Bootstrapping Entity Alignment with Knowledge Graph Embedding

Zequn Sun, Wei Hu*, **Qingheng Zhang** and **Yuzhong Qu**
State Key Laboratory for Novel Software Technology, Nanjing University, China
{zqsun, qhzhang}.nju@gmail.com, {whu, yzqu}@nju.edu.cn

4.1 Alignment-Oriented KG Embedding

Our alignment-oriented KG embedding model aims to encode different KGs into a unified embedding space, such that the alignment likelihood between entities can be directly measured via their embeddings. It captures the semantics hidden in KGs and is free of symbolic heterogeneity.

In a single KG, the diversified relations between entities characterize its semantics. The translation-based models have shown their success in modeling KG semantics. They define score function $f(\tau) = \|\vec{v}(h) + \vec{v}(r) - \vec{v}(t)\|_2^2$ to measure the plausibility of triple $\tau = (h, r, t)$. They optimize the margin-based ranking loss to make the scores of positive triples lower than those of negative ones. However, as studied in [Zhou *et al.*, 2017], this loss function cannot ensure that the scores of positive triples are absolutely low to fulfill the translation. For entity alignment, absolutely low scores of positive triples help reduce the drift of embeddings in the unified space and better capture the common semantics of different KGs. Therefore, we propose a new objective function, denoted by $\mathcal{O}_e$, based on the limit-based loss [Zhou *et al.*, 2017]:

$$\mathcal{O}_e = \sum_{\tau \in \mathbb{T}^+} [f(\tau) - \gamma_1]_+ + \mu_1 \sum_{\tau' \in \mathbb{T}^-} [\gamma_2 - f(\tau')]_+, \quad (3)$$

Parameter Swapping

To leverage prior alignment $\mathbf{A}'$ for bridging different KGs, we swap aligned entities in their triples to calibrate the embeddings of KG_1 and KG_2 in the unified embedding space. Given an aligned entity pair $(x, y) \in \mathbf{A}'$, we generate the following supervised triples:

$$\begin{aligned}\mathbb{T}^s_{(x,y)} =& \{(y,r,t) | (x,r,t) \in \mathbb{T}_1^+\} \cup \{(h,r,y) | (h,r,x) \in \mathbb{T}_1^+\} \\ & \cup \{(x,r,t) | (y,r,t) \in \mathbb{T}_2^+\} \cup \{(h,r,x) | (h,r,y) \in \mathbb{T}_2^+\},\end{aligned} \quad (4)$$

where $\mathbb{T}_1^+$ and $\mathbb{T}_2^+$ denote the positive triple sets of KG_1 and KG_2, respectively. In total, we have $\mathbb{T}^+ = \mathbb{T}_1^+ \cup \mathbb{T}_2^+ \cup \mathbb{T}^s$ in Eq. (3), where $\mathbb{T}^s = \bigcup_{(x,y) \in \mathbf{A}'} \mathbb{T}^s_{(x,y)}$. Then, we sample negative triples $\mathbb{T}^-$ for $\mathbb{T}^+$.

Title of Paper

Multi-Context Attention for Entity Matching

Dongxiang Zhang
Zhejiang University
zhangdongxiang@zju.edu.cn

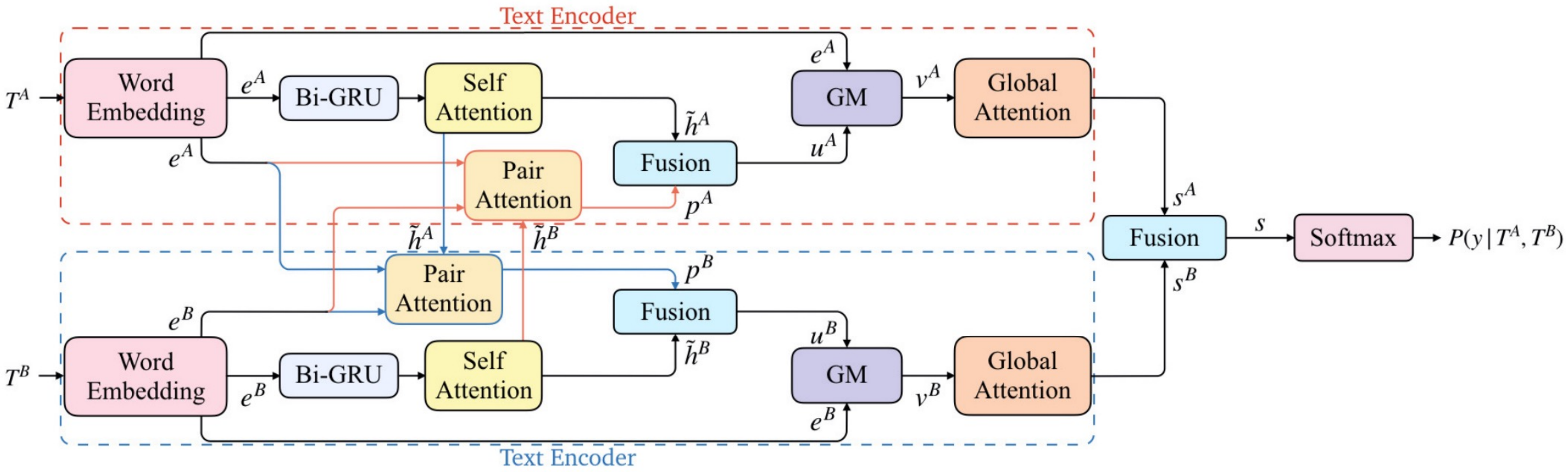
Yuyang Nie
University of Electronic Science and Technology of China
nyy477@qq.com

