Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks

Zhichun Wang, Qingsong Lv, Xiaohan Lan, Yu Zhang
College of Information Science and Technique
Beijing Normal University, Beijing, 100875, China
zcwang@bnu.edu.cn {lqs,lanxh,zybnu}@mail.bnu.edu.cn

Abstract

Multilingual knowledge graphs (KGs) such as DBpedia and YAGO contain structured knowledge of entities in several distinct languages, and they are useful resources for cross-lingual AI and NLP applications. Cross-lingual KG alignment is the task of matching entities with their counterparts in different languages, which is an important way to enrich the cross-lingual links in multilingual KGs. In this paper, we propose a novel approach for cross-lingual KG alignment via graph convolutional networks (GCNs). Given a set of pre-aligned entities, our approach trains GCNs to embed entities of each language into a unified vector space. Entity alignments are discovered based on the distances between entities in the embedding space. Embeddings can be learned from both the structural and attribute information of entities, and the results of structure embedding and attribute embedding are combined to get accurate alignments. In the experiments on aligning real multilingual KGs, our approach gets the best performance compared with other embedding-based KG alignment approaches.

1 Introduction

Knowledge graphs (KGs) represent human knowledge in the machine-readable format, are becoming the important basis of many applications in the areas of artificial intelligence and natural language processing. Multilingual KGs such as DBpedia (Bizer et al., 2009), YAGO (Suchanek et al., 2008; Rebele et al., 2016), and BabelNet (Navigli and Ponzetto, 2012) are especially valuable if cross-lingual applications are to be built. Besides the knowledge encoded in each distinct language, multilingual KGs also contain rich cross-lingual links that match the equivalent entities in different languages. The cross-lingual links play an important role to bridge the language gap in a multilingual KG; however, not all the equivalent entities are connected by cross-lingual links in most multilingual KGs. Therefore, increasingly more research work studies the problem of cross-lingual KG alignment, aiming to match entities in different languages in a multilingual KG automatically.

Traditional cross-lingual KG alignment approaches either rely on machine translation technique or defining various language-independent features to discover cross-lingual links. Most recently, several embedding-based approaches have been proposed for cross-lingual KG alignment, including MTransE (Chen et al., 2017) and JAPE (Sun et al., 2017). Given two KGs and a set of pre-aligned entities of them, embedding-based approaches project entities into low-dimensional vector spaces; entities are matched based on the computations on their vector representations. Following very similar ideas as above, JE (Hao et al., 2016) and ITransE (Zhu et al., 2017) are embedding-based approaches for matching entities between heterogeneous KGs, and they can also work for the problem of cross-lingual KG alignment. The above embedding-based approaches can achieve promising performance without machine translation or feature engineering.

However, we find that the above approaches all try to jointly model the cross-lingual knowledge and the monolingual knowledge in one unified optimization problem. The loss of two kinds of knowledge has to be carefully balanced during the optimization. For example, JE, MTransE, and ITransE all use hyper-parameters to weight the loss of entity alignments in the loss functions of their models; JAPE uses the pre-aligned entities to combine two KGs as one, and adds weight to the scores of negative samples in its loss function. In the above approaches, entities' embeddings have to encode both the structural information in KGs and the equivalent relations of entities. Further-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 349–357
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

more, the attributes of entities (e.g., the age of a people, the population of a country) have not been fully utilized in the existing models. MTransE and ITransE cannot use attributional information in KGs; although JAPE includes the attribute types in the model, the attribute values of entities are ignored. We believe that considering the attribute values can further improve the results of KG alignment.

Having the above observations, we propose a new embedding-based KG alignment approach which directly models the equivalent relations between entities by using graph convolutional networks (GCNs). GCN is a kind of convolutional network which directly operates on graph-structured data; it generates node-level embeddings by encoding information about the nodes' neighborhoods. The adjacencies of two equivalent entities in KGs usually contain other equivalent entities, so we choose GCNs to generate neighborhood-aware embeddings of entities, which are used to discover entity alignments. Our approach can also provide a simple and effective way to include entities' attribute values in the alignment model. More specifically, our approach has the following advantages:

- Our approach uses the entity relations in each KG to build the network structure of GCNs, and it only considers the equivalent relations between entities in model training. Our approach has small model complexity and can achieve encouraging alignment results.

