Neighborhood-Aware Attentional Representation for Multilingual Knowledge Graphs

Qiannan Zhu¹², Xiaofei Zhou¹², Jia Wu³, Jianlong Tan¹² and Li Guo¹²

¹Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
²School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
³Department of Computing, Macquarie University, Sydney, Australia

{zhouxiaofei,zhuqiannan}@iie.ac.cn, Jia.Wu@mq.edu.au

Abstract

Multilingual knowledge graphs constructed by entity alignment are the indispensable resources for numerous AI-related applications. Most existing entity alignment methods only use the triplet-based knowledge to find the aligned entities across multilingual knowledge graphs, they usually ignore the neighborhood subgraph knowledge of entities that implies more richer alignment information for aligning entities. In this paper, we incorporate neighborhood subgraph-level information of entities, and propose a neighborhood-aware attentional representation method NAEA for multilingual knowledge graphs. NAEA devises an attention mechanism to learn neighbor-level representation by aggregating neighbors’ representations with a weighted combination. The attention mechanism enables entities not only capture different impacts of their neighbors on themselves, but also attend over their neighbors’ feature representations with different importance. We evaluate our model on two real-world datasets DBP15K and DWY100K. and the experimental results show that the proposed model NAEA significantly and consistently outperforms state-of-the-art entity alignment models.

1 Introduction

The multilingual knowledge graphs (KGs) like YAGO [Suchanek et al., 2008] and DBpedia [Bizer et al., 2009] increasingly play an significant role in supporting various knowledge-driven tasks. Those multilingual knowledge graphs consist of monolingual knowledge, in forms of directed graphs, where entities are represented as nodes and relations as edges. The monolingual knowledge are stored as triplets (e_h, r, e_t), representing that the head entity e_h and tail entity e_t are linked by relation r. Besides monolingual knowledge, the multilingual KGs also embody cross-lingual knowledge (e_h, align(), e'_h) that matches the same real-world entities e_h and e'_h among different human languages L and L' by alignment operation align(), see Figure 1. A great deal of methods focus on exploiting monolingual knowledge graphs in recent years. Particularly, the embedding-based methods that encode entities and relations into continue low-dimensional vector spaces, have achieved promising performance. Exemplarily, given a triplet (e_h, r, e_t), TransE [Bordes et al., 2013] regards the relation embedding r as the translation vector between the head and tail entity embedding e_h and e_t, and expects e_h + r ≈ e_t when (e_h, r, e_t) holds. Other extended works such as TransH [Wang et al., 2014], TransR [Lin et al., 2015a] and TransD [Ji et al., 2015] also emerged with different translation forms in characterizing relation r. However, a few methods have been done for modeling multilingual knowledge graphs.

Entity alignment is an effective way to integrate the multilingual KGs, which is the task of finding the same real-world entities in different KGs. The traditional multilingual entity alignment methods mainly based on machine translation, have low accuracy due to the poor performance in translation between multiple languages. Most recently, following above popular embedding-based models, MTransE [Chen et al., 2017] provides the cross-lingual transitions for both entities and relations across different knowledge graph embeddings. IPTransE [Zhu et al., 2017] jointly encodes both entities and relations of various KGs into an unified low-dimensional semantic space via sharing parameters on a seed set of aligned entities, JAPE [Sun et al., 2017] further incorporates attribute triplets as additional information for learning KGs’ embeddings in an unified space. BootEA [Sun et al., 2018] adopts bootstrapping [Yarowsky, 1995; Abney, 2004] approach to iteratively label likely entity alignment as training data and leverage it for learning alignment-oriented embeddings. Existing entity alignment methods only use the triplet-based information, but ignore the inherent neighborhood information of entities for aligning entities.

*Corresponding Author
Compared with the triplet-based information, the neighbor-level information factually implies more richer alignment properties because arbitrary two aligned (equivalent) entities in different KGs usually contain equivalent neighborhood information.

This paper proposes a neighborhood-aware attentional representation method NAEA for multilingual knowledge graph. As Figure 2 shown, NAEA are composed of knowledge embedding component KE and entity alignment component EA. Both KE and EA components incorporate neighbor-level and relation-level information with different weights for learning KGs’ embedding representation and aligning entities respectively. KE component devises an attention mechanism to obtain neighbor-level representation by aggregating neighbors with a weighted combination, and uses the triplet-based characteristic information to learn relation-level representation. EA component determines alignments between entities across multilingual KGs by measuring the similarity of their integrated representations, which are also fused from neighbor-level and relation-level representations with different weights.

