

# Multi-Channel Graph Neural Network for Entity Alignment

Yixin Cao<sup>1</sup>   Zhiyuan Liu<sup>2</sup>   Chengjiang Li<sup>3</sup>  
Zhiyuan Liu<sup>3</sup>   Juanzi Li<sup>3</sup>   Tat-Seng Chua<sup>1</sup>

<sup>1</sup>School of Computing, National University of Singapore, Singapore

<sup>3</sup>School of Science, Xi'an Jiaotong University, Xi'an, China

<sup>3</sup>Department of CST, Tsinghua University, Beijing, China

{caoyixin2011, acharkq, iamlockelighting}@gmail.com

{liuzy, lijuanzi}@tsinghua.edu.cn, dcscts@nus.edu.sg

## Abstract

Entity alignment typically suffers from the issues of structural heterogeneity and limited seed alignments. In this paper, we propose a novel Multi-channel Graph Neural Network model (MuGNN) to learn alignment-oriented knowledge graph (KG) embeddings by robustly encoding two KGs via multiple channels. Each channel encodes KGs via different relation weighting schemes with respect to self-attention towards KG completion and cross-KG attention for pruning exclusive entities respectively, which are further combined via pooling techniques. Moreover, we also infer and transfer rule knowledge for completing two KGs consistently. MuGNN is expected to reconcile the structural differences of two KGs, and thus make better use of seed alignments. Extensive experiments on five publicly available datasets demonstrate our superior performance (5% Hits@1 up on average). Source code and data used in the experiments can be accessed at <https://github.com/thunlp/MuGNN>.

## 1 Introduction

Knowledge Graphs (KGs) store the world knowledge in the form of directed graphs, where nodes denote entities and edges are their relations. Since it was proposed, many KGs are constructed (e.g., YAGO (Rebele et al., 2016)) to provide structural knowledge for different applications and languages. These KGs usually contain complementary contents, attracting researchers to integrate them into a unified KG, which shall benefit many knowledge driven tasks, such as information extraction (Cao et al., 2018a) and recommendation (Wang et al., 2018a).

It is non-trivial to align different KGs due to their distinct surface forms, which makes the symbolic based methods (Suchanek et al., 2011) not

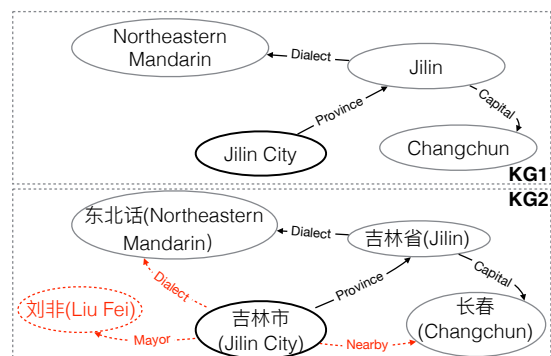


Figure 1: Illustration of the structural differences (dashed lines and ellipse) between different KGs.

always effective. Instead, recent work utilizes general KG embedding methods (e.g., TransE (Bordes et al., 2013)) and align equivalent entities into a unified vector space based on a few seed alignments (Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Chen et al., 2018; Sun et al., 2018; Wang et al., 2018b). The assumption is that entities and their counterparts in different KGs should have similar structures and thus similar embeddings. However, alignment performance is unsatisfactory mainly due to the following challenges:

**Heterogeneity of Structures** Different KGs usually differ a lot, and may mislead the representation learning and the alignment information from seeds. Take the entity *Jilin City* as an example (Figure 1), KG1 and KG2 present its subgraphs derived from English and Chinese Wikipedia, respectively. Since it is a Chinese city, KG2 is more informative than KG1 (denoted by dashed lines and ellipse), such as the relations of *Dialect* and *Nearby*, and the entity *Liu Fei* through relation *Mayor*. Clearly, the province *Jilin* in KG1 and *Jilin City* in KG2, which are incorrect alignment, are more probable close in the vector space, because they have more similar structures (e.g.,

Northeastern Mandarin and Changchun). What’s worse, this incorrect alignment shall spread further over the graph.

**Limited Seed Alignments** Recent efforts based on general embedding methods heavily rely on existing alignments as training data, while seed alignments are usually insufficient (Chen et al., 2017) for high-quality entity embeddings. Wang et al. (2018b) introduces Graph Convolution Network (GCN) (Kipf and Welling, 2017) to enhance the entity embeddings by modeling structural features, but fails to consider structural heterogeneity.

To address the issues, we propose to perform KG inference and alignment jointly to explicitly reconcile the structural difference between different KGs, and utilize a graph-based model to make better use of seed alignment information. The basic idea of structural reconciliation is to complete missing relations and prune exclusive entities. As shown in Figure 1, to reconcile the differences of *Jilin City*, it is necessary to complete the missing relations *Dialect* and *Nearby* in KG1, and filter out entity *Liu Fei* exclusive in KG2. The asymmetric entities and relations are caused not only by the incompleteness nature of KG, but also from their different demands.

In this paper, we propose a novel **Multi-channel Graph Neural Network** model MuGNN, which can encode different KGs to learn alignment-oriented embeddings. For each KG, MuGNN utilizes different channels towards KG completion and pruning, so as to reconcile two types of structural differences: missing relations and exclusive entities. Different channels are combined via pooling techniques, thus entity embeddings are enhanced with reconciled structures from different perspectives, making utilization of seed alignments effectively and efficiently. Between KGs, each channel transfers structure knowledge via shared parameters.