- Our approach only needs pre-aligned entities as training data, and it does not require any pre-aligned relations or attributes between KGs.

- Entity relations and entity attributes are effectively combined in our approach to improve the alignment results.

In the experiments on aligning real multilingual KGs, our approach gets the best performance compared with the baseline methods.

The rest of this paper is organized as follows, Section 2 reviews some related work, Section 3 introduces some background knowledge, Section 4 describes our proposed approach, Section 5 presents the evaluation results, Section 6 is the conclusion and future work.

2 Related Work

2.1 KG Embedding

In the past few years, much work has been done on the problem of KG embedding. KG embedding models embed entities and relations in a KG into a low-dimensional vector space while preserving the original knowledge. The embeddings are usually learned by minimizing a global loss function of all the entities and relations in a KG, which can be further used for relation prediction, information extraction, and some other tasks. TransE is a representative KG embedding approach (Bordes et al., 2013), which projects both entities and relations into the same vector space; if a triple (h, r, t) holds, TransE wants that $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$. The embeddings are learned by minimizing a margin-based ranking criterion over the training set. TransE model is simple but powerful, and it gets promising results on link prediction and triple classification problems. To further improve TransE, several enhanced models based on it have been proposed, including TransR (Lin et al., 2015), TransH (Wang et al., 2014) and TransD (Ji et al., 2015) etc. By introducing new representations of relational translation, later approaches achieve better performance at the cost of increasing model complexity. There are many other KG embedding approaches, recent surveys (Wang et al., 2017; Nickel et al., 2016) give detailed introduction and comparison.

2.2 Embedding-based KG Alignment

Here we introduce the KG Alignment approaches most related to ours, and discuss the main differences between our approach and them.

JE (Hao et al., 2016) jointly learns the embeddings of multiple KGs in a uniform vector space to align entities in KGs. JE uses a set of seed entity alignments to connect two KGs, and then learns the embeddings by using a modified TransE model, which adds a loss of entity alignments in its global loss function.

MTransE (Chen et al., 2017) encodes entities and relations of each KG in a separated embedding space by using TransE; it also provides transitions for each embedding vector to its cross-lingual counterparts in other spaces. The loss function of MTransE is the weighted sum of two component models' loss (i.e., knowledge model and alignment model). To train the alignment model, MTransE needs a set of aligned triples of two KGs.

JAPE (Sun et al., 2017) combines structure embedding and attribute embedding to match entities in different KGs. Structure embedding follows the TransE model, which learns vector representations of entities in the overlay graph of two KGs. Attribute embedding follows the Skip-gram model, which aims to capture the correlations of attributes. To get desirable results, JAPE needs the relations and attributes of two KGs to be aligned in advance.

ITransE (Zhu et al., 2017) is a joint knowledge embedding approach for multiple KGs, which is also suitable for the cross-lingual KG alignment problem. ITransE first learns both entity and relation embeddings following TransE; then it learns to map knowledge embeddings of different KGs into a joint space according to a set of seed entity alignments. ITransE performs iterative entity alignment by using the newly discovered entity alignments to update joint embeddings of entities. ITransE requires all relations being shared among KGs.

The above approaches follow the similar framework to match entities in different KGs. They all rely on TransE model to learn entity embeddings, and then define some kinds of transformation between embeddings of aligned entities. Compared with these approaches, our approach uses an entirely different framework; it uses GCNs to embed entities in a unified vector space, where aligned entities are expected to be as close as possible. Our approach only focuses on matching entities in two KGs, and it does not learn embeddings of relations. MTransE, JAPE, and ITransE all require relations being aligned or shared in KGs; our approach does not need this kind of prior knowledge.

3 Problem Formulation

KGs represent knowledge about real-world entities as triples. Here we consider two kinds of triples in KGs: relational triples, and attributional triples. Relational triples represents relations between entities, and it has the form $\langle entity_1, relation, entity_2 \rangle$. Attributional triples describe attributes of entities, and it has the form $\langle entity, attribute, value \rangle$. For example in the data of YAGO, *graduatedFrom* is a relation, and (*Albert_Einstein, graduatedFrom, ETH_Zurich*) is a relational triple; *diedOnDate* is an attribute, and (*Albert_Einstein, diedOnDate, 1955*) is an attributional triple. Both relational and attributional triples describe important information about entities, we will take both of them into account in the task of cross-lingual KG alignment.

