Iterative Entity Alignment with Improved Neural Attribute Embedding

Ning Pang[1], Weixin Zeng[1], Jiuyang Tang[1,2], Zhen Tan[1], and Xiang Zhao[1,2] ✉

[1] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China
[2] Collaborative Innovation Center of Geospatial Technology, Wuhan, China
xiangzhao@nudt.edu.cn

Abstract. Entity alignment (EA) aims to detect equivalent entities in different knowledge graphs (KGs), which can facilitate the integration of knowledge from multiple sources. Current EA methods usually harness KG embeddings to project entities in various KGs into the same low-dimensional space, where equivalent entities are placed close to each other. Nevertheless, most methods fail to take fully advantage of other sources of information, e.g., attribute information, and overlook the negative impact brought by lack of labelled data. To overcome these deficiencies, in this paper, we propose to generate neural attribute representation by considering both local and global signals. Besides, entity representations are refined via an iterative training process on the neural network. We evaluate our proposal on real-life datasets against state-of-the-art methods, and the results demonstrate the effectiveness of our solution.

Keywords: Entity alignment · Attribute information · Iterative training.

1 Introduction

Knowledge graphs (KGs) are becoming increasingly important for many downstream applications such as question answering [1] and sentence generation [2]. A large number of KGs, e.g., YAGO and DBpedia, have been constructed. However, in reality, these KGs are far from complete. To tackle this problem, various methods have been proposed, among which KG alignment attracts growing attention since it can incorporate complementary knowledge from multiple external KGs. Unfortunately, KGs are usually built in different natural languages or with various ontology systems, resulting in the obstacle of integrating knowledge from external KGs to refine the target KG. As thus, many research works have been devoted to improving the performance of KG alignment.

Current KG alignment approaches lay emphasis on entity alignment (EA), as entities are the pivots connecting different KGs. The task of EA aims to identify equivalent entities in different KGs. State-of-the-art methods [3, 4] normally harness translation-based KG embeddings to project entities and relations into a low-dimensional embedding space. The separated embedding spaces are then unified by harnessing seed entity pairs. Eventually given a target entity, its counterparts in other KGs can be determined in accordance to the distance in the unified embedding space. Nevertheless, Wang et al. [5] argued that KG embedding might fail to fully mine the structural information and instead they

utilize graph convolutional network (GCN) [6] to generate entity embeddings. Additionally, they proposed to incorporate attribute information to serve as additional signals for EA. Due to the limitation of dataset, attribute names are considered instead of attribute values. Their method has also achieved superior results on existing EA benchmarks.

In [5], attribute names are represented as one-hot embeddings of the most frequent attributes. However, the most frequent attributes appear with the majority of entities and are not able to help identify a specific entity. Additionally, the neighbourhood attribute information is completely ignored. For instance, to determine the equivalent entity of entity `Michael_Jordan`, optional attribute *hasSpouse* would be more useful than obligatory attribute *birthDate* since every person has a birthday while not necessarily a spouse. Besides, the attributes of `Michael_Jordan`'s neighbouring entities, e.g., *hasNBAChampionship* for `Chicago_Bulls`, can also be harnessed for representing `Michael_Jordan`. Also, the shortage of labelled data (seed entity pairs) is largely overlooked by previous works, which will restrain the quality of entity embeddings, and hence, the performance of EA.

In this paper, to handle these drawbacks, we devise an iterative entity alignment method with improved neural attribute embedding, Inga, which enhances EA performance by harnessing neural network, i.e., GCN, iterative training strategy and refined attribute information to generate entity representations. In specific, by incorporating the neighbouring attributes of an entity (local attribute information) and the frequency of an attribute (global attribute information) to form the improved attribute feature vector, more comprehensive signals can be captured in comparison to the one-hot representation [5]. To deal with the second limitation, an iterative training strategy is utilized to train GCN, which keeps labelling unlabelled instances and select high-quality ones to retrain itself so as to generate better entity embeddings.

The main contributions of this work are:

- Attribute representation is improved by considering both local and global information.
- We apply an iterative training mechanism on GCN to generate more accurate structure and attribute representations.
- We evaluate Inga against state-of-the-art methods on three cross-lingual EA datasets, and the results demonstrate the effectiveness of our proposal.