Specifically, for KG completion, we first employ AMIE+ (Galárraga et al., 2015) on each KG to induce rules, then transfer them between KGs towards consistent completion. Following Graph Attention Network (GAT) (Velickovic et al., 2018), we utilize KG self-attention to weighting relations for GNN channels. For KG pruning, we design cross-KG attention to filter out exclusive entities by assigning low weights to corresponding relations. We summarize the main contributions as follows:

- We propose a novel Multi-channel GNN model MuGNN that learns alignment-oriented embeddings by encoding graphs from different perspectives: completion and pruning, so as to be robust to structural differences.
- We propose to perform KG inference and alignment jointly, so that the heterogeneity of KGs are explicitly reconciled through completion by rule inference and transfer, and pruning via cross-KG attention.
- We perform extensive experiments on five publicly available datasets for entity alignment tasks, and achieve significant improvements of 5% Hits@1 on average. Further ablation study demonstrates the effectiveness of our key components.

## 2 Preliminaries and Framework

### 2.1 Preliminaries

**KG** is a directed graph  $G = (E, R, T)$  involving a set of entities  $E$ , relation types  $R$ , and triplets  $T$ . Each triplet  $t = (e_i, r_{ij}, e_j) \in T$  denotes that head entity  $e_i$  is related to tail entity  $e_j$  through relation  $r_{ij} \in R$ .

**Rule knowledge**  $\mathcal{K} = \{k\}$  can be induced from KG, e.g., in the form of  $\forall x, y \in E : \underline{(x, r_s, y) \Rightarrow (x, r_c, y)}$ , stating that two entities might be related through  $r_c$  if they are related through  $r_s$ . The left side of the arrow is defined as premise, and the right side is a conclusion. We denote rule as  $k = (r_c | r_{s1}, \dots, r_{sp})$  consisting of one or multiple  $|p|$  premises and only one conclusion.

**Rule Grounding** is to find suitable triplets satisfying the premise-conclusion relationship defined by rules. For rule  $k$ , we denote one of its grounds as  $g(k) = (t_c | t_{s1}, \dots, t_{sp})$  including  $|p| + 1$  triplets. The triplets satisfies:  $t_{s1} \wedge \dots \wedge t_{sp} \Rightarrow t_c$ , where  $\wedge$  is the logical conjunction that plays a similar role as ‘and’. Other compositions include disjunction  $\vee$  (similar to ‘or’) and negation  $\neg$  (similar to ‘not’). For example, given a rule  $\text{bornIn}(x, y) \wedge \text{cityOf}(y, z) \Rightarrow \text{nationality}(x, z)$ , we ground it in a KG, and obtain :  $\text{bornIn}(\text{Obama}, \text{Hawaii}) \wedge \text{cityOf}(\text{Hawaii}, \text{United States}) \Rightarrow \text{nationality}(\text{Obama}, \text{United States})$ . We use  $\mathcal{G}(k) = \{g(k)\}$  to denote all groundings of rule  $k$ .

**Entity alignment** takes two heterogeneous KGs  $G$  and  $G' = (E', R', T')$  as input, the goal is

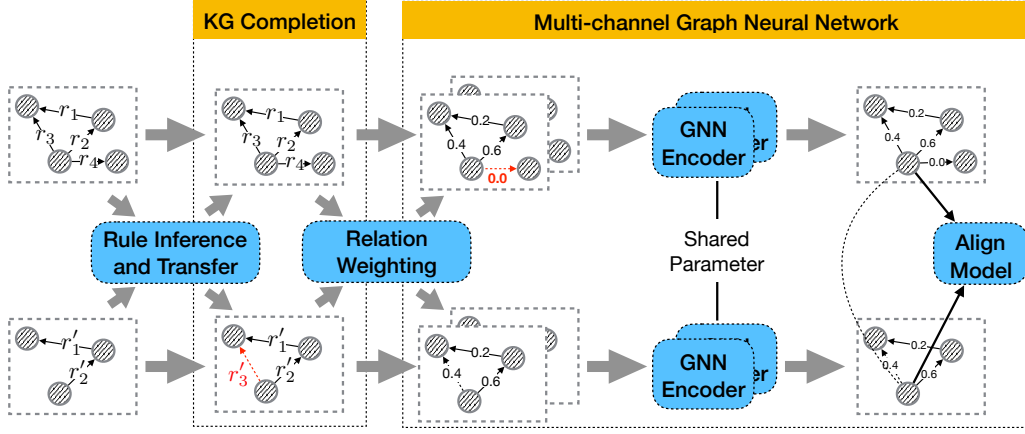


Figure 2: Framework. Rectangles denote two main steps, and rounded rectangles denote the key components of the corresponding step. After rule inference and transfer, we utilize rules to complete each KG, denoted by dashed lines  $r'_3$ . Through relation weighting, we obtain multiple weighted graphs for different GNN channels, in which relation  $r_4$  is weighted to 0.0 that prunes exclusive entities. These channels are combined as the input for align model for alignment-oriented KG embeddings.

to find as many alignments as possible  $\mathcal{A}_e = \{(e, e') \in E \times E' | e \leftrightarrow e'\}$  for which an equivalent relation  $\leftrightarrow$  holds between  $e$  and  $e'$ . That is,  $e$  and  $e'$  are in different KGs, but denote the same thing. As shown in Figure 1, *Jilin City* in English Wikipedia (i.e., KG1) and in Chinese Wikipedia (i.e., KG2) has different structures, but denote the same Chinese city. Normally, some prior alignments of entities  $\mathcal{A}_e^s$  and relations  $\mathcal{A}_r^s = \{(r, r') \in R \times R' | r \leftrightarrow r'\}$  can be easily obtained manually or by simple lexicon-based methods (e.g., entity title translation), namely seed alignments (seed for short). We use bold-face letters to denote the vector representations of the corresponding terms throughout of the paper.