Formally, we represent a KG as $G = (E, R, A, T_R, T_A)$, where E, R, A are sets of entities, relations and attributes, respectively; $T_R \subset E \times R \times E$ is the set of relational triples, $T_A \subset E \times A \times V$ is the set of attributional triples, where V is the set of attribute values.

Let $G_1 = (E_1, R_1, A_1, T_1^R, T_1^A)$ and $G_2 = (E_2, R_2, A_2, T_2^R, T_2^A)$ be two KGs in different languages, and $S = \{(e_{i_1}, e_{i_2}) | e_{i_1} \in E_1, e_{i_2} \in E_2\}_{i=1}^m$ be a set of pre-aligned entity pairs between G_1 and G_2. We define the task of cross-lingual KG alignment as finding new entity alignments based on the existing ones. In multilingual KGs such as DBpedia and YAGO, the cross-lingual links in them can be used to build the sets of pre-aligned entity pairs. The already known entity alignments are used as seeds or training data in the process of KG alignment.

4 The Proposed Approach

The framework of our proposed approach is shown in Figure 1. Given two KGs G_1 and G_2 in different languages, and a set of known aligned entity pairs $S = \{(e_{i_1}, e_{i_2})\}_{i=1}^m$ between them, our approach automatically find new entity alignments based on GCN-based entity embeddings. The basic idea of our approach is to use GCNs to embed entities from different languages into a unified vector space, where equivalent entities are expected to be as close as possible. Entity alignments are predicted by applying a pre-defined distance function to entities' GCN-representations.

4.1 GCN-based Entity Embedding

GCNs (Bruna et al., 2014; Henaff et al., 2015; Defferrard et al., 2016; Kipf and Welling, 2017) are a type of neural network that directly operates on graph data. GCNs allow end-to-end learning of prediction pipelines whose inputs are graphs of arbitrary size and shape. The inputs of a GCN are feature vectors of nodes and the structure of the graph; the goal of a GCN is to learn a function of features on the input graph and produces a node-level output. GCNs can encode information about the neighborhood of a node as a real-valued vector, which was usually used for classification or regression. When solving the problem of KG alignment, we assume that (1) equivalent entities tend

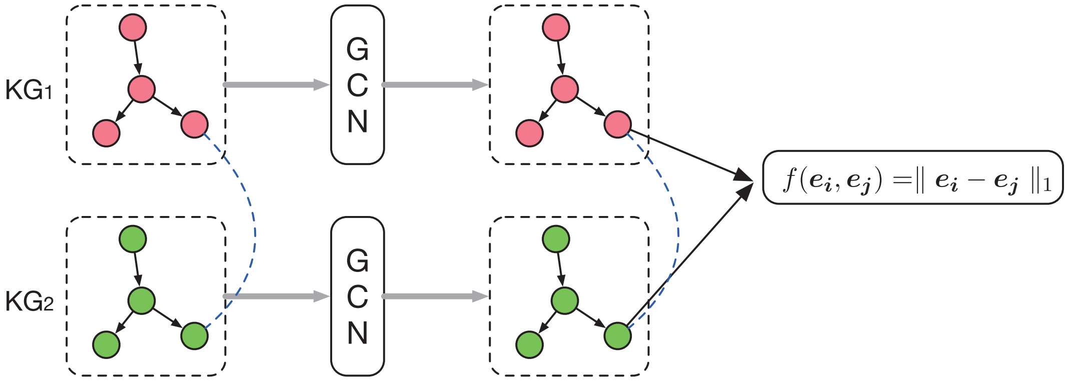

KG1
KG2
GCN
GCN
$f(\boldsymbol{e_i}, \boldsymbol{e_j}) = \| \boldsymbol{e_i} - \boldsymbol{e_j} \|_1$

Figure 1: Framework of our approach (dashed blue lines connects equivalent entities in two KGs)

to have similar attributes, and (2) equivalent entities are usually neighbored by some other equivalent entities. GCNs can combine the attribute information and the structure information together, therefore our approach uses GCNs to project entities into low-dimensional vector space, where equivalent entities are close to each other.

