Global Structure and Local Semantics-Preserved Embeddings for Entity Alignment

Hao Nie[1,3], **Xianpei Han**[1,2 *], **Le Sun**[1,2],
Chi Man Wong[4], **Qiang Chen**[4], **Suhui Wu**[4] and **Wei Zhang**[4]
[1]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
[2]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Alibaba Group
{niehao2016, xianpei, sunle}@iscas.ac.cn,
{chiman.wcm, lapu.cq, linnai.wsh, lantu.zw}@alibaba-inc.com

Abstract

Entity alignment (EA) aims to identify entities located in different knowledge graphs (KGs) that refer to the same real-world object. To learn the entity representations, most EA approaches rely on either translation-based methods which capture the local relation semantics of entities or graph convolutional networks (GCNs), which exploit the global KG structure. Afterward, the aligned entities are identified based on their distances. In this paper, we propose to jointly leverage the global KG structure and entity-specific relational triples for better entity alignment. Specifically, a global structure and local semantics preserving network is proposed to learn entity representations in a coarse-to-fine manner. Experiments on several real-world datasets show that our method significantly outperforms other entity alignment approaches and achieves the new state-of-the-art performance.

1 Introduction

KGs, such as DBpedia [Lehmann *et al.*, 2015] and YAGO [Rebele *et al.*, 2016], store structured knowledge as triples of the form `<head, relation, tail>`, which can support many natural language processing applications, e.g., question answering [Zhang *et al.*, 2018] and information extraction [Hoffmann *et al.*, 2011]. Unfortunately, KGs are usually separately constructed and often contain different but complementary knowledge, thus it is essential to merge different KGs into a unified one. Entity alignment (EA) [Hao *et al.*, 2016; Trisedya *et al.*, 2019; Cao *et al.*, 2019], also known as Entity Resolution (ER) [Nie *et al.*, 2019; Fu *et al.*, 2019] in database community, is such a technique that aims to find entities in different data sources that refer to the same real-world object. For example, by aligning entities "CA" and "California" from triples <Apple Inc., locatedIn, CA> and <California, country, USA>, we can obtain additional knowledge <Apple Inc., locatedIn, USA>.

*Corresponding author.

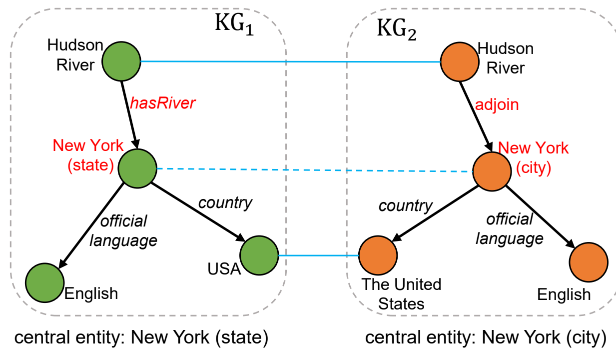

Figure 1: An entity alignment example where *New York (state)* and *New York (city)* will be aligned only using the KG structure, therefore the local relations *hasRiver* and *adjoin* are critical to distinguish them as different real-world objects.

Currently, most EA approaches firstly represent entities as distributed vectors, and then align two entities if they have similar representations [Hao *et al.*, 2016]. To accurately represent an entity, existing approaches can be classified into two categories. The first one is translation-based [Bordes *et al.*, 2013; Wang *et al.*, 2014] embedding methods, which represent an entity by exploiting its local semantics, i.e., the relational triples in which this entity appears. The other one is GCN-based [Kipf and Welling, 2017; Wang *et al.*, 2018] methods, which represent an entity by exploiting the global KG structure, i.e., all neighbors of this entity. The global structure-based methods represent an entity by aggregating all its neighbors' features, which can provide comprehensive and robust information for entity alignment, as it is less vulnerable to the missing of partial information and the schema heterogeneity of different KGs. However, it may lose fine-grained details for EA. For example in Figure 1, entities *New York (state)* and *New York (city)* cannot be distinguished using the KG structure only as they have the same neighbors, and fine-grained relations are needed to distinguish them, i.e., *hasRiver* and *adjoin*. In contrast, the local semantics-based methods represent an entity by considering the triples in which it appears, which can provide fine-grained information for EA. However, it is not robust enough if some triples are missing or two KGs are represented using different schemas. Therefore, we believe global KG structure and local relation semantics can provide complementary information for each other and can be jointly exploited for better EA performance.