Sai Wu*
Zhejiang University
wusai@zju.edu.cn

Yanyan Shen
Shanghai Jiao Tong University
shen-yy@cs.sjtu.edu.cn

Kian-Lee Tan
National University of Singapore
tankl@comp.nus.edu.sg

MCA


Text Encoder
T^A
Word Embedding
e^A
Bi-GRU
Self Attention
$\tilde{h}^A$
Fusion
e^A
GM
v^A
Global Attention
u^A
e^A
Pair Attention
p^A
s^A
Fusion
s
Softmax
$P(y \mid T^A, T^B)$
$\tilde{h}^A$
$\tilde{h}^B$
Pair Attention
p^B
Fusion
s^B
e^B
u^B
$\tilde{h}^B$
T^B
Word Embedding
Bi-GRU
Self Attention
GM
Global Attention
e^B
v^B
e^B
Text Encoder

MCA

- Self-attention mechanism

➢ building connection for the terms within the same sentence but at different positions

➢ capturing both long-range and local dependencies and can facilitate word disambiguation

The self-attention is applied on the output of bi-directional GRU to produce context-aware representation:

$$h^A = Bi - GRU(e^A)$$

$$\alpha = softmax((h^A)^T \cdot h^A)$$

$$\tilde{h}^A = h^A \cdot \alpha^T$$

- Pair-attention mechanism

➢ perform soft alignment and pairwise token comparison across the two input sequences

➢ obtain soft-aligned encoding of e_i^T using all elements in the sequence of word embeddings in T^B

$$\beta = softmax((e^A)^T W^p e^B)$$

$$p^A = \tilde{h}^A \cdot \beta^T$$

MCA

- Fusion module

➢ Using highway network to aggregate the attended output $\tilde{h}^A$ of self-attention mechanism and output p^A of pair-attention mechanism.

$$u^A = Highway([\tilde{h}^A, p^A, |\tilde{h}^A, p^A|, \tilde{h}^A \odot p^A])$$

$$Highway(\boldsymbol{x}) = g \cdot H(\boldsymbol{x}, \boldsymbol{W_H}) + (1 - g) \cdot \boldsymbol{x}$$

- Gate Mechanism

➢ utilize GM to assign different weights to two input channels e^A and u^A, e^A is the raw word embeddings and u^A is the aggregated representation from self-attention and pair-attention.

$$g = \sigma(\boldsymbol{W}_1 \boldsymbol{e}^A + \boldsymbol{W}_2 \boldsymbol{u}^A + b)$$

$$v^A = g \cdot \boldsymbol{e}^A + (1 - g) \cdot \boldsymbol{u}^A$$

MCA

- Global attention mechanism

  - Capture the salience of discriminative terms in application domain
  - apply global weight vector on v^A obtain representation for the whole text record T^A

$$\lambda = softmax((v^A)^T \cdot c_x)$$

$$s^A = v^A \cdot \lambda^T$$

- Binary classifier

  - use Highway to generate an aggregated representation $\boldsymbol{s}$ for binary classification

$$s = Highway([s^A, s^B, |s^A - s^B|, s^A \odot s^B])$$

  - the matching probability P(y|T A,T B) is yielded by:

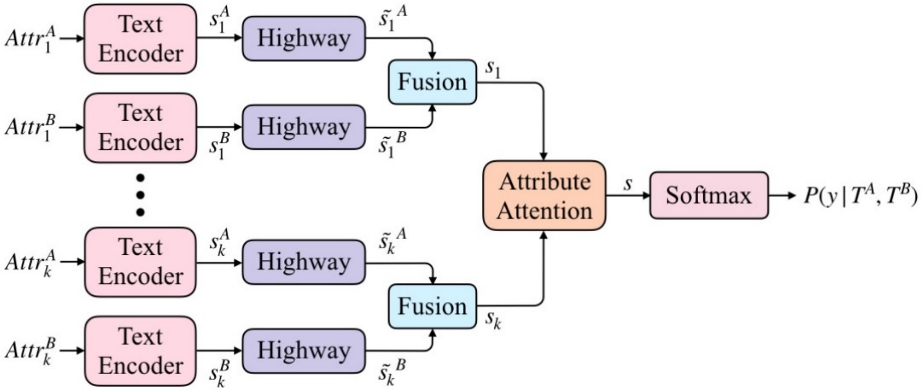
$$P(y|T^A, T^B) = softmax(w_o s + b)$$

MCA

- Extended structural EM

➢ apply the textual encoding proposed previously to obtain the semantic representation for each attribute.

➢ use the two proposed function blocks, Highway and Fusion, to derive the encoding of pair representation in a particular attribute.



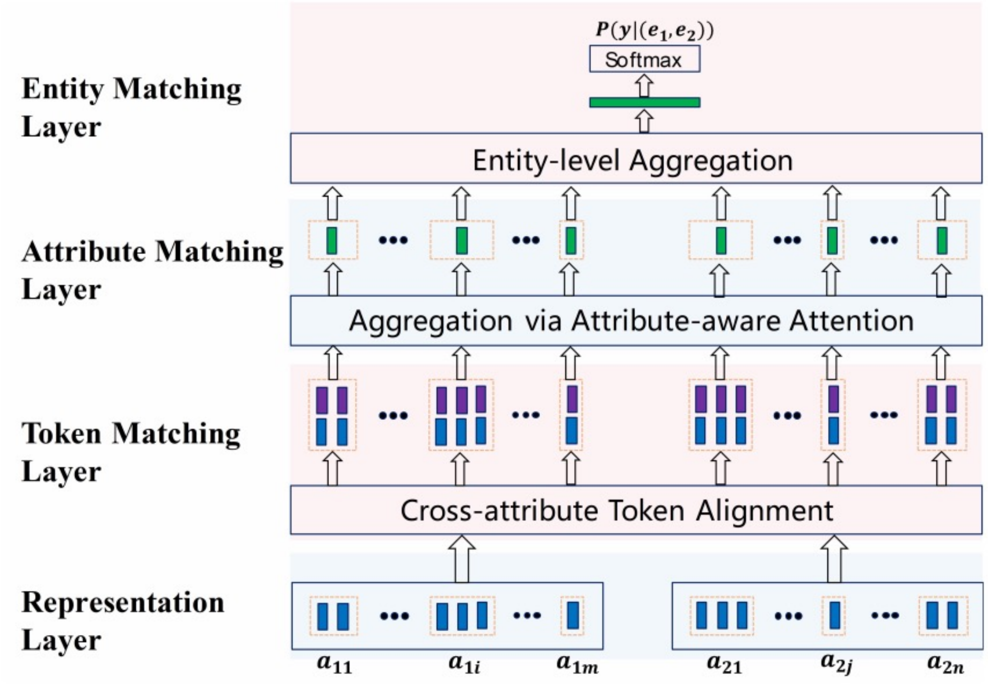
Title of Paper

Hierarchical Matching Network for Heterogeneous Entity Resolution

Cheng Fu[1,3*], **Xianpei Han**[1,2*], **Jiaming He**[4] and **Le Sun**[1,2]

[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]Brandeis University
{fucheng, xianpei, sunle}@iscas.ac.cn, germing@cs.brandeis.edu

Hier-Matcher



SOUTHEAST UNIVERSITY

Hier-Matcher

Token Matching via Cross-Attribute Alignment

cross-attribute token alignment module uses a global selection mechanism to select the most similar token from the other entity for each token.

- Compare w_{1it} with all tokens in entity e' to get a compare matrix $C_{1it} \in R^{2u \times Q}$

$$C_{1it} = |h_{1it} \otimes e_Q - H_2|$$

- Feed the compare matrix through HighwayNet, and transform the output matrix $G_{1it} \in \mathbb{R}^{2u \times Q}$ to an attention vector $v_{1it} \in \mathbb{R}^{Q}$ using a linear layer followed by a softmax function

$$G_{1it} = HighwayNet(C_{1it})$$

$$v_{1it} = softmax(wG_{1it} + b)$$

- Obtain selection vector $s_{1it} \in \mathbb{R}^{Q}$ by elements-wise function f which assigns 1 to $s_{1it}[l]$ if $v_{1it}[l] = \max(v_{1it})$ otherwise 0:
- use the global selection vector s_{1it} to pick out the aligned token for w_{1it} : $r_{1it} = C_{1it} s_{1it}^{T}$