A GCN-model consists of multiple stacked GCN layers. The input to the l-th layer of the GCN model is a vertex feature matrix, $H^{(l)} \in \mathbb{R}^{n \times d^{(l)}}$, where n is the number of vertices and $d^{(l)}$ is the number of features in the l-th layer. The output of the l-th layer is a new feature matrix $H^{(l+1)}$ by the following convolutional computation:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{1}$$

where σ is an activation function; A is a $n \times n$ connectivity matrix that represents the structure information of the graph; $\hat{A} = A + I$, and I is the identity matrix; $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$; $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$ is the weight matrix of the l-th layer in the GCN, $d^{(l+1)}$ is the dimensionality of new vertex features.

Structure and Attribute Embedding. In our approach, GCNs are used to embed entities of two KGs in a unified vector space. To utilize both structure and attribute information of entities, our approach assigns two feature vectors to each entity in GCN layers, structure feature vector h_s and attribute feature vector h_a. In the input layer, $h_s^{(0)}$ is randomly initialized and updated during the training process; $h_a^{(0)}$ is the attribute vectors of entities and it is fixed during the model training. Let H_s and H_a be the structure and attribute feature matrices of all the entities, we redefine the convolutional computation as:

$$\begin{aligned}&[H_s^{(l+1)}; H_a^{(l+1)}] \\ &= \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}[H_s^{(l)}W_s^{(l)}; H_a^{(l)}W_a^{(l)}]\right)\end{aligned} \tag{2}$$

where $W_s^{(l)}$ and $W_a^{(l)}$ are the weight matrices for structure features and attribute features in the l-th layer, respectively; [;] denotes the concatenation of two matrices. The activation function σ is chosen as $ReLU(\cdot) = max(0, \cdot)$.

Model Configuration. More specifically, our approach uses two 2-layer GCNs, and each GCN processes one KG to generate embeddings of its entities. As defined in Section 3, we denote two KGs as $G_1 = (E_1, R_1, A_1, T_1^R, T_1^A)$ and $G_2 = (E_2, R_2, A_2, T_2^R, T_2^A)$; and let their corresponding GCN models be denoted as GCN_1 and GCN_2. As for the structure feature vectors of entities, we set the dimensionality of feature vectors to d_s in all the layers of GCN_1 and GCN_2; and two GCN models share the weight matrices $W_s^{(1)}$ and $W_s^{(2)}$ for the structure features in two layers. As for the attribute vectors of entities, we set the dimensionality of output feature vectors to d_a. Because two KGs may have different number of attributes (i.e. $|A_1| \neq |A_2|$), the dimensionalities of the input attribute feature vectors in two GCN models are different. The first layer of each GCN model transforms the input attribute feature vectors into vectors of size d_a; and two GCN-models generate attribute embeddings of the same dimensionality. Table 1 outlines the parameters of two GCNs in our approach. The final outputs of two GCNs are $(d_s + d_a)$-dimensional embeddings of entities, which are further used to discover entity alignments.

Parameter	GCN_1	GCN_2
Initial Structure Feature Matrices	$H_{s1}^{(0)} \in \mathbb{R}^{\lvert E_1 \rvert \times d_s}$	$H_{s2}^{(0)} \in \mathbb{R}^{\lvert E_2 \rvert \times d_s}$
Weight Matrix for Structure Features in Layer 1	$W_s^{(1)} \in \mathbb{R}^{d_s \times d_s}$	
Weight Matrix for Structure Features in Layer 2	$W_s^{(2)} \in \mathbb{R}^{d_s \times d_s}$	
Output Structure Embeddings	$H_{s1}^{(2)} \in \mathbb{R}^{\lvert E_1 \rvert \times d_s}$	$H_{s2}^{(2)} \in \mathbb{R}^{\lvert E_2 \rvert \times d_s}$
Initial Attribute Feature Matrices	$H_{a1}^{(0)} \in \mathbb{R}^{\lvert E_1 \rvert \times \lvert A_1 \rvert}$	$H_{a2}^{(0)} \in \mathbb{R}^{\lvert E_2 \rvert \times \lvert A_2 \rvert}$
Weight Matrix for Attribute Features in Layer 1	$W_{a1}^{(1)} \in \mathbb{R}^{\lvert A_1 \rvert \times d_a}$	$W_{a2}^{(1)} \in \mathbb{R}^{\lvert A_2 \rvert \times d_a}$
Weight Matrix for Attribute Features in Layer 2	$W_a^{(2)} \in \mathbb{R}^{d_a \times d_a}$	
Output Attribute Embeddings	$H_{a1}^{(2)} \in \mathbb{R}^{\lvert E_1 \rvert \times d_a}$	$H_{a2}^{(2)} \in \mathbb{R}^{\lvert E_2 \rvert \times d_a}$