Our contributions. We propose a neighborhood-aware attentional representation model NAEA for multilingual knowledge graphs, which can better capture the entity and relation features for targeting aligned entities. Two components, knowledge embedding component KE and entity alignment component EA, are included in NAEA.

- NAEA incorporates neighbor-level with relation-level feature information in knowledge graphs to perform multilingual entity alignment task.
- KE component devises a neighborhood-aware attention mechanism to learn neighbor-level representation by aggregating neighbors’ embeddings with different impacts.
- EA component performs entity alignment by the similarity of entity integrated representations, which integrates neighbor-level and relation-level feature representation.

We evaluate our NAEA model on two real-world datasets DBP15K and DWY100K. Experimental results show that NAEA significantly achieves state-of-the-art performance on entity alignment task.

2 Problem Formulation

We describe an entity set as $E$, a relation set as $R$, a knowledge graph as $G = \{\tau | \tau = (e_h, r, e_l)\}$ with $e_h, e_l \in E, r \in R$. A KG pair specified by languages $L_1$ and $L_2$ is $G' = (G_{L_1}, G_{L_2})$, and its known aligned entities set is $A = \{(e_{j_1}, e_{j_2}) | e_{j_1} \in E_{L_1}, e_{j_2} \in E_{L_2}\}$, where $E_{L_1}$ and $E_{L_2}$ are respectively the entity set of $G_{L_1}$ and $G_{L_2}$. Each aligned entity pair $(e_{j_1}, e_{j_2})$ represents an entity $e_{j_1}$ in $G_{L_1}$, has its synonymous counterpart $e_{j_2}$ in $G_{L_2}$ with language-specific surface names.

The entity alignment task in this paper is as follows: given the language-specific knowledge graph pair $G' = (G_{L_1}, G_{L_2})$, and the known alignment seeds $A = \{(e_{j_1}, e_{j_2})\}$, entity alignment aims to automatically find and align more unaligned entities.

3 Model

This paper proposes a neighborhood-aware attentional representation method NAEA for multilingual knowledge graphs. As illustrated in Figure 2, NAEA consists of knowledge embedding component KE and entity alignment component EA: (1) KE in section 3.1 aims to learn alignment-oriented knowledge embedding representation, and (2) EA in section 3.2 aims to perform alignment in various KGs.

3.1 Neighborhood-aware Knowledge Embedding

KE fuses neighbor-level and relation-level information, and introduces an attention mechanism to learn neighbor-level representation of heterogeneous KGs.

Neighbor-level Attentional Representation

For obtaining neighbor-level entity representation, we introduce an attention mechanism to aggregate entities’ neighbors with different importance. As shown in Figure 2, the attention mechanism takes as input an entity $e_h$ with its neighborhood set $\{(r_{h1}, e_{h1}), (r_{h2}, e_{h2}), \ldots, (r_{hn}, e_{hn})\}$, and outputs its corresponding neighbor-level feature representation $Ne(e_h)$.

Here $r_{hj}$ with $j = 1, \ldots, n$ is the relation that links from entity $e_h$ to $e_{hj}$ or vice versa, and $n$ is the number of neighbors of entity $e_h$. 

![Figure 2: Simple Visualization of NAEA Model.](image-url)
Specifically, for each entity-neighbor pair, we first use two shared linear transformations, parameterized by two weight matrices $W_1 \in R^{m \times m}$ and $W_2 \in R^{m \times 2m}$, to transform the input features into higher-level features, and then adopt self-attention on entities to compute attention coefficients between entity $e_h$ and its neighborhood entity $e_hj$ with relation $r_{hj}$, i.e.,

$$c_{h,j} = a_h^T [W_1 e_h, W_2 r_{hj}, e_{hj}]$$

where $c_{h,j}$ represents the importance of entity $e_hj$’s features to entity $e_h$ with relation $r_{hj}$, $a_h \in R^{2m}$ is an entity-specific weight vector, $e_h, e_{hj}, r_{hj} \in R^m$ are the vector embeddings of $e_h, e_{hj}, r_{hj}$, and $[\cdot]_k$ indicates the concatenation operation. $m$ is the dimension of entity and relation embedding space.