In this paper, we propose **S**tructure and **S**emantics **P**reserving (SSP) network for entity alignment, which can learn the entity representations by simultaneously exploiting both the global structure and local semantics of knowledge graphs. Given a knowledge graph, our method learns entity representations in a coarse-to-fine manner. First, we employ a GCN to explicitly encode the global structure features into entity representations, which can provide comprehensive and robust information for entity alignment. Then, we adopt a modified translation-based method to refine entity representations by further exploiting the local semantics of entities. This learning strategy will ensure that the entity representation is both robust and accurate: the global structure can provide a comprehensive representation by aggregating all information about an entity's neighbors, and then the fine-grained local semantics is used to refine the representation so that even similar entities, such as *New York (state)* and *New York (city)*, can be distinguished. We conduct extensive experiments on several EA benchmarks (DBP15k and DWY100k), the experimental results show that our model achieves new state-of-the-art performance by outperforming previous baselines by a large margin. Detailed analyses verify the effectiveness and robustness of our proposed model.

The main contributions of this paper are:

- We propose to jointly leverage the global structure and local semantics for entity alignment. By exploiting both the global structure and local semantics, the entity representations can be both robust and accurate.
- We design a **S**tructure and **S**emantics **P**reserving (SSP) network which can learn the entity representations by simultaneously exploiting both the global structure and local semantics of KGs in a coarse-to-fine manner.
- Because entity representation is fundamental to many NLP applications, we believe the proposed method can also benefit many other entity representation-based tasks, e.g., link prediction and entity linking.

2 Related Work

In this section we briefly review both the *Translation-based* and *GCN-based* entity alignment methods.

Translation-based Entity Alignment. Translation-based methods [Bordes *et al.*, 2013; Wang *et al.*, 2014] represent entities by estimating the plausibility of relational triples using a scoring function, e.g., MtransE [Chen *et al.*, 2016] learns cross-lingual transitions between different embedding spaces, JAPE [Sun *et al.*, 2017] jointly trains the attribute and structure embeddings, TransEdge [Zequn *et al.*, 2019] extends TransE to an edge-centric fashion. The main advantage of translation-based methods is that they can model fine-grained relation semantics. However, they cannot preserve the global structure of KGs.

In the meantime, there are also some efforts that address the data scarcity issue. Iterative methods have been employed to improve EA performance. IPTransE [Zhu *et al.*, 2017] and BootEA [Sun *et al.*, 2018] fuse knowledge from different KGs by expanding the prior alignment seeds in a bootstrapping way. KDCoE [Chen *et al.*, 2018] performs iterative co-training of a KG embedding model and an entity description embedding model, which utilizes the entity description to assist entity alignment. Though effective, these iterative methods require more training time and computing resources. Compared with these methods, we tackle the EA problem in an end-to-end, non-iterative paradigm.

GCN-based Entity Alignment. GCN-based [Kipf and Welling, 2017] methods represent an entity by recursively aggregating its neighbors' features. GCN-Align [Wang *et al.*, 2018] seeks to embed entities and attributes as low-dimensional vectors. AliNet [Sun *et al.*, 2020] introduces distant neighbors to overcome graph heterogeneity. The main advantage of GCN-based methods is that they can obtain comprehensive and robust entity representations. However, it is difficult to model fine-grained relation semantics – because GCN jointly considers all neighbors and often ignores semantic types of relations. Following this idea, RDGCN [Wu *et al.*, 2019a] introduces the relation information of KGs through graph interactions, AVR-GCN [Ye *et al.*, 2019] devises the relation embeddings via entity embeddings, while HGCN [Wu *et al.*, 2019b] approximated relation representations via adjacent entity embeddings. Although these methods can encode relations implicitly, they cannot accurately model fine-grained relation semantics, which may severely restrict their EA performance.