Table 1: Parameters of two GCNs

Computation of Connectivity Matrix. In a GCN model, the connectivity matrix A defines the neighborhoods of entities in the convolutional computation. For an undirected graph, the adjacency matrix can be directly used as A_s. But KGs are relational multi-graphs, entities are connected by typed relations. Therefore, we design a particular method for computing A of a KG; we let $a_{ij} \in A$ indicate to what extent the information of alignments propagates from the i-th entity to the j-th entity. The probability of two entities being equivalent differs greatly considering they connect to aligned entities by different relations (e.g., *hasParent* vs. *hasFriend*). Therefore, we compute two measures, which are called functionality and inverse functionality, for each relation:

$$fun(r) = \frac{\#Head\_Entities\_of\_r}{\#Triples\_of\_r} \tag{3}$$

$$ifun(r) = \frac{\#Tail\_Entities\_of\_r}{\#Triples\_of\_r} \tag{4}$$

where $\#Triples\_of\_r$ is the number of triples of relation r; $\#Head\_Entities\_of\_r$ and $\#Tail\_Entities\_of\_r$ are the numbers of head entities and tail entities of r, respectively. To measure the influence of the i-th entity over the j-the entity, we set $a_{ij} \in A$ as:

$$a_{ij} = \sum_{\langle e_i, r, e_j \rangle \in G} ifun(r) + \sum_{\langle e_j, r, e_i \rangle \in G} fun(r) \tag{5}$$

4.2 Alignment prediction

Entity alignments are predicted based on the distances between entities from two KGs in the GCN-representation space. For entities e_i in G_1 and v_j in G_2, we compute the following distance measure between them:

$$\begin{aligned} D(e_i, v_j) =& \beta \frac{f(\boldsymbol{h_s}(e_i), \boldsymbol{h_s}(v_j))}{d_s} \\ &+ (1-\beta) \frac{f(\boldsymbol{h_a}(e_i), \boldsymbol{h_a}(v_j))}{d_a} \end{aligned} \tag{6}$$

where $f(\boldsymbol{x}, \boldsymbol{y}) = \parallel \boldsymbol{x} - \boldsymbol{y} \parallel_1$, $\boldsymbol{h_s}(\cdot)$ and $\boldsymbol{h_a}(\cdot)$ denote the structure embedding and attribute embedding of an entity, respectively; d_s and d_a are dimensionalities of structure embeddings and attribute embeddings; β is a hyper-parameter that balances the importance of two kinds of embeddings.

The distance is expected to be small for equivalent entities and large for non-equivalent ones. For a specific entity e_i in G_1, our approach computes the distances between e_i and all the entities in G_2, and returns a list of ranked entities as candidate alignments. The alignment can be also performed from G_2 to G_1. In the experiments, we report the results of both directions of KG alignment.

4.3 Model Training

To enable GCNs to embed equivalent entities as close as possible in the vector space, we use a set of known entity alignments S as training data to train GCN models. The model training is performed by minimizing the following margin-based ranking loss functions:

$$\begin{aligned} L_s = \sum_{(e,v) \in S} \sum_{(e',v') \in S'_{(e,v)}} & [f(\boldsymbol{h_s}(e), \boldsymbol{h_s}(v)) + \gamma_s \\ & - f(\boldsymbol{h_s}(e'), \boldsymbol{h_s}(v')]_+ \end{aligned} \tag{7}$$

$$L_a = \sum_{(e,v)\in S} \sum_{(e',v')\in S'_{(e,v)}} [f(\boldsymbol{h_a}(e), \boldsymbol{h_a}(v)) + \gamma_a - f(\boldsymbol{h_a}(e'), \boldsymbol{h_a}(v'))]_+ \tag{8}$$