## 2.2 Framework

MuGNN aims at learning alignment-oriented KG embeddings for entity alignment. It introduces KG inference and transfer to explicitly complete KGs, and utilizes different relation weighting schemes: KG self-attention and cross-KG attention, to encode KGs robustly. As shown in Figure 2, there are two main steps in our framework:

**KG Completion** aims at reconciling the structural differences by completing the missing relations. It not only induces rules by using a popular rule mining system AMIE+ (Galárraga et al., 2015), but also transfers them into each other based on seed aligned relations between KGs. Rule transferring is based on the assumption that knowledge can be generalized into various KGs, no matter in which

languages or domains.

**Multi-channel Graph Neural Network** is to encode each KG through different channels. The channels enhance the entity embeddings from different perspectives: towards completion and pruning, so that the entities and their counterparts have similar structures. MuGNN contains three main components: (1) relation weighting that generates weight matrix for each KG according to two schemes: KG self-attention and cross-KG attention. Each type of attention refers to a GNN channel that shares parameters between KGs for structural knowledge transfer; (2) GNN encoder to model the entire graph features by improving entity embeddings with its neighbors, thus the seed alignment information shall be propagated over the entire graph; We combine the outputs of GNN encoders in different channels via pooling techniques as the input of (3) Align model, which embeds two KGs into a unified vector space by pushing the aligned entities (and relations) of seeds together.

## 3 KG Completion

In this section, we introduce how to utilize rule knowledge to explicitly complete KG, which first infers rules from each KG, then transfers these rules between KGs based on knowledge invariant assumption, and finally grounds rules in each KG for consistent completion.

### 3.1 Rule Inference and Transfer

Since the acquirement of rule knowledge is not our focus in this paper, we utilize AMIE+ (Galárraga et al., 2015), a modern rule mining system, to efficiently find Horn rules from large-scale KG, such as  $marriedTo(x, y) \wedge liveIn(x, z) \Rightarrow liveIn(y, z)$ . Its source code is available online<sup>1</sup>.

Formally, given two KGs  $G$  and  $G'$ , we first mine rules separately and obtain two sets of rule knowledge  $\mathcal{K}$  and  $\mathcal{K}'$ . These rule knowledge are quite different since KGs are constructed to meet different demands of applications or languages. Although they can be used to complete their own KGs separately, we further transfer the two sets of rules into each other through **Knowledge Invariant Assumption**:

*Knowledge has universality no matter in which languages or domains.*

Given aligned relations  $\mathcal{A}_r^s$  and a rule  $k \in \mathcal{K}$ , we replace all relations involved in the rule  $k = (r_c | r_{s1}, \dots, r_{sp})$  with its counterparts if there are  $(r_c, r'_c), (r_{si}, r'_{si}) \in \mathcal{A}_r^s, i = 1, \dots, p$ . Thus, we obtain such a rule  $k' = (r'_c | r'_{s1}, \dots, r'_{sp})$  and add it to  $\tilde{\mathcal{K}}' = \mathcal{K}' \cup k'$  if  $k' \notin \mathcal{K}'$ . Real examples of transferred rules can be found in experiments. Note that there may be no transferred rules if aligned relations can not be found  $\mathcal{A}_r^s = \emptyset$ .

### 3.2 Rule Grounding

We now ground each rule sets on the corresponding KG for completion, which not only accelerates the efficiency of align model through denser KG for propagation, but also adds extra constraints that is helpful for high-quality entity embedding learning.

Take KG  $G$  as an example, given a rule  $k \in \mathcal{K}$ , we collect its grounds that the premise triples can be found in the KG, but not the conclusion triplet:  $\mathcal{G}(k) = \{g(k) | t_{s1}, \dots, t_{sp} \in T, t_c \notin T\}$ . Thus, we add all conclusion triples into the KG  $\tilde{G} = G \cup t_c, t_c \in \mathcal{G}(k)$ . Similarly, we can complete KG  $G'$  to  $\tilde{G}'$ .

As shown in Figure 1, we obtain the rule  $province(x, y) \wedge dialect(y, z) \Rightarrow dialect(x, z)$  from the informative KG2, then transfer it to KG1 based on the aligned relation *province* and *dialect*. Thus, in KG1, we find the suitable triplets  $province(Jilin\ City, Jilin) \wedge dialect(Jilin, Northeastern\ Mandarin)$ , thus obtain a new triplet  $dialect(Jilin\ City, Northeastern\ Mandarin)$ .

<sup>1</sup><https://www.mpi-inf.mpg.de/>

It is worth noting that the inferred rules do not hold in all cases, and maybe we can consider the confidence value for each grounding. We leave it in future work.

## 4 Multi-Channel Graph Neural Network

In this section, we describe the three main components involved in MuGNN to encode different graphs towards alignment-oriented embedding learning: relation weighting, multi-channel GNN encoder and align model.