In addition, there are also some other works that utilize additional information, such as attribute [Trisedya *et al.*, 2019] or description [Yang *et al.*, 2019] information to enhance entity representations. Since such additional information isn't always available, this paper merely utilize KG structure to tackle EA problem.

Based on the above discussion, we find that global structure and local semantics provide complementary information for EA: global structure can provide comprehensive and robust information for entity representations, while local semantics can provide fine-grained details to refine the coarse-grained entity representations. In this paper, we make complementary use of them to learn the global structure and local semantics-preserved entity representations for EA.

3 Problem Formulation

This section formulates the entity alignment problem. A knowledge graph $G = (E, R, T)$ is a directed graph consisting of a set of entities E, relations R, and triples T. A triple $t = (e_i, r_{ij}, t_j) \in T$ denotes head entity e_i is connected to tail entity e_j through relation r_{ij}.

Now suppose we have two different KGs to be aligned, where the source KG is denoted as $G_1 = (E_1, R_1, T_1)$ and the target KG is denoted as $G_2 = (E_2, R_2, T_2)$. Given a set of pre-aligned entity pairs $S = \{(u, v)|u \in E_1, v \in E_2, u \leftrightarrow v\}$, where $\leftrightarrow$ denotes equivalence, entity alignment task can be formulated as automatically discovering more aligned en-

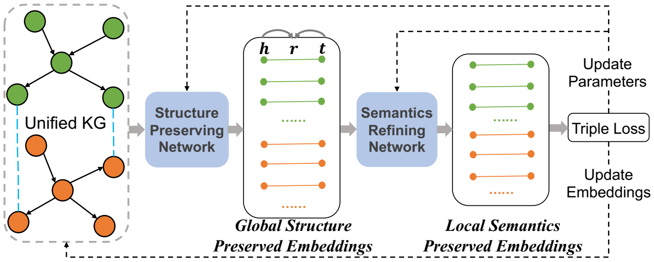

Figure 2: Overall architecture of the Structure and Semantics Preserving networks.

tity pairs based on the existing alignment seeds. Specifically, given two KGs G_1 and G_2, and a set of alignment seeds S, our method will first merge two KGs by injecting the alignment seeds, then learn entity representations with a guarantee that aligned entities should have as similar embeddings as possible, finally new aligned pairs are found using similarities between entities.

4 Structure and Semantics Preserving Networks for Entity Alignment

This section describes how to simultaneously exploit the global structure and local semantics information for entity alignment. To this end, we employ a coarse-to-fine strategy. Firstly, we employ a GCN [Kipf and Welling, 2017] to encode the KG structure into entity representations, which is comprehensive and robust. Afterward, we refine the entity representations by leveraging its local semantics via a modified translation-based embedding model. The above two modules are learned in an end-to-end manner. Figure 2 depicts the overall architecture of our model, which mainly consists of two parts: **Global Structure Preserving Network** and **Local Semantics-based Refining Network**. In follows, we describe these two networks in detail.

4.1 Global Structure Preserving Network

The global structure of KG provides useful information for entity alignment, i.e., entities with similar neighboring structures are highly likely to be aligned. This paper employs a GCN to explicitly encode the global KG structure information. Specifically, GCN follows a neighborhood aggregation scheme, where the representation vector of a node is computed by recursively aggregating and transforming representation vectors of its neighboring nodes. Concretely, GCN consists of several stacked layers. Given a set of node features $X^{(l)} = \{x_1^{(l)}, x_2^{(l)}, \cdots, x_n^{(l)} | x_i^{(l)} \in \mathbb{R}^{d^{(l)}}\}$ as input to GCN layer l, where n is the number of nodes (entities) in the unified KG, and $d^{(l)}$ is the number of features in layer l, the output of the l-th layer is obtained following the convolution computation:

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

where σ is an activation function; A is a $n \times n$ adjacency matrix that denotes the structure information of the KG; $\hat{A} = A + I$, and I is the identity matrix; $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$; $W^{(l)} \in d^{(l)} \times d^{(l+1)}$ is the weight matrix of the l-th layer in the GCN, $d^{(l+1)}$ is the number of features in the $(l+1)$-th layer.