where $[x]_+ = \max\{0, x\}$, $S'_{(e,v)}$ denotes the set of negative entity alignments constructed by corrupting (e, v), i.e. replacing e or v with a randomly chosen entity in G_1 or G_2; $\gamma_s, \gamma_a > 0$ are margin hyper-parameters separating positive and negative entity alignments. L_s and L_a are loss functions for structure embedding and attribute embedding, respectively; they are independent of each other and hence are optimized separately. We adopt stochastic gradient descent (SGD) to minimize the above loss functions.

5 Experiment

5.1 Datasets

We use the DBP15K datasets in the experiments, which were built by Sun et al. (2017). The datasets were generated from DBpedia, a large-scale multilingual KG containing rich inter-language links between different language versions. Subsets of Chinese, English, Japanese and French versions of DBpedia are selected following certain rules. Table 2 outlines the detail information of the datasets. Each dataset contains data two KGs in different languages and 15 thousand inter-language links connecting equivalent entities in two KGs. In the experiments, the known equivalent entity pairs are used for model training and testing.

5.2 Experiment Settings

In the experiments, we compared our approach with JE, MTransE and JAPE. We also build JAPE′, a variant of JAPE which does not use pre-aligned relations and attributes. Because the approach ITransE performs iterative alignment and it requires two KGs sharing the same relations, we do not include it in the comparison. The inter-language links in each dataset are used as the gold standards of entity alignments. For all the compared approaches, we use 30% of inter-language links for training and 70% of them for testing; the split of training and testing are the same for all approaches. We use $Hits@k$ as the evaluation measure to assess the performance of all the approaches. $Hits@k$ measures the proportion of correctly aligned entities ranked in the top k candidates. For the parameters of our approach, we set $d_s = 1,000$, $d_a = 100$; the margin $\gamma_s = \gamma_a = 3$ in the loss function, and β in the distance measure is emperically set to 0.9.

5.3 Results

Table 3 shows the results of all the compared approaches on DBP15K datasets. We report $Hits@1$, $Hits@10$ and $Hits@50$ of approaches on each dataset. Because we use the same datasets as in (Sun et al., 2017), the results of JE, MTransE, and JAPE are obtained from (Sun et al., 2017). For JAPE and JAPE′, each of them has three variants: Structure Embedding without negative triples (SE w/o neg.), Structure Embedding (SE), Structure and attribute joint embedding (SE+AE). We use GCN(SE) and GCN(SE+AE) to denote two variants of our approach: one only uses relational triples to perform structure embedding, and the other uses both relational and attributional triples to perform structure and attribute embedding.

GCN(SE) vs. GCN(SE+AE)

We first compare the results of GCN(SE) and GCN(SE+AE) to see whether the attributional information is helpful in the KG alignment task. According to the results, adding attributes in our approach do lead to slightly better results. The improvements range from 1% to 10%, which are very similar to the improvements of JAPE(SE) over JAPE(SE+AE). It shows that the KG alignment mainly relays on the structural information in KGs, but the attributional information is still useful. Our approach uses the same framework for embedding structure and attribute information, the combination of two kinds of embeddings works effectively.

GCN(SE+AE) vs. Baselines

On the dataset of DBP15K$_{ZH-EN}$, JAPE(SE+AE) performs best and gets five best $Hits@k$ values; our approach GCN(SE+AE) gets the best $Hits@1$ in the alignment direction of ZH→EN. The results of GCN(SE+AE) and JAPE gets very close results regarding $Hits@1$ and $Hits@10$ in the direction of ZH→EN. In the alignment direction of EN→ZH, JAPE(SE+AE) outperforms GCN(SE+AE) by about 2-3%. But it should be noticed that JAPE uses additional aligned relations and attributes as its inputs,