Obviously the above self-attention mechanism allows to calculate the importance of arbitrary two linked entities. To make coefficients easily comparable across different entities, we normalize them across all choices of $j = 1, 2, ..., n$ using the softmax function:

$$\alpha_{h,j} = \text{softmax}(c_{h,j}) = \frac{\exp(\text{LeakyReLU}(c_{h,j}))}{\sum_{i=1}^{n} \exp(\text{LeakyReLU}(c_{h,i}))}$$

To obtain the final output neighbor-level feature representation $Ne(e_h)$ of input entity $e_h$, we compute a linear combination of its neighbors’ high-level features with different weight coefficients $\alpha_{h,j}$, and adopt multi-head attention [Vaswani et al., 2017] to stabilize the process of self-attention, i.e.,

$$Ne(e_h) = \phi \left( \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{n} \alpha_{h,j}^k W_2^k [r_{hj}, e_{hj}] \right)$$

where $\phi = \text{sigmoid}$ is the activation function, $\alpha_{h,j}^k$ are the normalized attention coefficients computed by the $k$-th attention mechanism, and $K$ is the number of head. Although the multi-head attention mechanism expands the parameter requirements by a factor of $K$, the individual heads’ computations are fully independent and can be parallelized.

We mainly perform such attention mechanism to get neighbor-level entity representations. For the sake of fairness, we use $Nr(r) = Wr$ to represent the neighbor-level relation representation of relation $r$ with $W \in R^{m \times m}$.

Relation-level Representation
Besides utilizing the neighbor-level information, we also consider the triplet-based information for our model NAEA. In order to fully preserve the inherent structural property of triplets, we use $(e_h, r, e_t)$ to describe the relation-level embedding representation of a triplet $(e_h, r, e_t)$.

Loss Function
For capturing neighbor-level information of knowledge graphs, similar to the widely-used embedding based model TransE [Bordes et al., 2013], we define the neighbor-level score function $f_1(\tau) = \parallel Ne(e_h) + Nr(r) - Ne(e_t) \parallel^2_2$ for a triplet $\tau = (e_h, r, e_t)$. Different from the margin-based ranking loss function used in TransE, we adopt following list-based scoring loss function [Zhou et al., 2017] since it not only ensures the discrimination between the scores of positive and negative triplets, but also ensures the lower scores for positive triplets, i.e.,

$$L_1 = \sum_{\tau \in T} \sum_{\tau' \in T'} [f_1(\tau) + \mu_1 - f_1(\tau') + \gamma \sum_{\tau \in T} [f_1(\tau) - \mu_2]_]$$

where $[\cdot]_+$ means that $\max(\cdot, 0)$, $\tau$ and $\tau'$ are respectively the positive and negative triplet from positive and negative triplet set $T$ and $T'$, $\gamma > 0$ is a balance hyper-parameter, and $\mu_1, \mu_2$ are two hyper-parameters with the constraint $\mu_1, \mu_2 > 0$.

For capturing relation-level information of knowledge graphs, we also define the relation-level score function $f_2(\tau) = \parallel e_h + r - e_t \parallel^2_2$ for measuring the plausibility of the triplet $\tau = (e_h, r, e_t)$, and use the same loss framework as $L_1$ to define relation-level scoring loss function $L_2$.

We aim to learn embedding representations of KGs by fusing neighbor-level and relation-level information of knowledge graphs with different weights. Therefore, we propose following combined loss function for optimization:

$$L_E = \beta L_1 + (1 - \beta) L_2$$

where $\beta$ is the weight hyper-parameter that balances the importance of neighbor-level of information.

3.2 Entity Alignment
Entity alignment aims to find $A' = \{(e_i, e_j) \in E_{L_1} \times E_{L_2} \mid e_i \equiv e_j\}$, where $e_i \equiv e_j$ means that an equivalence relation $\equiv$, holds between $e_i$ and $e_j$. Usually the already known aligned pairs subset $A = \{(e_i, e_j) \mid e_i \in E_{L_1}, e_j \in E_{L_2}\}$ of $A'$ is used as training data.