### 4.1 Relation Weighting

Relation weighting is to generate weighted connectivity matrix  $A$  based on a graph  $G$  as the input structural features of GNN encoder, which will be detailed later. Each element  $a_{ij}$  in the matrix denote the weighted relation between  $e_i$  and  $e_j$ .

As mentioned in Section 1, there are two types of structure differences: the missing relations due to the incompleteness nature of KG, and the exclusive entities caused by different construction demands of applications or languages. We utilize two channels of GNN encoder for each KG, so as to reconcile the two types of differences separately. That is, we generate two adjacency matrices for each channel:  $A_1$  based on KG self-attention and  $A_2$  based on cross-KG attention. Next, we will describe how to compute each element  $a_{ij}$  in  $A_1$  and  $A_2$ . Similarly, we can obtain  $A'_1$  and  $A'_2$  for KG  $G'$ .

#### KG Self-Attention

KG self-attention aims at making better use of seed alignments based on the KG structure itself. This component selects informative neighbors according to the current entity and assigns them with high weights. Following GAT (Velićković et al., 2018), we define the normalized element  $a_{ij}$  in  $A_1$  representing the connectivity from entity  $e_i$  to  $e_j$  as follows:

$$a_{ij} = softmax(c_{ij}) = \frac{exp(c_{ij})}{\sum_{e_k \in N_{e_i} \cup \{e_i\}} exp(c_{ik})} \quad (1)$$

where  $e_k \in N_{e_i} \cup \{e_i\}$  denotes neighbors of  $e_i$  with self-loop, and  $c_{ij}$  is the attention coefficient measuring the importance of  $e_i$  to  $e_j$  and is calcu-



lated by an attention function  $attn$  as follows:

$$\begin{aligned} c_{ij} &= attn(\mathbf{W}e_i, \mathbf{W}e_j) \\ &= LeakyReLU(\mathbf{p}[\mathbf{W}e_i || \mathbf{W}e_j]) \end{aligned} \quad (2)$$

where  $||$  indicates vector concatenation,  $\mathbf{W}$  and  $\mathbf{p}$  are trainable parameters.

### Cross-KG Attention

Cross-KG Attention aims at modeling the common subgraph of two KGs as structural features towards consistency. It prunes exclusive entities by assigning lower weights for corresponding relations that have no counterparts in another KG. We define the  $a_{ij}$  in  $A_2$  as follows:

$$a_{ij} = \max_{r \in R, r' \in R'} \mathbf{1}((e_i, r, e_j) \in T) sim(r, r') \quad (3)$$

where  $\mathbf{1}(\cdot)$  indicates 1 if holds true, otherwise 0.  $sim(\cdot)$  is a similarity measure between relation types and is defined as inner-product  $sim(r, r') = \mathbf{r}^T \mathbf{r}'$ . Thus,  $a_{ij}$  is to find the best mapping between two KGs, and shall be 0 if there is no such relation types for exclusive entities.

### 4.2 Multi-Channel GNN Encoder

GNN is a type of neural network model that deals with graph-structured data, the main idea of which is similar to a propagation model: to enhance the features of a node (i.e., entity) according to its neighbor nodes. Thus, we may stack multiple  $L$  layers of GNNs to achieve further propagation.

One of its variants is based on spectral graph convolutions, such as GCN (Kipf and Welling, 2017). Every GNN encoder takes the hidden states of node representations in the current layer as inputs, and computes new node representations as:

$$GNN(A, H, W) = \sigma(\mathbf{A}\mathbf{H}\mathbf{W}) \quad (4)$$

where  $\mathbf{A}$  is an adjacency matrix showing the connectivity between nodes,  $\mathbf{H}$  is the current node representations,  $\mathbf{W}$  is the learned parameters, and  $\sigma$  is the activation function chosen as  $ReLU(\cdot) = \max(0, \cdot)$ .

Inspired by the multi-head attention networks (Velickovic et al., 2018), we use the two above-mentioned strategies to calculate connectivity matrices as different channels to propagate information from different aspects and aggregate

them with a *Pooling* function. As for our multi-channel GNN encoder, it is built by stacking multiple GNN encoder defined as:

$$\begin{aligned} \text{MultiGNN}(H^l; A_1, \dots, A_c) = \\ \text{Pooling}(H_1^{l+1}, \dots, H_c^{l+1}) \end{aligned} \quad (5)$$

where  $c$  is the number of the channels,  $A_i$  is the connectivity matrices in the  $i_{th}$  channel, and  $H_i^{l+1}$  is the computed hidden states in the  $(l+1)_{th}$  layer and  $i_{th}$  channel, which can be formulated as:

$$\mathbf{H}_i^{l+1} = GNN(A_i, H^l, W_i) \quad (6)$$

where  $W_i$  is the weight parameters in the  $i_{th}$  channel. Here, we set  $i = 1, 2$  referring to the above two attention schemes. We set  $H^0$  as the entity embeddings initialized randomly. In experiments, we select average pooling techniques for Pooling function due to its superior performance.

We use such multi-channel GNN encoders to encode each KG, and obtain  $\mathbf{H}^L, \mathbf{H}'^L$  representing the enhanced entity embeddings, where each channel shares parameters  $W_1 = W'_1$  and  $W_2 = W'_2$  for structural knowledge transferring.