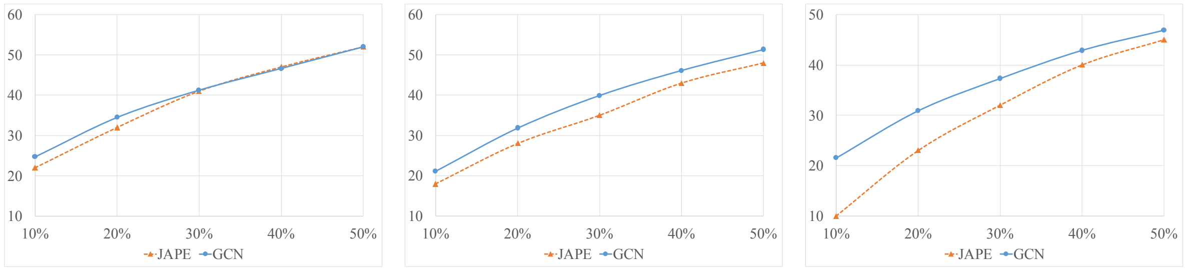
Datasets		Entities	Relations	Attributes	Rel. triples	Attr. triples
$DBP15K_{ZH-EN}$	Chinese	66,469	2,830	8,113	153,929	379,684
	English	98,125	2,317	7,173	237,674	567,755
$DBP15K_{JA-EN}$	Japanese	65,744	2,043	5,882	164,373	354,619
	English	95,680	2,096	6,066	233,319	497,230
$DBP15K_{FR-EN}$	French	66,858	1,379	4,547	192,191	528,665
	English	105,889	2,209	6,422	278,590	576,543

Table 2: Details of the datasets

$DBP15K_{ZH-EN}$		$ZH \rightarrow EN$			$EN \rightarrow ZH$		
		Hits@1	Hits@10	Hits@50	Hits@1	Hits@10	Hits@50
*JE		21.27	42.77	56.74	19.52	39.36	53.25
*MTransE		30.83	61.41	79.12	24.78	52.42	70.45
*JAPE	SE w/o neg.	38.34	68.86	84.07	31.66	59.37	76.33
	SE	39.78	72.35	87.12	32.29	62.79	80.55
	SE + AE	41.18	**74.46**	**88.90**	**40.15**	**71.05**	**86.18**
JAPE$'$	SE w/o neg.	30.10	62.58	80.28	23.04	52.91	72.17
	SE	30.54	66.41	83.94	23.91	57.02	77.31
	SE + AE	33.32	69.28	86.40	33.02	66.92	85.15
GCN	SE	38.42	70.34	81.24	34.43	65.68	77.03
	SE + AE	**41.25**	74.38	86.23	36.49	69.94	82.45
$DBP15K_{JA-EN}$		$JA \rightarrow EN$			$EN \rightarrow JA$		
		Hits@1	Hits@10	Hits@50	Hits@1	Hits@10	Hits@50
*JE		18.92	39.97	54.24	17.80	38.44	52.48
*MTransE		27.86	57.45	75.94	23.72	49.92	67.93
*JAPE	SE w/o neg.	33.10	63.90	80.80	29.71	56.28	73.84
	SE	34.27	66.39	83.61	31.40	60.80	78.51
	SE + AE	36.25	68.50	85.35	38.37	67.27	82.65
JAPE$'$	SE w/o neg.	28.90	60.61	80.03	25.34	53.36	71.94
	SE	29.35	63.31	82.76	26.37	57.35	76.87
	SE + AE	31.06	64.11	81.57	32.45	62.21	79.08
GCN	SE	38.21	72.49	82.69	36.90	68.50	79.51
	SE + AE	**39.91**	**74.46**	**86.10**	**38.42**	**71.81**	**83.72**
$DBP15K_{FR-EN}$		$FR \rightarrow EN$			$EN \rightarrow FR$		
		Hits@1	Hits@10	Hits@50	Hits@1	Hits@10	Hits@50
*JE		15.38	38.84	56.50	14.61	37.25	54.01
*MTransE		24.41	55.55	74.41	21.26	50.60	69.93
*JAPE	SE w/o neg.	29.55	62.18	79.36	25.40	56.55	74.96
	SE	29.63	64.55	81.90	26.55	60.30	78.71
	SE + AE	32.39	66.68	83.19	32.97	65.91	82.38
JAPE$'$	SE w/o neg.	28.23	60.99	78.47	24.68	55.25	74.19
	SE	27.58	62.03	79.98	24.93	58.95	77.79
	SE + AE	30.21	65.81	82.57	31.42	63.86	80.95
GCN	SE	36.51	73.42	85.93	36.08	72.37	85.44
	SE + AE	**37.29**	**74.49**	**86.73**	**36.77**	**73.06**	**86.39**