In this paper, we consider the alignment task as the classification problem that labels entity $e_i$ in $G_{L_1}$ using entity $e_j$ in $G_{L_2}$. As said in [Lacoste-Julien et al., 2013; Zhang et al., 2015; Sun et al., 2018], this entity alignment task has the one-to-one alignment constraint: an entity can be aligned with at most one label, and a label can be assigned to at most one entity. This constraint makes difference between our problem and the common classification problem [Liu et al., 2019; Liu et al., 2017]. We define the alignment probability of using entity $e_j$ to label $e_i$ as

$$\pi(e_j|e_i) = \beta \phi(\text{cos}(Ne(e_i), Ne(e_j))) + (1 - \beta) \phi(\text{cos}(e_i, e_j))$$

where $\phi(\cdot)$ is the cosine similarity. Based on neighbor-level and relation-level knowledge information, we minimize the following negative log-likelihood function to obtain the optimal parameters with the highest alignment likelihood:

$$L_A = - \sum_{e_i \in E_{L_1}} \sum_{e_j \in E_{L_2}} \log \pi(e_j|e_i)$$

where 1 is the indicator function that equals to 1 when $e_j$ is the true label of entity $e_i$, otherwise 0.

3.3 Training
To enable parameters of our model not only to capture the alignment likelihood information, but also model the inherent
semantic knowledge of KGs. We minimize following joint objective function to perform model training:
\[ L = L_E + \lambda L_A \]
where \( \lambda \) is the positive hyper-parameter that measures the importance of \( L_A \).

In training stage, for regularization and avoiding over-fitting, we apply dropout to the weight parameters \( W, W_1, W_2 \) and L2 regularization to embeddings of KGs.

For addressing the inadequate prior alignment, we use bootstrapping strategy used in [Sun et al., 2018] that iteratively label likely alignment as training data to train our model.

**Construct \( T \) and \( T' \).** For constructing the positive triplets set \( T \), we deliberately exchange the entity with its counterpart in triplets to calibrate different KGs in the unified embedding space. Given an aligned entity pair \((e_f, e_t) \in A\), following positive triplets are generated:

\[ T_a(e_f, e_t) = \{(e_f, r, e_t) | (e_f, r, e_t) \in T_{L_1} \} \]
\[ \cup \{(e_t, r, e_f) | (e_t, r, e_f) \in T_{L_2} \} \]
\[ \cup \{(e_f, r, e_t) | (e_f, r, e_t) \in T_A \} \]

where \( T_{L_1} \) and \( T_{L_2} \) are respectively the positive triplets set of knowledge graphs \( G_{L_1} \) and \( G_{L_2} \). Thus we have the general positive triplets set in Eq. (1) as \( T = T_{L_1} \cup T_{L_2} \cup T_A \) with \( T_A = \bigcup_{(e_f, e_t) \in A} T_a(e_f, e_t) \). For constructing the negative triplets set \( T' \), instead of the traditional uniform negative sampling in [Bordes et al., 2013] that samples the replacer of entity from entire entity set, we adopt Nearest Neighbor Sampling strategy to limit the scope of sampling candidates. Specifically, for an entity \( e \) that will be replaced, we choose the top-s nearest neighbors of entity \( e \) in the embedding space as candidates. In this way, those entities that have low embedding similarities with entity \( e \), are truncated and would not be sampled. Here, we use cosine similarity between integrated representation of entities to determine the top-s nearest neighbors of entities. The integrated representation of entity \( e \) is defined as \( \beta \text{Ne}(e) + (1 - \beta)e \).