### 4.3 Align Model

Align model is to embed two KGs into a unified vector space by pushing the seed alignments of entities (and relations) together. We judge whether two entities or two relations are equivalent by the distance between them. The objective of the align model is given as below:

$$\begin{aligned} \mathcal{L}_a = & \sum_{(e, e') \in \mathcal{A}_e^s} \sum_{(e_-, e'_-) \in \mathcal{A}_e^{s-}} [d(e, e') + \gamma_1 - d(e_-, e'_-)]_+ + \\ & \sum_{(r, r') \in \mathcal{A}_r^s} \sum_{(r_-, r'_-) \in \mathcal{A}_r^{s-}} [d(r, r') + \gamma_2 - d(r_-, r'_-)]_+ \end{aligned} \quad (7)$$

where  $[\cdot]_+ = \max\{0, \cdot\}$  represents the maximum between 0 and the input,  $d(\cdot) = \|\cdot\|_2$  is the distance measure chosen as L2 distance,  $\mathcal{A}_e^{s-}$  and  $\mathcal{A}_r^{s-}$  represents for the negative pair set of  $\mathcal{A}_e^s$  and  $\mathcal{A}_r^s$ , respectively, and  $\gamma_1 > 0$  and  $\gamma_2 > 0$  are margin hyper-parameters separating positive and negative entity and relation alignments. During the experiments, by calculating cosine similarity, we select 25 entities closest to the corresponding entity in the same KG as negative samples (Sun et al., 2018). Negative samples will be re-calculated every 5 epochs.

### Rule Knowledge Constraints

Since we have changed the KG structure by adding new triplets (i.e., grounded rules), we also introduce the triplet loss to hold the grounded rules as valid in the unified vector space.

Taking KG  $G$  as an example, following Guo et al. (2016), we define the loss function as follows:

$$\begin{aligned} \mathcal{L}_r = & \sum_{g^+ \in \mathcal{G}(\mathcal{K})} \sum_{g^- \in \mathcal{G}^-(\mathcal{K})} [\gamma_r - I(g^+) + I(g^-)]_+ \\ & + \sum_{t^+ \in T} \sum_{t^- \in T^-} [\gamma_r - I(t^+) + I(t^-)]_+ \end{aligned} \quad (8)$$

where  $g$  is short for rule grounding  $g(k)$ ,  $\mathcal{G}(\mathcal{K})$  and  $T$  denote all rule grounds and all triplets.  $\mathcal{G}^-(\mathcal{K})$  and  $T^-$  are negative sample sets obtained by replacing one of the involved entity using nearest sampling (Sun et al., 2018).  $I(\cdot)$  is the true value function for triplet  $t$ :

$$I(t) = 1 - \frac{1}{3\sqrt{d}} \|\mathbf{e}_i + \mathbf{r}_{ij} - \mathbf{e}_j\|_2 \quad (9)$$

or for grounding  $g = (t_c | t_{s1}, \dots, t_{sp})$ , which is recursively calculated by:

$$\begin{aligned} I(t_s) &= I(t_{s1} \wedge t_{s2}) = I(t_{s1}) \cdot I(t_{s2}) \\ I(t_s \Rightarrow t_c) &= I(t_s) \cdot I(t_c) - I(t_s) + 1 \end{aligned} \quad (10)$$

where  $d$  is the embedding size. Similarly, we obtain the loss  $\mathcal{L}'_r$  for KG  $G'$ . Thus, the overall loss function for multi-channel GNN is as follows:

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}'_r + \mathcal{L}_r \quad (11)$$

## 5 Experiment

In this section, we conduct experiments on five publicly available datasets involving both different language pairs and sources. We further investigate the key components of MuGNN and analyze how the knowledge inference and transfer mechanism contribute to KG alignment.

### 5.1 Experiment Settings

**Datasets** Following Sun et al. (2017, 2018), we conduct experiments on benchmark datasets DBP15K and DWY100K. DBP15K contains three cross-lingual datasets: DBP<sub>ZH-EN</sub> (Chinese to English), DBP<sub>JA-EN</sub> (Japanese to English), and DBP<sub>FR-EN</sub> (French to English). All the above datasets

are extracted from multilingual DBpedia and include 15,000 entity pairs as seed alignments. DWY100K consists of two large-scale cross-resource datasets: DWY-WD (DBpedia to Wikidata) and DWY-YG (DBpedia to YAGO3). Each dataset includes 100,000 alignments of entities in advance. As for the seed alignments of relations, we employ the official relation alignment list published by DBpedia for DWY100K. As for DWY-YG, we manually align the relations because there are only a small set of relation types (31) in YAGO3. The statistics<sup>2</sup> is listed in Table 1.

Datasets	$ \mathcal{A}_r^s $	#Relation	#Entity	#Triple
DBP <sub>ZH</sub>		2,830	66,469	153,929
DBP <sub>EN</sub>	891	2,317	98,125	237,674
DBP <sub>JA</sub>		2,043	65,744	164,373
DBP <sub>EN</sub>	582	2,096	95,680	233,319
DBP <sub>FR</sub>		1,379	66,858	192,191
DBP <sub>EN</sub>	75	2,209	105,889	278,590
DWY <sub>DB</sub>		330	100,000	463,294
DWY <sub>WD</sub>	62	220	100,000	448,774
DWY <sub>DB</sub>		302	100,000	428,952
DWY <sub>YG</sub>	24	31	100,000	502,563

Table 1: Statistics of DBP15K and DWY100k.

For each dataset, we employ AMIE+ for rule mining by setting the max number of premise as  $p = 2$  and PCA confidence not less than 0.8. The statistical results of rules, transferred rules (Tr.Rule for short), ground triples and ground triples based on transferred rules (Tr.ground for short) are exhibited in Table 2.