Table 3: Results comparison of cross-lingual KG alignment (* marks the results obtained from (Sun et al., 2017))

while our approach does not use these kinds of prior knowledge. If compared with JAPE$'$, GCN(SE+AE) performs better than it regrading $Hits@1$ and $Hits@10$. While compared with



(a) ZH-EN (b) JA-EN (c) FR-EN

Figure 2: GCN and JAPE using different sizes of training data (horizontal coordinates: proportions of pre-aligned entities used in training data; vertical coordinates: $Hits@1$)

JE and MTransE, GCN(SE+AE) outperform them by more than 10% in most cases. On the datasets of DBP15K$_{JA-EN}$ and DBP15K$_{FR-EN}$, GCN(SE+AE) outperforms all the compared approaches regarding all the $Hits@k$ measures. Even compared with JAPE which uses extra relation and attribute alignments, GCN(SE+AE) still gets better results than it.

Comparing with all the baselines, both GCN(SE) and GCN(SE+AE) outperform JE and MTransE significantly. Among all the baselines, JAPE is the strongest one; it might due to its ability of using both relational and attributional triples, and the extra alignments of relations and attributes that it consumes. Our approach achieves better results than JAPE on two datasets; Although JAPE performs better than our approach, the differences between their results are small. If there are no existing relation and attribute alignments between two KGs, our approach will have distinct advantage over JAPE.

GCN vs. JAPE using different sizes of training data

To investigate how the size of training set affects the results of our approach, we further compare our approach with JAPE by using different number of pre-aligned entities as training data. For JAPE, the pre-aligned entities are used as seeds to make their vectors overlapped. In our approach, all the pre-aligned entities are used to train GCN models. Intuitively, the more pre-aligned entities used, the better results should be obtained by both GCN and JAPE.

Here we use different proportions of pre-aligned entities as training data, which ranges 10% to 50% with step 10%; all the rest of pre-aligned entities are used for testing. Figure 2 shows the $Hits@1$ of two approaches in three datasets. It shows that both approaches perform better as the size of training data increases. And our approach always outperforms JAPE except using 40% pre-aligned entities as training data in Figure 2(a). Especially in the tasks of aligning Japanese to English and French to English, our approach has a distinct advantage over JAPE.

6 Conclusion and Future Work

This paper presents a new embedding-based KG alignment approach which discovers entity alignments based on the entity embeddings learned by GCNs. Our approach can make use of both the relational and the attributional triples in KGs to discover the entity alignments. We evaluate our method on the data of real multilingual KGs, and the results show the advantages of our approach over the compared baselines.

In the future work, we will explore more advanced GCN models for KG alignment task, such as Relational GCNs (Schlichtkrull et al., 2017) and Graph Attention Networks (GATs) (Velickovic et al., 2017). Furthermore, how to iteratively discover new entity alignments in the framework of our approach is another interesting direction that we will study in the future.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (No. 61772079) and the National Key R&D Program of China (No. 2017YFC0804004).

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics:*

science, services and agents on the world wide web, 7(3):154–165.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of Advances in neural information processing systems (NIPS2013)*, pages 2787–2795.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of International Conference on Learning Representations (ICLR2014)*.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (AAAI2017)*, pages 1511–1517.

Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings Advances in Neural Information Processing Systems (NIPS2016)*, pages 3844–3852.

Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint embedding method for entity alignment of knowledge bases. In *Proceedings of China Conference on Knowledge Graph and Semantic Computing (CCKS2016)*, pages 3–14.

Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. 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, pages 687–696.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations (ICLR2017)*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI2015)*, volume 15, pages 2181–2187.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Asia Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *Proceedings of the Fifteenth International Semantic Web Conference (ISWC2016)*, pages 177–185.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of the Sixteenth International Semantic Web Conference (ISWC2017)*, pages 628–644.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *CoRR*, abs/1710.10903.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-eighth AAAI Conference on Artificial Intelligence (AAAI2014)*, volume 14, pages 1112–1119.

Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*, pages 4258–4264.