4 Experiment

4.1 Experiment Setup

**Datasets.** We conduct experiments on two real-world datasets DBP15K and DWY100K. DBP15K [Sun et al., 2017] is selected from the multilingual versions of DBpedia that includes entity alignment links from entities of English version to those in other languages. Three multilingual datasets DBPZH-EN (Chinese to English), DBPJ-A-EN (Japanese to English) and DBPF-EN (French to English) are built in DBP15K. Each dataset contains 15 thousand reference alignment links with popular entities from English to Chinese, Japanese and French respectively. DWY100K [Sun et al., 2018] is built from three large-scale multi-lingual knowledge graph DBpedia, Wikidata and YAGO3. Two large-scale datasets, DBP-WD and DBP-YG, are respectively extracted with 100 thousand reference alignment links from the English version of DBpedia to that of Wikidata and YAGO3. Table 1 illustrates the statistics of those data sets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>#Ent</th>
<th>#Rel</th>
<th>#Att</th>
<th>#Rel.tri</th>
<th>#Att.tri</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBP15KZH-EN</td>
<td>66,469</td>
<td>2,830</td>
<td>8,113</td>
<td>153,929</td>
<td>379,684</td>
</tr>
<tr>
<td>DBP15KJA-EN</td>
<td>65,744</td>
<td>2,043</td>
<td>5,882</td>
<td>164,373</td>
<td>354,619</td>
</tr>
<tr>
<td>DBP15KFR-EN</td>
<td>66,858</td>
<td>1,379</td>
<td>4,547</td>
<td>192,191</td>
<td>528,665</td>
</tr>
<tr>
<td>DBP-WD</td>
<td>100,000</td>
<td>330</td>
<td>351</td>
<td>463,294</td>
<td>818,156</td>
</tr>
<tr>
<td>DBP-YG</td>
<td>100,000</td>
<td>302</td>
<td>334</td>
<td>428,952</td>
<td>789,815</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the Datasets.

**Parameter settings.** In our model, we set the maximum number of neighbors \( n \) as 200, and select the dimension of entity/relations embeddings \( m \) from \{50, 75, 100, 150, 200\}, the learning rate \( \eta \) from \{0.001, 0.01, 0.1\}, \( \beta \) from \{0.2, 0.4, 0.6, 0.8, 1\}, \( \lambda \) from \{0.1, 0.5, 1, 1.5, 2\}, \mu_1 from \{0.5, 1, 2, 3, 4\}, \mu_2 from \{0.01, 0.1, 0.5, 0.8, 1, 1.5, 2\}, \gamma from \{0.1, 0.5, 1, 1.5, 2, 2.5\}, the number of head \( K \) from \{1, 2, 4, 6, 8\}. For our model, the best optimal parameter configurations are \( m = 75, \beta = 0.8, \lambda = 1, \mu_1 = 1, \mu_2 = 0.1, \gamma = 2, K = 4, \eta = 0.01 \). For each positive triplet, we select 10 negative triples for training, and set the training epochs as 1000. Following BootEA [Sun et al., 2018], we used 30% of the gold standards as seed alignment while left the remaining as testing data, i.e., the latent aligned entities to discover.

4.2 Baselines

For comparative models, we select current four state-of-the-art methods as baselines in our experiments.

- MTransE [Chen et al., 2017] provides the transformation for entities and relations in different language-specific knowledge graphs.
- IPTTransE [Zhu et al., 2017] incorporates relational paths and learns joint embeddings by sharing parameters based on iteratively aligning entities according to their semantic distance.
- JAPE [Sun et al., 2017] is an attribute-preserving embedding model that incorporates the relation and attribute embeddings for entity alignment.
- BootEA [Sun et al., 2018] adopts the bootstrapping process to label likely alignment as training data and edit alignment during iterations.

In this experiment, we directly copy the experiment results of baseline models reported in their papers since the same datasets are used.

4.3 Multilingual Entity Alignment

The objective of this task is to find the same semantic entities from different languages in knowledge graphs. Comparing with baseline models, we consider two measures as evaluation metrics: (1) Hits@k: the proportion of correct alignment ranked in top-k. Here Hits@1 and Hits@10 are adopted, (2) Mean Reciprocal Rank(MRR): the average of the reciprocal ranks of results. Higher Hits@k and MRR are expected, which indicates better alignment performance.
Table 2: Results on DBP15K and DWY100K.