Related Works. The task of KG alignment can be traced back to traditional ontology matching task [7]. With the emergence and prevalence of embedding techniques, most KG alignment solutions resort to KG embedding for determining equivalent elements in different KGs. Chen et al. [3] (MTransE) are the first to utilize TransE to embed entities in each KG into separated embedding spaces, which are then unified by different alignment models using seed entities pairs. The distance in the unified embedding space is used to determine entity pairs. JAPE [4] introduces attribute type information for refining structure representation captured by KG embedding. GCN [5], on the other hand, harnesses GCN, instead of KG embedding, to generate entity representation. Attribute information, represented as one-hot vectors of most frequent attributes, is also utilized to complement structure information.

2 Methodology

Task Definition. A KG is usually represented as $G = (E, R, A, V)$, where E, R, A, V denotes entities, relations, attributes and attribute values respectively. Given two KGs, G_1 and G_2, EA aims to automatically mine new aligned entity pairs based on existing seed entity pairs $S = \{(e_{i1}, e_{i2})|e_{i1} \in E_1, e_{i2} \in E_2\}_{i=1}^{m}$.

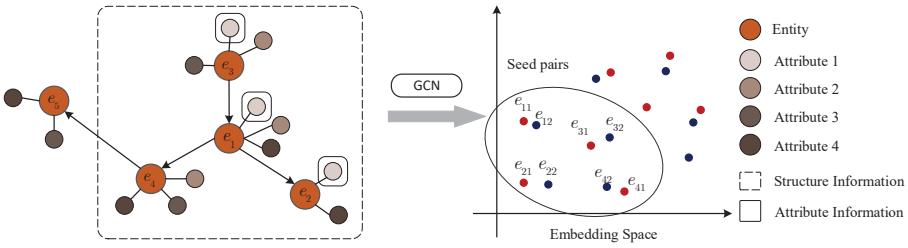

Fig. 1. The framework of our model. Dashed-line rectangle represents the structure information of e_1. The solid-line rectangle represents the attribute information of e_1 with respect to attribute 4. GCN embeds entities into a unified embedding space.

Structure and Attribute Embedding. Equivalent entities in multiple KGs are assumed to have similar neighbours (structure information) and attribute names (attribute information). To capture these information, GCN is utilized to operate on KGs and produce node-level embeddings for all entities. An entity's structure information can be represented by $\mathbf{x}_s$, as thus, the matrix encoding structure information of all entities is denoted by $\mathbf{X}_s$, which is randomly initialized and updated during model training in our setting. Similarly, the attribute feature of an entity can be represented by a vector $\mathbf{x}_a$, and the corresponding attribute feature matrix for all entities is $\mathbf{X}_a$. The initial attribute matrix is pre-computed, as detailed in the following. Note that following previous works, here we focus on attribute names, instead of attribute values.

In [5], attribute information is converted into a k-dimension one-hot vector encoding k most frequent attributes. Nonetheless, this setting fails to differentiate entities or consider the neighboring information. In our model, the most frequent attributes (which we define as attributes appearing with more than 80% of entities) are discarded for representing an entity since they appear with many entities and are not discriminative. Among the rest of attributes, we select k most frequent ones as they can better distinguish entities and are not too long-tail (which might result in very sparse attribute matrix). For an entity, its attribute feature vector can be denoted by $\mathbf{x}_a = [x_a^1, x_a^2, ..., x_a^k]$, where

$$x_a^i = \frac{n_i}{\sum_{j=1}^{k} n_j}. \tag{1}$$

n_i is the total times of i-th attribute appearing among the attributes of an entity and its one-hop neighbours, which is harnessed to capture local attribute information. As thus, both local and global attribute information can be encoded.

The inputs of GCN model include $\mathbf{X}_s$, $\mathbf{X}_a$, and adjacency matrix $\mathbf{A}$. By feeding the inputs into GCN model, the output entity embedding matrix can be obtained:

$$[\mathbf{C}_s; \mathbf{C}_a] = GCN(\mathbf{A}, [\mathbf{X}_s; \mathbf{X}_a]), \tag{2}$$

where $[;]$ denotes the concatenation of two matrices, $\mathbf{C}_s \in \mathbb{R}^{N \times d_s}$ is the final structure embedding matrix and $\mathbf{C}_a \in \mathbb{R}^{N \times d_a}$ represents the final attribute embedding matrix. In our model, we harness two 2-layer GCNs to generate embeddings for entities in two KGs respectively. The dimensionalities of structure and attribute feature vectors are set to d_s and d_a for all layers in respective models.