Datasets	#Rule	#Tr.Rule	#Ground	#Tr.ground
DBP <sub>ZH</sub>	2,279	1,058	46,959	19,278
DBP <sub>EN</sub>	1,906	578	78,450	24,018
DBP <sub>JA</sub>	1,440	651	61,733	25,337
DBP <sub>EN</sub>	1,316	259	77,614	17,838
DBP <sub>FR</sub>	1,263	25	77,342	1,527
DBP <sub>EN</sub>	1,252	12	75,338	1,364
DWY <sub>DB</sub>	843	40	281,271	13,136
DWY <sub>WD</sub>	630	51	184,010	56,373
DWY <sub>DB</sub>	503	4	277,031	92,923
DWY <sub>YG</sub>	39	16	129,334	10,446

Table 2: Statistics of KG inference and transfer.

**Baselines** To investigate MuGNN’s ability on entity alignment, we select four competitive baselines including three translation based mod-

<sup>2</sup> $|\mathcal{A}_r^s|$  denotes the number of seed alignments of relations.

Methods	DBP <sub>ZH-EN</sub>			DBP <sub>JA-EN</sub>			DBP <sub>FR-EN</sub>			DBP <sub>WD</sub>			DBP <sub>YG</sub>		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
<b>MTransE</b>	.308	.614	.364	.279	.575	.349	.244	.556	.335	.281	.520	.363	.252	.493	.334
<b>JAPE</b>	.412	.745	.490	.363	.685	.476	.324	.667	.430	.318	.589	.411	.236	.484	.320
<b>AlignEA</b>	.472	.792	.581	.448	.789	.563	.481	.824	.599	.566	.827	.655	.633	.848	.707
<b>GCN-Align</b>	.413	.744	.549	.399	.745	.546	.373	.745	.532	.506	.772	.600	.597	.838	.682
<b>MuGNN w/o <math>\mathcal{A}_r^s</math></b>	.479	.833	.597	.487	.851	.604	<b>.496</b>	.869	<b>.621</b>	.590	.887	.693	.730	.934	.801
<b>MuGNN</b>	<b>.494</b>	<b>.844</b>	<b>.611</b>	<b>.501</b>	<b>.857</b>	<b>.621</b>	.495	<b>.870</b>	<b>.621</b>	<b>.616</b>	<b>.897</b>	<b>.714</b>	<b>.741</b>	<b>.937</b>	<b>.810</b>

Table 3: Overall performance.

els and one graph-based model for comparison. **MTransE** (Chen et al., 2017) trains independent embedding of knowledge graph with TransE, and assigns the entity pairs in seed alignments with similar embeddings by minimizing their Euclidean distances. **JAPE** (Sun et al., 2017) learns the representation of entities and relations from different KGs in a unified embedding space. It takes advantage of attribute triples to capture homogeneous entity properties cross KGs. **GCN-Align** (Wang et al., 2018b) employs Graph Convolution Networks to construct entity representation by propagating information from the neighborhood. **AlignEA** (Sun et al., 2018) swaps aligned entities in triples to calibrate the embedding of KGs in a unified embedding space. AlignEA is the up to date non-iterative state of the art model.

**Training Details** Following Sun et al. (2017, 2018), we split 30% of entity seed alignments as training data and left the remaining data for testing. By convention, Hits@N and Mean Reciprocal Rank are used as evaluation metrics. Hits@N indicates the percentage of the targets that have been correctly ranked in top N (H in Table 3 for short). MRR is the average of the reciprocal of the rank results. Higher Hits@N and MRR refer to higher performance.

To make a fair comparison, we set embedding size to 128 for MuGNN and all baselines. All graph models stack two layers of GNN. We utilize Adagrad (Duchi et al., 2011) as the optimizer. For the margins in MuGNN, we empirically set  $\gamma_1 = 1.0$  and  $\gamma_2 = 1.0$ . We set  $\gamma_r = 0.12$  to ensure rule knowledge constraints have less impact than the alignment model. Other hyperparameters are chosen by running an exhaustively search over the following possible values: learning rate in  $\{0.1, 0.01, 0.001\}$ , L2 in  $\{0.01, 0.001, 0.0001\}$ , dropout in  $\{0.1, 0.2, 0.5\}$ . The optimal configuration of MuGNN for entity alignment is: learn-

ing rate = 0.001, L2 = 0.01, dropout = 0.2. We implement MuGNN with PyTorch-1.0. The experiments are conducted on a server with two 6-core Intel Xeon E5-2620 v3@2.40ghz CPUs, two GeForce GTX TITAN X and 128 GB of memory. 500 epochs cost nearly one hour.

## 5.2 Overall Performance

Table 3 shows the experimental results on DBP15K and DWY100K. In general, MuGNN significantly outperforms all baselines regarding all metrics, mainly because it reconciles the structural differences by two different schemes for KG completion and pruning, which are thus well modeled in multi-channel GNN.

More specifically, on three small-scale cross-lingual datasets, the average gains of MuGNN regarding Hits@1, Hits@10 and MRR are 3%, 6%, and 4%, respectively. While on large-scale datasets, MuGNN achieves significant improvements (8%, 8% and 8% regarding Hits@1, Hits@10 and MRR, respectively). This is mainly because the large-scale datasets (e.g., DBP-YG) provide more prior knowledge (more than 3.5 facts per entity v.s. less than 2.5 facts in DBP15K) for rule mining, thus our proposed method has more capability in reconciling the structural differences between KGs, and makes better use of seed alignments.