Inspired by HGCN [Wu *et al.*, 2019b], we also employ layer-wise highway gates [Srivastava *et al.*, 2015] to control the balance of how much neighborhood information should be passed to a node and reduce noise propagation, where the output of a layer is the weighted sum of its input and the original output via gating weights $T(X^{(l)})$:

$$T(X^{(l)}) = \sigma(X^{(l)}W_T^{(l)} + b_T^{(l)}) \tag{2}$$

$$X^{(l+1)} = T(X^{(l)}) \cdot X^{(l+1)} + (1 - T(X^{(l)})) \cdot X^{(l)} \tag{3}$$

We also find that initializing the weight matrix W_T in each GCN layer using a diagonal identity matrix can further reduce parameters and boost performance. Based on GCN, our model can explicitly preserve the global structure information of KG and encode it into the learned entity representations.

4.2 Local Semantics-based Refining Network

The structure information of KGs provides useful information for representing entities. However, as mentioned above, it is hard to distinguish similar entities using KG structure only, especially for entities that have fine-grained differences. To accurately disambiguating entities with similar structures, we refine the structure-based representation using the local semantics information via translational algorithms. However, conventional translational methods are not expressive enough to model complex relation types due to their strong assumptions on the relational triples. Inspired by the recent work ELMo [Peters *et al.*, 2018] that has achieved state-of-the-art performance in many NLP tasks, we propose a modified translational method with contextualized relation learning for knowledge embedding. Our method is based on the assumption that relations occurred in different entity contexts should have distinct embeddings, regardless of whether they have the same surface forms or not. Following this idea, we calculate the relation embeddings based on the adjacent entities and the relation itself. This is intuitive that even relations with the same surface form but occurred in different contexts should have minor differences in their semantics.

Contextualized Relation Learning. Given a relational triple $(h, r, t) \in T$, we first calculate the contextualized relation embedding based on the head h, tail t and relation r itself. The entity embeddings are the output of the last layer in the HGCN described previously. Specifically, we first concatenate the embeddings of head entity h and tail entity t to aggregate context information and then use a one-layer MLP with non-linear activation to compress the entity contexts:

$$\mathbf{n}_{ht} = \sigma([\mathbf{h}; \mathbf{t}]W + \mathbf{b}) \tag{4}$$

where $[;]$ means vector concatenation. Then, we utilize context projection inspired by TransEdge [Zequn *et al.*, 2019] to further incorporate the relation embedding itself. As pointed out in TransEdge, projecting embeddings onto hyperplane has shown promising effects on the processing of disparate feature representations. The major differences between our model and TransEdge are that they rely solely on a translational method to model KG structure and introduce extra interaction entity vectors for edge-centric relation calculation. While our model employs an HGCN network to explicitly preserve the graph structure information and utilize parameter efficient relation representation method to learn knowledge embeddings, which requires less computing cost. Let $\mathbf{n}_{ht}$ be the norm vector of the hyperplane supported by $\mathbf{h}$ and $\mathbf{t}$. The contextualized relation embedding $\mathbf{r}_c$ is computed via vector projection as follows:

$$\mathbf{r}_c = \mathbf{r} - \mathbf{n}_{ht}{}^{\top}\mathbf{r}\mathbf{n}_{ht} \tag{5}$$

The contextualized relation learning mechanism can accurately model fine-grained relation semantics. We will testify the effectiveness of relation contextualization in the experiments.

Entity Embeddings Refinement. Having obtained the contextualized relation embedding $\mathbf{r}_c$, we apply it to the translational model to refine entity embeddings. Like TransE, we define a scoring function to estimate the plausibility of relational triples as:

$$f(h,r,t) = \|\mathbf{h} + \mathbf{r}_c - \mathbf{t}\| \tag{6}$$

where relational triples that exist in the KG are expected to be more plausible with low energy score. By distinctly modeling relational triples in the KG, we can further incorporate complex relation semantics information into entity embeddings and obtain more expressive knowledge embeddings. Therefor better entity alignment performance can be achieved.