Distance Function. A weighted distance function, which combines structure embedding and attribute embedding, is designed for entity alignment prediction. Concretely, for $e_{i1} \in G_1$ and $e_{i2} \in G_2$, the distance can be calculated by:

$$Dis(e_{i1}, e_{i2}) = \theta Dis_s(e_{i1}, e_{i2}) + (1 - \theta) Dis_a(e_{i1}, e_{i2}), \tag{3}$$

where θ is a hyper-parameter balancing the importance of structure embedding distance and attribute embedding distance. The structure (attribute) embedding distance is defined as the vector norm of $\mathbf{c}_s^{i1} - \mathbf{c}_s^{i2}(\mathbf{c}_a^{i1} - \mathbf{c}_a^{i2})$ divided by the dimensionality $d_s(d_a)$. The distance between equivalent entities is expected to be as small as possible. As thus, the entity in G_2 with the smallest distance from a specific entity $e_{i1} \in G_1$ can be regraded as the counterpart of e_{i1}.

Loss Function. We use pre-aligned entity pairs S to train GCN models. The training objectives for learning structure embedding and attribute embedding are to minimize the following margin-based ranking loss functions,

$$\mathcal{J}_s = \sum_{(e_1,e_2)\in S} \sum_{(v_1,v_2)\in S^-} d_s \cdot [Dis_s(e_1, e_2) - Dis_s(v_1, v_2) + \gamma_s]_+, \tag{4}$$

$$\mathcal{J}_a = \sum_{(e_1,e_2)\in S} \sum_{(v_1,v_2)\in S^-} d_a \cdot [Dis_a(e_1, e_2) - Dis_a(v_1, v_2) + \gamma_a]_+, \tag{5}$$

where $[x]_+ = max\{0, x\}$, S^- denotes the set of negative aligned entity pairs; γ_s and γ_a are two positive margins separating positive and negative aligned entity pairs. Loss functions $\mathcal{J}_s$ and $\mathcal{J}_a$ are optimized by stochastic gradient descent (SGD) separately.

Iterative Training. Considering the lack of labelled data, inspired by [8], we adopt semi-supervised training strategy to enlarge the training set iteratively by including aligned pairs with high confidence during training process.

Once newly-aligned entity pairs are added into S, they are considered as valid training data. However, some false positive pairs may be included, which will hurt the following training process. Consequently, the key challenge is how to choose highly confident samples from newly-aligned entity pairs to enlarge S. In consequence, we consider candidate entity pairs $\{(e_{i1}, e_{j2}) | e_{i1} \in G_1 \backslash S_1, e_{j2} \in G_2 \backslash S_2\}$, which satisfy $e_{i1} = \arg\min Dis(\_, e_{j2}), e_{j2} = \arg\min Dis(e_{i1}, \_)$, as reliable aligned pairs for iterative training, where S_1 and S_2 are the set of pre-aligned entities in G_1 and G_2 respectively.

3 Experiment

Datasets. We adopt the widely used DBP15K datasets in the experiments, which were developed by [4]. The datasets were constructed from subsets of DBpedia, which has multiple versions in different languages. DBP15K consists of three datasets, Chinese-English (Zh-En), Japanese-English (Ja-En), and French-English (Fr-En). In each dataset, there are 15 thousand already-known equivalent entity pairs, 30% of which are used for training and 70% of which are for testing.

Parameter Settings. In our GCN models, the dimensionality of structure embedding and attribute embedding in all layers were set to $d_s = 300$ and $d_a = 600$ respectively. The number of top attributes k is set to 1000. The iterative training processing would not stop until the size of the newly-included set $|C|$ is under a threshold $\alpha = 100$. The margins γ_s and γ_a are set to 3. The hyper-parameter θ in weighted distance function is set to 0.9.

Competing Approaches and Evaluation Metric. Three approaches are utilized for comparison, including MTransE [3], JAPE [4], and GCN [5]. The evaluation metric, $Hits@k$, measures the proportion of correctly aligned entities in top k ranked candidates. We report the results of $Hits@1$ (accuracy), $Hits@10$, and $Hits@50$ in the experiment.