Hier-Matcher

Attribute Matching via Attribute-aware Attention

- compute the attribute level comparison result of a_{1i} as a weighted sum of comparison results of all its tokens:

$$r_{1i} = \sum_{t=1}^{T} \alpha_{1it} r_{1it}$$

r_{1it} is the comparison vector of the t_{th} token w_{1it}

- α_{1it} is an attention score representing the importance of w_{1it}:

$$\alpha_{1it} = exp(p_{it}^{T} h_{1it}) / \sum_{t=1}^{T} exp(p_{it}^{T} h_{1it})$$

p_{it}^{T} is the context vector of attribute A_{1i} , it is randomly initialized and jointly learned during training.

Hier-Matcher

Entity Matching Layer

- concatenate all the attribute level comparison results as a $2u(m + n)$ dimension evidence vector:

$$r = [r_{11}; r_{11}; \dots; r_{11}; r_{21}; r_{22}; \dots; r_{2m}]$$

- feed r into a two-layer fully-connected ReLU HighwayNet followed by a softmax layer, which outputs the matching probability $P(y = 1|e, e')$.

Title of Paper

Knowledge Graph Alignment Network with Gated Multi-Hop Neighborhood Aggregation

Zequn Sun,[1*] Chengming Wang,[1*] Wei Hu,[1†] Muhao Chen,[2] Jian Dai,[3] Wei Zhang,[3] Yuzhong Qu[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Department of Computer Science, University of California, Los Angeles, USA
[3]Alibaba Group, China
{zqsun.nju, cmwang.nju}@gmail.com, {whu, yzqu}@nju.edu.cn, muhaochen@ucla.edu, {yiding.dj, lantu.zw}@alibaba-inc.com

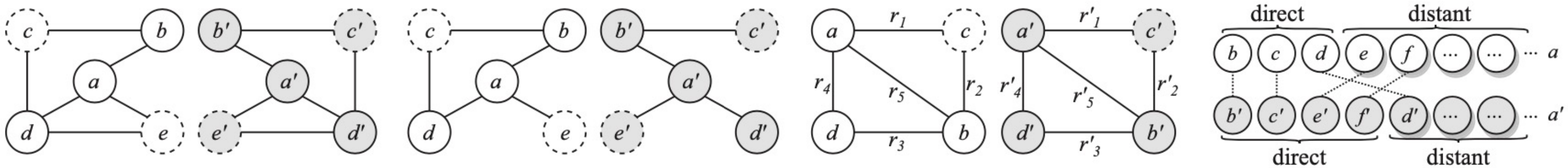
AliNet

- isomorphic structures are beneficial

  the alignment information between entities can be propagated across the different GNN layers and different isomorphic graphs given partially pre-aligned neighborhood

- only structures are not enough

  Conventional GNNs fall short of characterizing some special subgraph structures such as triangular graphs

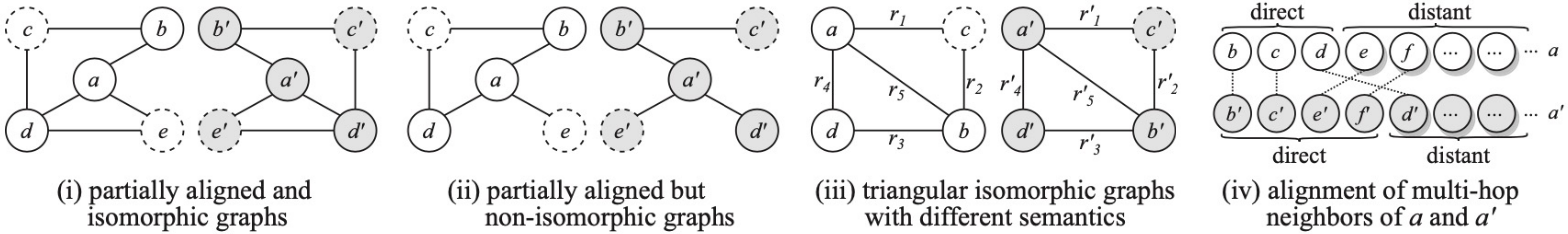


(i) partially aligned and isomorphic graphs (ii) partially aligned but non-isomorphic graphs (iii) triangular isomorphic graphs with different semantics (iv) alignment of multi-hop neighbors of a and a'

AliNet

➢ Compensation with distant neighborhood and relations

- the schema heterogeneity of different KGs usually brings about the mixture of direct and distant neighbors of counterpart entities.
- the aggregation of distant neighbors should be attentive and selective as not all the distant neighbors are helpful. .