Since some methods do not rely on seed alignments of relations, we also test MuGNN without them, marked as MuGNN w/o  $\mathcal{A}_r^s$ . This also implies that we have no transferred rules between KGs. We can see that our method still performs competitively, and even achieves the best Hits@1 and MRR on DBP<sub>FR-EN</sub>. This is because the culture difference between French and English is much smaller than that between Chinese/Japanese and English, thus there is only a few exclusive rules mined from each KG, which can be transferred towards consistent completion (25 and 12 pieces of

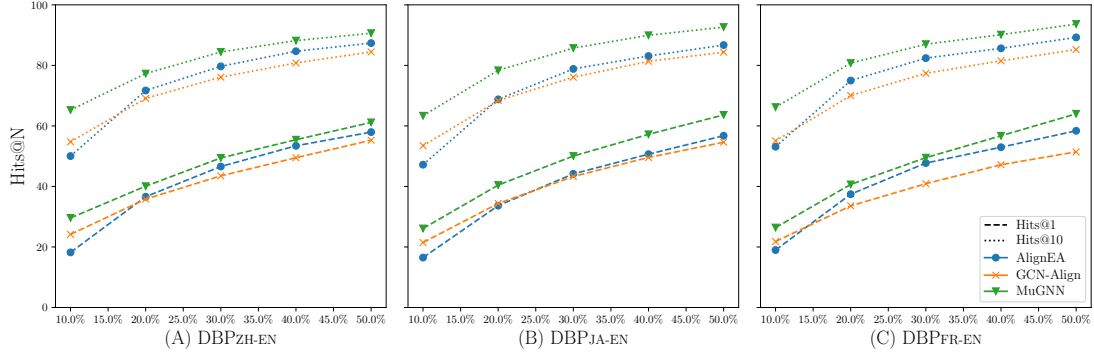


Figure 3: Sensitivity to entity seed alignments (x-axis: proportion of seed alignments used for training).

rules transferred between two KGs, as shown in Table 2).

We also observe that GNN-based method (i.e., GCN-Align) performs better than translation-based methods except AlignEA. To better understand their advantages and disadvantages, we further conduct ablation study as follows.

### 5.3 Impact of Two Channels and Rule Transfer

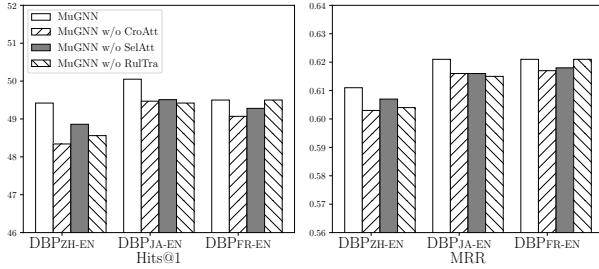


Figure 4: Impact of two channels and rule transfer.

The core components of MuGNN involve two channels based on KG self-attention and cross-KG attention, and rule transfer towards consistent completion based on knowledge invariant assumption. We thus remove them from our model to investigate their impacts to reconcile the structural differences, marked as MuGNN w/o SelAtt, MuGNN w/o CroAtt and MuGNN w/o RulTra.

As shown in Figure 4, there is a performance drop in MuGNN w/o SelAtt and MuGNN w/o CroAtt as compared to MuGNN, which demonstrates the effectiveness of both channels. Specifically, the performance decrease more with the loss of cross-KG attention channel than that of KG self-attention, which implies the importance of utilizing cross-KG information for entity alignment. As for rule transfer, we can see that in most cases, it contributes much in performance.

However, the performance difference between MuGNN and MuGNN w/o RulTra is negligible on DBPFR-EN. The reason is that the ground rule triple amounts for French and English datasets are limited (Table 2), which are less than 1% of the oracle triples. Therefore, rule transfer cannot provide sufficient cross-graph heterogeneous structure information. As a contrast, DBPJA-EN and DBPZH-EN provide more than 10k ground-rule triples, which gain decent performance improvements from rule transfer.

### 5.4 Impact of Seed Alignments

To investigate the advantages and disadvantages between GNN-based method and translation-based methods, we test MuGNN, GCN-Align and AlignEA using different size of seed alignments. We gradually increase the proportion of entity seeds from 10% to 50%, and we can see the model’s sensitivity to seed alignments.

As shown in Figure 3, GNN-based methods perform better than translation-based methods when there is only limited seeds available (10%), but perform worse along with the increase of seed alignments. This is because graph models can make better of seeds by propagating them over the entire structure, while they suffer from the heterogeneity between KGs due to the GNN’s sensitivity to structural differences, which lead to propagation errors aggregation. However, the performance of translation-based methods increases gradually along with the growing seeds since it can implicitly complete KG via knowledge representation learning, such as transE. MuGNN utilizes AMIE+ to explicitly complete two KGs via rule mining and transfer, which reconciles the structural differences; meanwhile, the GNN encoders make better use of seed information via two channels over the graphs.



---

$(U.S., leaderTitle, U.S.President) \wedge (U.S.Secretary\ of\ State, reports\ to, U.S.President) \Rightarrow (U.S.Secretary\ of\ State, seat, U.S.)$ $(Chiang\ Kaishek, party, Kuomintang) \wedge (Chiang\ Weikuo, president, Chiang\ Kaishek) \Rightarrow (Chiang\ Weikuo, party, Kuomintang)$
---

---

Table 4: Examples of groundings of transferred rules.

