] Muhao Chen, Yingtao Tian, Kaiwei Chang, Steven Skiena, and Carlo Zaniolo. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *IJCAI'18*, pages 3998–4004, 2018.

[Chen *et al.*, 2020] Bo Chen, Jing Zhang, Xiaobin Tang, Hong Chen, and Cuiping Li. Jarka: Modeling attribute interactions for cross-lingual knowledge alignment. In *PAKDD'20*, 2020.

[Conneau *et al.*, 2018] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *ICLR'18*, 2018.

[Fey *et al.*, 2020] Matthias Fey, Jan E. Lenssen, Christopher Morris, Jonathan Masci, and Nils M. Kriege. Deep graph matching consensus. In *ICLR'20*, 2020.

[Guo *et al.*, 2019] Lingbing Guo, Zequn Sun, and Wei Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *ICML'19*, 2019.

[Li *et al.*, 2019] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *EMNLP'19*, pages 2723–2732, 2019.

[Lin *et al.*, 2019] Xixun Lin, Hong Yang, Jia Wu, Chuan Zhou, and Bin Wang. Guiding cross-lingual entity alignment via adversarial knowledge embedding. In *ICDM'19*, pages 429–438, 2019.

[Pei *et al.*, 2019] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *WWW'19*, pages 3130–3136, 2019.

[Shi and Xiao, 2019] Xiaofei Shi and Yanghua Xiao. Modeling multi-mapping relations for precise cross-lingual entity alignment. In *EMNLP'19*, pages 813–822, 2019.

[Sun *et al.*, 2017] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC'17*, pages 628–644, 2017.

[Sun *et al.*, 2018] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI'18*, pages 4396–4402, 2018.

[Sun *et al.*, 2019] Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *ISWC'19*, pages 612–629, 2019.

[Sun *et al.*, 2020] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI'20*, 2020.

[Tong *et al.*, 2019] Peihao Tong, Junjie Yao, Linzi He, and Liang Xu. Leveraging domain context for question answering over knowledge graph. *Data Science and Engineering*, 4:323–335, 2019.

[Trsedya *et al.*, 2019] Bayu Distiawan Trsedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI'19*, 2019.

[Wang *et al.*, 2018] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP'18*, pages 349–357, 2018.

[Wu *et al.*, 2019a] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI'19*, pages 5278–5284, 2019.

[Wu *et al.*, 2019b] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. Jointly learning entity and relation representations for entity alignment. In *EMNLP'19*, pages 240–249, 2019.

[Xiong *et al.*, 2017] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. End-to-end neural ad-hoc ranking with kernel pooling. In *SIGIR'17*, pages 55–64, 2017.

[Xu *et al.*, 2019] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual knowledge graph alignment via graph matching neural network. In *ACL'19*, pages 1452–1461, 2019.

[Yang *et al.*, 2019] Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. Aligning cross-lingual entities with multi-aspect information. In *EMNLP'19*, pages 4430–4440, 2019.

[Ye *et al.*, 2019] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI'19*, pages 4135–4141, 2019.

[Zeng *et al.*, 2020] Wexin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. Ceaff:collective embedding-based entity alignment via adaptive features. In *ICDE'20*, 2020.

[Zhang *et al.*, 2019] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Multi-view knowledge graph embedding for entity alignment. In *IJCAI'19*, pages 5429–5435, 2019.

[Zhu *et al.*, 2017] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative entity alignment via joint knowledge embeddings. In *IJCAI'17*, pages 4258–4264, 2017.

[Zhu *et al.*, 2019] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *IJCAI'19*, pages 3231–3237, 2019.