Alignment Prediction. Having obtained entity embeddings with global structure and local semantics information preserved, we align entities using the similarities between their embeddings. Specifically, for each entity to be aligned, we rank all candidate entities based on their similarities to the target entity.

4.3 End-to-End Learning

For training, we regard all observed relational triples in KGs as positive samples, while all unobserved ones as negative samples (either false or missing triples). It is expected that positive samples are supposed to have lower energy than their negative counterparts. We thus minimize the following limit-based loss function as in [Zequn *et al.*, 2019] to distinguishably separate positive triples from negative ones, aiming to gather aligned entities together, and in the meantime tell unaligned ones apart.

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{T}^+} [f(h,r,t) - \beta_1]_+ + \sum_{(h',r',t')\in\mathcal{T}^-} \alpha\left[\beta_2 - f\left(h',r',t'\right)\right]_+ \tag{7}$$

α is a hyper parameter for balancing positive and negative samples, β_1 and β_2 denote positive and negative margins respectively. $[x]_+$ denotes $max(0,x)$. $\mathcal{T}^+$ is the set of observed triples, and $\mathcal{T}^-$ is the negative triples set with each triple created using the truncated uniform sampling strategy.

5 Experiments and Discussion

5.1 Experimental Settings

Datasets. We evaluate different methods using the following benchmark datasets.

- DBP15K [Sun *et al.*, 2017] is extracted from DBpedia, which contains three cross-lingual datasets: DBP_{ZH-EN}, DBP_{JA-EN} and DBP_{FR-EN}, with popular entities from English to Chinese, Japanese and French respectively. Each dataset has 15 thousand interlingual links (ILLs) as reference alignments.
- DWY100K [Sun *et al.*, 2018] contains two large-scale monolingual datasets, namely DBP-WD (DBpedia-Wikidata) and DBP-YG (DBpedia-YAGO3). Each dataset has 100,000 aligned entity pairs and hundreds of thousands of relational triples. All the datasets use the 3:7 train-test split. We refer the readers to the original papers for detailed data statistics.

Baselines. We compare with both translation-based and GCN-based EA baselines: 1) translation-based methods – MtransE [Chen *et al.*, 2016], IPtransE [Zhu *et al.*, 2017], BootEA [Sun *et al.*, 2018], TransEdge [Zequn *et al.*, 2019], MMR [Shi and Xiao, 2019]; and 2) GCN-based methods – GCN-Align [Wang *et al.*, 2018], AliNet [Sun *et al.*, 2020], where MMR is state-of-the-art end-to-end EA method that merely uses KG structure information. For fair comparison, we don't compare with EA methods which require additional information (entity names, attributes or descriptions), such as KDCoE [Chen *et al.*, 2018] and HMAN [Yang *et al.*, 2019].

System Settings. The hyper-parameters used in our model are tuned on development set and their values are: $\beta_1 = 0.2$, $\beta_2 = 2.0$, $\alpha = 0.8$. The dimensionality for embeddings in DBP15K and DWY100K datasets are 300 and 200 respectively. We set different learning rates for the structure preserving network and semantics-based refining network, they are 0.001 and 0.005 respectively. For DBP15K dataset, we uniformly sample 20 negative examples in its nearest neighbors for each relational triple (half for head and tail entities respectively), for DWY100K, the sampling size is 30. We implement our model using Pytorch – a popular deep learning framework.

Evaluation Metrics. Following previous work [Sun *et al.*, 2017], we use Hits@k, mean rank (MR) and mean reciprocal rank (MRR) to measure the performance. For MR, the smaller, the better; for other metrics, the larger, the better.