Table 1. Experimental Results

	$Zh-En$			$En-Zh$		
	$Hits@1$	$Hits@10$	$Hits@50$	$Hits@1$	$Hits@10$	$Hits@50$
MTransE	30.83	61.41	79.12	24.78	52.42	70.45
JAPE	41.18	74.46	88.9	40.15	71.05	86.18
GCN	41.25	74.38	86.23	36.49	69.94	82.45
Inga	**50.45**	**79.42**	**89.79**	**49.36**	**76.05**	**86.38**
	$Ja-En$			$En-Ja$		
	$Hits@1$	$Hits@10$	$Hits@50$	$Hits@1$	$Hits@10$	$Hits@50$
MTransE	27.86	57.45	75.94	23.72	49.92	67.93
JAPE	36.25	68.5	85.35	38.37	67.27	82.65
GCN	39.91	74.46	86.1	38.42	71.81	83.72
Inga	**51.46**	**79.46**	**88.25**	**51.05**	**77.04**	**86.27**
	$Fr-En$			$En-Fr$		
	$Hits@1$	$Hits@10$	$Hits@50$	$Hits@1$	$Hits@10$	$Hits@50$
MTransE	24.41	55.55	74.41	21.26	50.6	69.93
JAPE	32.39	66.68	83.19	32.97	65.91	82.38
GCN	37.29	74.49	86.73	36.77	73.06	86.39
Inga	**50.45**	**79.42**	**87.79**	**49.36**	**76.05**	**86.48**

Experiment Results. The experimental results of Inga and three competitors on DBP15K datasets are shown in Table 1. It can be easily observed that

Inga achieves the best performance among most settings on three bi-directional datasets.

Among the four approaches, MTransE achieves relatively worse results. The $Hits@1$ values of MTransE on all datasets are between 20% and 30%, indicating that translation-based KG embeddings can capture structure information and serve as useful signals for EA. Another KG embedding based method, JAPE, outperforms MTransE significantly by over 10% in most cases due to its ability to incorporate attribute information for refining entity structure embeddings. GCN attains slightly better results than JAPE on $Ja-En$ and $Fr-En$ language pairs, indicating the effectiveness of GCN model for generating structure representation. Inga is built on the architecture of GCN, whereas it improves the results by a large margin. In both alignment directions, Inga outperforms GCN and JAPE by about $3\% - 12\%$ regarding all $Hits@k$ metrics. This demonstrates the usefulness of the improved attribute feature representation and iterative training strategy.

Noteworthy is that the gap between Inga and the rest approaches is much larger on $Hits@1$ (accuracy) than other metrics. This reveals that Inga can align more *accurate* entity pairs, which is critical to EA task.

4 Conclusion

In this paper, we propose a GCN-based model to align entities in different KGs by projecting entities into a unified embedding space, where equivalent entities are placed close to each other. Attribute representation is improved by capturing more informative attribute features. Furthermore, we devise an iterative training strategy to enlarge training set and generate better entity embeddings via neural network. Our proposal is then evaluated on real-life datasets and the results demonstrate that our model outperforms three state-of-the-art competitors by a large margin. For further work, to take more information especially attribute values as guidance for our model is also necessary.

Acknowledgements. This work was partially supported by NSFC under grants Nos. 61872446, 61876193 and 71690233.

References

1. J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li. Neural generative question answering. In *Proceedings of IJCAI*, pages 2972–2978, 2016.
2. B. D. Trisedya, J. Qi, R. Zhang, and W. Wang. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of ACL*, pages 1627–1637, 2018.
3. M. Chen, Y. Tian, M. Yang, and C. Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*, pages 1511–1517, 2017.
4. Z. Sun, W. Hu, and C. Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of ISWC, Part I*, pages 628–644, 2017.
5. Z. Wang, Q. Lv, X. Lan, and Y. Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of EMNLP*, pages 349–357, 2018.
6. T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.
7. F. M. Suchanek, S. Abiteboul, and P. Senellart. PARIS: probabilistic alignment of relations, instances, and schema. *PVLDB*, 5(3):157–168, 2011.
8. H. Zhu, R. Xie, Z. Liu, and M. Sun. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of IJCAI*, pages 4258–4264, 2017.