<table>
<thead>
<tr>
<th>Model</th>
<th>DBP15KZH-EN</th>
<th>DBP15KJA-EN</th>
<th>DBP15KFR-EN</th>
<th>DBPW-D</th>
<th>DBPYG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hits@1</td>
<td>Hits@10</td>
<td>MRR</td>
<td>Hits@1</td>
<td>Hits@10</td>
</tr>
<tr>
<td>MTransE</td>
<td>30.83</td>
<td>61.41</td>
<td>0.364</td>
<td>27.86</td>
<td>57.45</td>
</tr>
<tr>
<td>IPTransE</td>
<td>40.59</td>
<td>73.47</td>
<td>0.516</td>
<td>36.69</td>
<td>69.26</td>
</tr>
<tr>
<td>JAPE</td>
<td>41.18</td>
<td>74.46</td>
<td>0.509</td>
<td>36.25</td>
<td>68.50</td>
</tr>
<tr>
<td>BootEA</td>
<td>62.94</td>
<td>84.75</td>
<td>0.701</td>
<td>62.23</td>
<td>85.39</td>
</tr>
<tr>
<td>NAEA</td>
<td>65.01</td>
<td>86.73</td>
<td>0.720</td>
<td>64.14</td>
<td>87.27</td>
</tr>
</tbody>
</table>

Figure 3: Performance of NAEA on Different Number of Head K.

Table 2 gives the convinced results of entity alignment on DBP15K and DWY100K. From Table 2, we can observe that our model NAEA consistently outperforms all baselines on five data sets. More specifically, NAEA achieves at least 1.76% on Hits@1, 1.81% on Hits@10 and 1.3% on MRR higher performance than other models. We attribute the superiority of our model to its three advantages: (1) Our model devises knowledge embedding component for learning knowledge embeddings and entity alignment component for aligning entities ingeniously. (2) Our model incorporates the neighbor-level and relation-level information of KGs with different weights for multilingual entity alignment. (3) Our model introduces a multi-head attention mechanism to learn the neighbor-level representations by integrating neighbors’ embeddings with a weighted combination.

MTransE [Chen et al., 2017] obtains the worst alignment performance because it learns the embeddings of KGs in different vector spaces, and looses information when modeling the translation between different embedding spaces. IPTransE [Zhu et al., 2017] and JAPE [Sun et al., 2017] achieve better performance than MTransE due to relational path and entity attribute information of KGs are respectively leveraged. Expect for our model NAEA, BootEA has the highest results among baselines since it uses bootstrapping process to accurately label likely alignment as training data, and address the problem caused by the small proportion of prior alignment.

Link prediction in monolingual KG. Effective link prediction can help to improve the completeness of a knowledge graph, and further helps to improve the alignment performance. Link prediction task is to predict the missing head/tail entity given a triplet (h, r, ?)/(? , r, t). Usually two evaluation metrics are used: (1) mean rank of correct entities (Mean Rank), (2) proportion of correct answers ranked in top-10 (Hit@10). In above settings, lower Mean and higher Hit@10 are expected. To testify the effectiveness of neighbor-level information for improving link prediction performance, we compare NAEA with TransE, BootEA on a monolingual KG with the representations obtained by only using individual embedding objective, such as Eq. (2). In this experiment, we randomly select monolingual triplets from DBP15KZH-EN and DBP-WD to organize the training, valid and test set according to ratio 8 : 1 : 1. The results of link prediction on datasets are shown in Table 3, which are that NAEA achieves best performance among baselines on all metrics since NAEA successfully alleviates the neighbor-level information of entities in knowledge graphs. It suggests that the neighbor-level information in KGs can provide indispensable characteristic information of knowledge graphs, and is indeed beneficial to enhance link prediction performance.

4.4 Discussion

In this section, we evaluate how different choices of parameters affect our model’s performance. In the following experiments, except for the parameter being tested, the rest parameters are set as the optimal configurations.

Performance on different K. The above experimental results show that the usage of neighbor-level information through the multi-head attention mechanism is beneficial for improving alignment performance. Thus this subsection mainly explores how the performance of our model changes with different number of head K. In this experiment, we test our model with selecting K from {1, 2, 3, 4, 5, 6}. Figure 3 gives the convinced results. From this figure, we can see that(1) NAEA has the best performance when K = 4, indicating that K = 4 setting best expresses the neighbor-level information of entities, and delivers alignment characteristics of entities in multilingual knowledge graphs. (2) The performance on all datasets begins to gradually rise to the highest
point and then declines as the number of head $K$ grows. This mainly because that too small $K$ can not capture the richer neighbor-level information, and too large $K$ may introduce noisy and lead to over-fitting problem.