5.2 Overall Results

Table 1 and Table 2 show the overall results of all approaches on the DBP15K and DWY100K datasets. Since MMR [Shi and Xiao, 2019] doesn't release their source code, we don't report the results on DWY100K dataset. From the results, it can be seen that:

- By exploiting both the global structure and the local semantics for entity representation, the proposed model – SSP can effectively solve the entity alignment problem. Compared with other methods, our method obtains significant performance improvement on almost all

Methods	DBP15K_{ZH-EN}				DBP15K_{JA-EN}				DBP15K_{FR-EN}			
	Hits@1	Hits@10	MRR	MR	Hits@1	Hits@10	MRR	MR	Hits@1	Hits@10	MRR	MR
MtransE	0.308	0.614	0.364	154	0.279	0.575	0.349	159	0.244	0.556	0.335	139
IPTransE	0.406	0.735	0.516	-	0.367	0.693	0.474	-	0.333	0.685	0.451	-
BootEA	0.629	0.848	0.703	-	0.622	0.854	0.701	-	0.653	0.874	0.731	-
MMR†	0.679	0.867	-	-	0.654	0.858	-	-	0.675	0.890	-	-
TransEdge	0.659	0.903	0.748	50	0.646	0.907	0.741	36	0.649	0.921	0.746	25
TransEdge‡	0.735	0.919	0.801	32	0.719	0.932	0.795	25	0.710	0.941	0.796	**12**
GCN-Align	0.413	0.744	-	-	0.399	0.745	-	-	0.373	0.745	-	-
AliNet	0.539	0.826	0.628	-	0.549	0.831	0.645	-	0.552	0.852	0.657	-
SSP(-HW)	0.506	0.823	0.618	75	0.520	0.835	0.632	52	0.480	0.819	0.600	51
SSP(-RC)	0.667	0.908	0.758	42	0.665	0.918	0.757	30	0.682	0.934	0.775	17
SSP	**0.739**	**0.925**	**0.808**	**26**	**0.721**	**0.935**	**0.800**	**19**	**0.739**	**0.947**	**0.818**	13

Table 1: Overall performance on DBP15K datasets. All comparable results are taken from their original papers. The first two parts of the results separated by the dashed line denote the translational knowledge embedding methods and GCN-based methods respectively. The last part of the table denotes the results of our models. - denotes unreported results on their original papers. † denotes that the entities are aligned from two directions, and we just take the average of them. ‡ denotes the model uses a bootstrapping strategy to iteratively select likely-aligned entity pairs to enlarge the training set.

datasets. We believe that is because the structure and semantics-preserving network can effectively leverage global structure and local semantics information of KGs, both of them benefit entity alignment.

- Our method can effectively leverage the global structure and local semantics for entity representation. Compared with the naive joint method that simply concatenates the GCN-based and Translation-based entity embeddings (see SSP(Concate) in Table 2), our method outperforms it by a large performance margin. This verifies the reasonableness and effectiveness of our joint structure and semantics preserving framework.
- Our joint structure and semantics preserving network can effectively make use of the information contained in the KG triples, thus achieving robust performance on either small or large-scale EA datasets. Given no additional information, our model even significantly outperforms TransEdge‡ [Zequn *et al.*, 2019], except for MR on the DBP_{FR-EN} dataset and Hits@1 on the DBP-WD dataset. Note that, TransEdge‡ employs a bootstrapping strategy to iteratively select likely-aligned entity pairs to enlarge the training set, while our model – SSP follows a completely end-to-end paradigm and requires no data augmentation.

5.3 Detailed Analyses

To better understand how and why our method works, we conduct a series of detailed analyses.

Effect of Highway Gates. To analyze the effectiveness of highway gates in EA. We remove the highway gates from the model and use exactly the original output of each GCN layer as structure-preserved entity embeddings. SSP(-HW) from Table 1 and Table 2 denotes the ablated model. From the results, we can see that removing the highway gates results in a large performance drops on all datasets. This verifies the effectiveness of highway gates in EA. We believe that highway networks can make use of gating units to regulate the flow of information through a network and reduce noise propagation during updating node features.