## 5.5 Qualitative Analysis

We qualitatively analyze how the rule works by presenting the transferred rules and their groundings in Table 4. We can see the rule grounding in the first line indicates a common knowledge in the United States, which thus is easily mined in English KG DBP<sub>EN</sub>. Meanwhile, we find that such knowledge is missing in DBP<sub>ZH</sub>, the Chinese KG. By transferring the corresponding rules from DBP<sub>EN</sub> to DBP<sub>ZH</sub>, the asymmetric information is smoothed. Corresponding entities in Chinese DBP<sub>ZH</sub> shall have a similar structure with their counterparts in English DBP<sub>EN</sub>, thus similar embeddings. That is, MuGNN indeed reconciles structural differences by rule transfer, and learns alignment-oriented embeddings. The second line presents a similar case that transfers a Chinese common rule knowledge into English KG. This demonstrates the effectiveness of rule transfer.

**Error Analysis:** As shown in Table 2, the only 4 rules transfer from YAGO3 to DBpedia are grounded to 92,923 new ground rule triples, which is shocking and not informative. Further investigation finds that the rule  $(a, team, b) \Rightarrow (a, affiliation, b)$  alone contributes 92,743 ground rule triples. Although the rule is logically correct, it is suspicious such a rule that establishes similar relations between entities would benefit entity alignment. We will deal with such noise in future.

## 6 Related Work

Merging different KGs into a unified one has attracted much attention since it shall benefit many Knowledge-driven applications, such as information extraction (Cao et al., 2017a, 2018b), question answering (Zhang et al., 2015) and recommendation (Cao et al., 2019). Early approaches for entity alignment leverage various features to overcome the heterogeneity between KGs, such as machine translation and external lexicons (Suchanek et al., 2011; Wang et al., 2013). Following the success of KG representation learning, recent work embeds entities in different KGs into a low-dimensional vector space with the help of seed alignments (Chen et al., 2017). However, the limited seeds and structural differences

take great negative impacts on the quality of KG embeddings, which performs alignment poorly. JAPE (Sun et al., 2017) and KDCoE (Chen et al., 2018) introduced attributes or descriptions information to improve entity embeddings, while IP-TransE (Zhu et al., 2017) and BootEA (Sun et al., 2018) enlarged the seed set by selecting predicted alignments with high confidence iteratively.

Clearly, the above strategies can be seen as a general enhancement for most alignment approaches (Sun et al., 2018), thus we focus on improving the alignment performance without any external information and in a non-iterative way. Inspired by Wang et al. (2018b), which utilize Graph Convolutional Network (GCN) (Kipf and Welling, 2017) to encode the entire KGs, we aim at reconciling the heterogeneity between KGs through completion and pruning, and learn alignment-oriented KG embeddings by modeling structural features from different perspectives via Multi-channel GNNs.

## 7 Conclusions

In this paper, we propose a novel Multi-channel Graph Neural Network model, MuGNN, which learns alignment-oriented KG embeddings for entity alignment. It is able to alleviate the negative impacts caused by the structural heterogeneity and limited seed alignments. Through two channels, MuGNN not only explicitly completes the KGs, but also pruning exclusive entities by using different relation weighting schemes: KG self-attention and cross-KG attention, showing robust graph encoding capability. Extensive experiments on five publicly available datasets and further analysis demonstrate the effectiveness of our method.

In future, we are interested in introducing text information of entities for alignment by considering word ambiguity (Cao et al., 2017b); and meanwhile, through cross-KG entity proximity (Cao et al., 2015).

## Acknowledgments

NExT++ research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its IRC@SG Funding Initiative.

## References

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.
- Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018a. Neural collective entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686.
- Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Chengjiang Li, Xu Chen, and Tiansi Dong. 2018b. Joint representation learning of cross-lingual words and entities via attentive distant supervision. In *EMNLP*.
- Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017a. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *ACL*.
- Yixin Cao, Juanzi Li, Xiaofei Guo, Shuanhu Bai, Heng Ji, and Jie Tang. 2015. Name list only? target entity disambiguation in short texts. In *EMNLP*.
- Yixin Cao, Jiaxin Shi, Juanzi Li, Zhiyuan Liu, and Chengjiang Li. 2017b. On modeling sense relatedness in multi-prototype word embedding. In *IJCNLP*.
- Yixin Cao, Xiang Wang, Xiangnan He, Tat-Seng Chua, et al. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. *arXiv preprint arXiv:1902.06236*.
- Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *IJCAI*.
- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*.
- Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie++. *The VLDB Journal—The International Journal on Very Large Data Bases*.
- Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *ACL*.
- Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *ISWC*.
- Fabian M Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. Paris: Probabilistic alignment of relations, instances, and schema. *Proceedings of the VLDB Endowment*.
- Zequan Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*.
- Zequan Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
- Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2018a. Explainable reasoning over knowledge graphs for recommendation. *arXiv preprint arXiv:1811.04540*.
- Zhichun Wang, Juanzi Li, and Jie Tang. 2013. Boosting cross-lingual knowledge linking via concept annotation. In *IJCAI*.
- Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018b. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*.
- Mengdi Zhang, Tao Huang, Yixin Cao, and Lei Hou. 2015. Target detection and knowledge learning for domain restricted question answering. In *NLPCC*.
- Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*.