**Performance on different $\beta$.** In our model, $\beta$ is to weight the importance of neighbor-level information for multilingual entity alignment. To evaluate the influences of different $\beta$ on alignment performance, we test our model with different $\beta$ selected from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ on Hit@10 metric. We report the experimental results in Figure 4. From Figure 4, we find that (1) Under $\beta = 0.8$ setting, NAEA can achieve the best alignment performance on all datasets, which indicates that such setting can sufficiently reflect the significance of neighbor-level information in aligning entities. (2) The performance of our model first is increasing with the growth of $\beta$ and then drops with $\beta$ further increases, and has the lowest accuracy when $\beta = 0$ or $\beta = 1$. It suggests that the neighbor-level and relation-level information in KGs are both necessary.

## 5 Related Work

### 5.1 Knowledge Representation

In recent years, a series of embedding-based methods such as [Bordes et al., 2013; Wang et al., 2014; Zhu et al., 2019] have been quickly developed for knowledge representation learning. Those methods attempt to encode entities and relations into low-dimension vector spaces while preserving KGs’ properties. Among those methods, the most widely used translation-based method TransE [Bordes et al., 2013] projects entities and relations into a continuous low-dimensional vector space, and treats relation vector as the translation between head and tail entity vectors, i.e., expect $e_h + r \approx e_t$ for a triplet $(e_h, r, e_t)$. TransE is effective and promising for knowledge graph completion, while it has issues in modeling complex relations. Therefore, later works such as [Wang et al., 2014; Lin et al., 2015a; Ji et al., 2015; Yang et al., 2015] are proposed to address the issues of TransE. Also, there exists some non-translation based models, such as [Socher et al., 2013; Nickel et al., 2016; Shi and Weninger, 2017; Dettmers et al., 2018], for learning knowledge graph embedding representations.

Besides above models that use one-step relational path to complete knowledge graphs, many models such as [Garcfa-Durán et al., 2015; Lin et al., 2015b] based on multi-step relational paths are also presented and achieve better performance in knowledge graph embedding.

### 5.2 Knowledge Alignment

Traditional knowledge alignment methods heavily rely on human efforts [Vrandecic, 2012] or well-designed hand-crafted features [Mahdisoltani et al., 2015]. Though achieving better alignment performance, they are time-consuming, labor-expensive and usually suffer from inflexible extension.

Automated knowledge alignment mainly leverages various heterogeneous information in different KGs for knowledge alignment. Some models such as [Wang et al., 2013; Suchanek et al., 2011] make use of external lexicons or Wikipedia links to address the heterogeneity between different KGs. Differing from these models, many embedding-based models only using internal triplet-based information, were proposed for entity alignment. MTransE [Chen et al., 2017] utilizes TransE to embed language-specific KGs into separate embedding spaces, and learns the transition between different spaces. IPTransE [Zhu et al., 2017] uses P-TransE [Lin et al., 2015b] and parameter sharing module to jointly encode entities and relations into a unified semantic spaces. IAPE [Sun et al., 2017] employs semantic structures and attribute correlations of KGs to embed entity and relation embedding representations into an unified embedding spaces. BootEA [Sun et al., 2018] leverages bootstrapping idea to label likely alignment as training data and edit alignment during iterations, which is state-of-the-art entity alignment model.

## 6 Conclusion

This paper presents a neighborhood-aware attentional representation method NAEA for multilingual knowledge graphs, which incorporates the neighbor-level and relation-level information of KGs. NAEA devises knowledge embedding component KE for learning knowledge embeddings and entity alignment component EA for aligning entities. KE introduces an attention mechanism to obtain neighbor-level representation by assigning different importance to entities’ neighbors. EA discovers aligned entities based on the integration of their neighbor-level and relation-level representation with different weights. We empirically conduct experiments on multilingual entity alignment task and monolingual link prediction task with two data sets DBP15K and DB100K. The experimental results show that NAEA significantly and consistently achieves state-of-the-art performance.

### Acknowledgments

This work is supported by National Key R&D Program No.2017YFB0803003, the National Natural Science Foundation of China (No.61202226), MQEPS (No. 96804590) and MQRSG (No. 95109718). We thank all anonymous reviewers for their constructive comments.
References