Methods	DBP-WD				DBP-YG			
	Hits@1	Hits@10	MRR	MR	Hits@1	Hits@10	MRR	MR
MTransE	0.281	0.520	0.363	-	0.252	0.493	0.334	-
IPTransE	0.349	0.638	0.447	-	0.297	0.558	0.386	-
BootEA	0.748	0.898	0.801	-	0.761	0.894	0.808	-
TransEdge	0.692	0.898	0.770	106	0.726	0.909	0.792	46
TransEdge‡	**0.788**	0.938	0.824	72	0.792	0.936	0.832	43
GCN-Align	0.479	0.760	0.578	1988	0.601	0.841	0.686	299
AliNet	0.690	0.908	0.766	-	0.786	0.943	0.841	-
SSP(-HW)	0.617	0.898	0.716	61	0.726	0.928	0.798	35
SSP(-RC)	0.710	0.949	0.796	15	0.799	0.964	0.859	14
SSP(Concate)	0.591	0.721	0.639	83	0.638	0.715	0.714	69
SSP	0.772	**0.960**	**0.842**	**12**	**0.811**	**0.968**	**0.869**	**11**

Table 2: Overall performance on DWY100K datasets.

We also find that the performance drop on DWY100K dataset is not as large as that of the DBP15K dataset. This is because the DBP15K dataset is much sparser, thus removing the highway gates would add to the difficulty of disambiguating entities with similar embeddings.

Effect of Relation Contextualization. Relation contextualization is an important component in our local semantics-based refining network. To explore the effectiveness of contextualized relation learning on entity alignment, we replace it with a standard translational method, where the relation representation only depends on its type. SSP(-RC) from Table 1 and Table 2 denotes this ablated model. From the results, we can see that the model performance drops significantly, which verifies the effectiveness of contextualized relation learning. These results confirm that distinctly representing relations in different contexts can help modeling fine-grained relation semantics in KG, which benefits entity alignment.

From the above ablation studies, we can also find that removing the highway gates from SSP leads to much larger performance decline compared to SSP(-RC), this phenomenon verifies the usefulness of highway gates in regulating information flow and reducing noise propagation in GCN layers.

5.4 Discussion

Impact of Number of Alignment Seeds. To explore the impact of the number of alignment seeds (training data) on

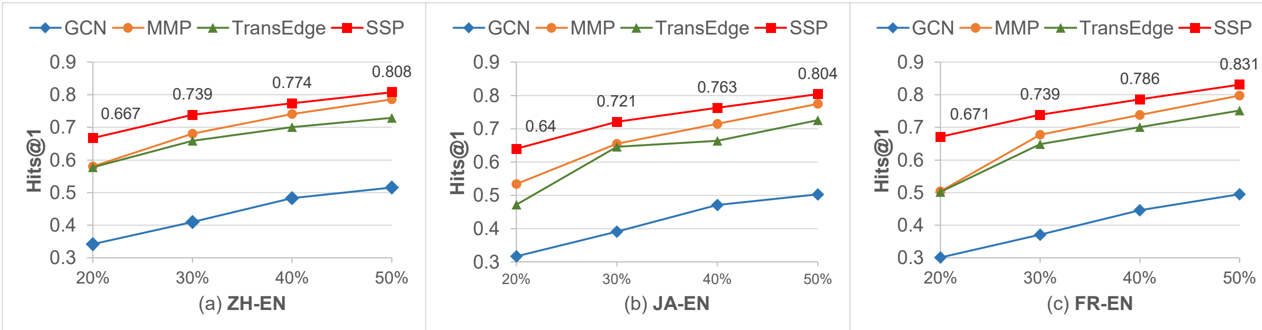

Figure 3: (a)-(c) report the Hits@1 performance of GCN, MMR, TransEdge and SSP on DBP15K datasets when they are given different proportions of alignment seeds as training data.

our model, we compare our model SSP with GCN [Wang *et al.*, 2018], MMR [Shi and Xiao, 2019] and TransEdge [Zequn *et al.*, 2019] by varying the proportion of pre-aligned entity pairs from 20% to 50% with a step of 10% and regard the rest of the alignment seeds as the testing set. As depicted in Figure 3, the performances of all models increase consistently on all datasets as the proportion of alignment seeds rises. It is clear that our model consistently outperforms its comparative counterparts by a significant margin, which verifies the robustness of our model. Notably, given 20% of alignment seeds as training data, our model achieves even comparable performance with TransEge when the latter is trained with 40% of alignment seeds. This verifies the robustness of SSP under scenarios where training data is scarce.

Impact of Entity Sparsity. In real-world KGs, the node degree (number of adjacent entities) usually follows a long-tail distribution, which means a large proportion of entities have a moderate degree and the proportion of entities with a very high or low degree is relatively small. This practical situation can have a big critical impact on EA. For sparse nodes (entities) with little structure information, it is also very hard for an embedding method to learn high-quality entity embeddings, thus would deteriorate the EA performance.

To explore the model performance on different levels of node (entity) degree, we conduct experiments on DBP15k$_{ZH-EN}$ dataset. Three representative models (SSP, SSP(-HW) and TransEdge) are selected. Specifically, we first count the degree (number of adjacent entities) of all entities in the knowledge graph. Then, we evaluate all models on the testing set and subsequently classify entities to be aligned in the testing set into three groups (with node degree in {Low, Medium, High} respectively) in ascending order according to their node degree. Finally, we count the correctly aligned entities in these three groups and calculate the respective alignment accuracy (Hits@1). Figure 4 shows the results of the above experiments. It can be seen that, under each setting, the accuracy consistently increases as the degree rises (from low degree to a high degree) in these groups. This confirms the hypothesis that densely connected (high degree) entities in the KG contain rich structure information which benefits entity alignment, while sparse entities (low degree) lack enough structure information to learn expressive embeddings, thus resulting in inferior performance.

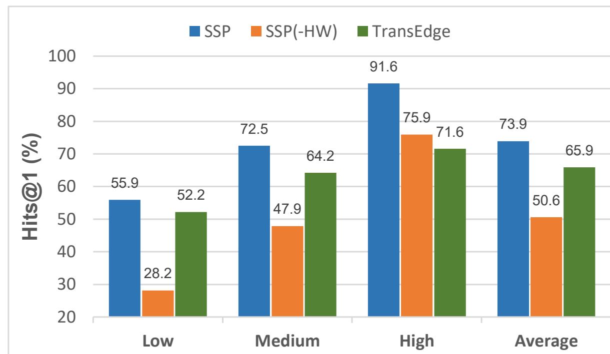

Figure 4: Model performance on different levels of node degree. Gray bar "avg" denotes model performances on the original test set.

We also find in Figure 4 that, the models exhibit much larger performance decline in the Low degree group than in High degree group when the highway gates are removed from SSP. For example, in Low and Medium degree groups, the accuracy decline are 27.7% and 24.6% respectively when it changes from SSP to SSP(-HW), while that for the High degree group is only 15.7%. This further reveals that the highway gates can help make the best of structure information of sparse entities and further boost EA performance.

6 Conclusions

In this paper, we propose a Structure and Semantics-Preserved network for entity alignment that can learn robust and accurate entity representations by leveraging the global KG structure and local semantics. Experimental results show that, our model obtains substantial performance gains on 5 standard benchmarks, and achieves new SOTA performance.

Acknowledgments

This work is supported by National Key R&D Program of China under Grant 2018YFB1005100, the National Natural Science Foundation of China under Grants no. U1936207 and 61772505, and Beijing Academy of Artificial Intelligence (BAAI2019QN0502).

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

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

[Chen *et al.*, 2016] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954*, 2016.

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

[Fu *et al.*, 2019] Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. End-to-end multi-perspective matching for entity resolution. In *IJCAI*, pages 4961–4967, 2019.

[Hao *et al.*, 2016] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. A joint embedding method for entity alignment of knowledge bases. In *CCKS*, pages 3–14, 2016.

[Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550, 2011.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Lehmann *et al.*, 2015] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[Nie *et al.*, 2019] Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *CIKM*, pages 629–638, 2019.

[Peters *et al.*, 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLP*, pages 2227–2237, 2018.

[Rebele *et al.*, 2016] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *ICLR*, pages 177–185, 2016.

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

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[Sun *et al.*, 2017] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, 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*, pages 4396–4402, 2018.

[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*, 2020.

[Trisedya *et al.*, 2019] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI*, pages 297–304, 2019.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119, 2014.

[Wang *et al.*, 2018] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, 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*, 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*, pages 240–249, 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*, 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*, pages 4135–4141, 2019.

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

[Zhang *et al.*, 2018] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *AAAI*, 2018.

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