AliNet

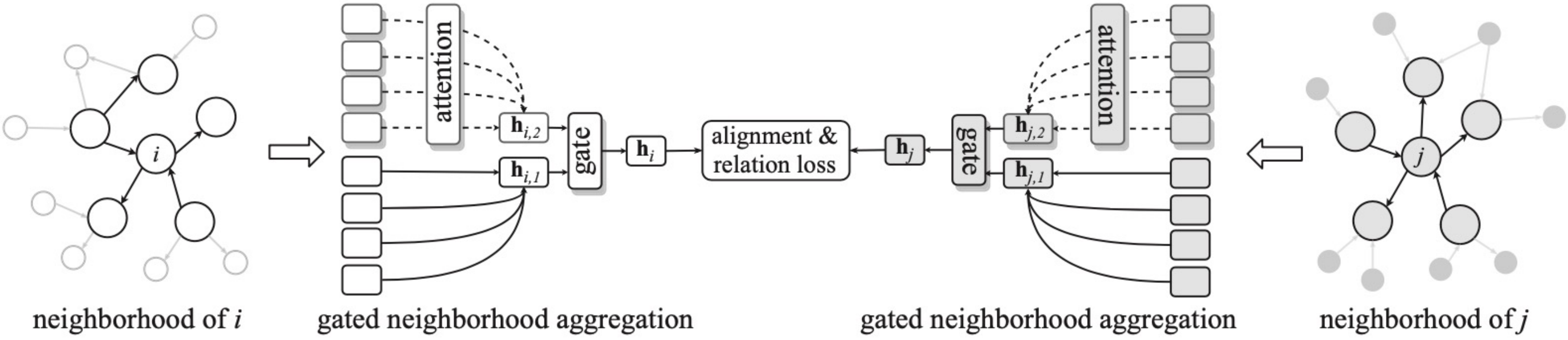


Figure 3: Overview of the KG alignment network (AliNet) with gated two-hop neighborhood aggregation.

AliNet

➢ Gated Multi-hop Neighborhood Aggregation

- aggregate one-hop neighbor representations using the vanilla GCN layers.

$$h_{i,1}^{(l)} = \sigma\Big(\sum_{j \in N_1(i) \cup \{i\}} \frac{1}{c_i} W^{(l)} h_j^{l-1}\Big)$$

- aggregate two-hop neighbor information using the attention mechanism because directly employing the original aggregation of GCN would cause noise information to propagate through layers.

$$h_{i,2}^{(l)} = \sigma\Big(\sum_{j \in N_2(i) \cup \{i\}} \alpha_{ij}^{(l)} W_2^{(l)} h_j^{l-1}\Big)$$

- use the gating mechanism to combine the information from one-hop and two-hop neighbors directly

$$h_i^{(l)} = g\left(h_{i,2}^{(l)}\right) \cdot h_{i,1}^{(l)} + (1 - g\left(h_{i,2}^{(l)}\right)) \cdot g\left(h_{i,2}^{(l)}\right)$$

AliNet

➢ Attention for Distant Neighborhood

- use two matrices $M_1^{(l)}$ and $M_2^{(l)}$ for the linear transformations of the central entity and its neighbors, respectively
- the attention weight $c_{ij}^{(l)} \in \mathbb{R}$ between i and j at the l-th layer is computed as follows:

$$c_{ij}^{(l)} = LeaklyReLU[(M_1^{(l)} h_i^{(l)})^T (M_1^{(l)} h_i^{(l)})]$$

- normalize attention weights using the softmax function to make them comparable across different entities

$$\alpha_{ij}^{(l)} = softmax_j\left(c_{ij}^{(l)}\right) = \frac{\exp(c_{ij}^{(l)})}{\sum_{n \in N_2(i) \cup \{i\}} \exp(c_{ij}^{(l)})}$$

AliNet

➢ Contrastive alignment loss

- minimize the contrastive alignment loss to let the representations of aligned entities have a very small distance while those of unaligned entities have a large distance

$$L_1 = \sum_{(i,j)\in A^+} ||h_i - h_j|| + \sum_{(i,j)\in A^-} \alpha_1[\lambda - ||h_{i'} - h_{j'}||]_+$$

- use the hidden representations of all layers as the representations of each layer all contribute to propagating alignment information

$$h_i = \oplus_{l=1}^{L} norm(h_i^{(l)})$$

AliNet

➢ Relation Semantics Modeling

- To avoid overhead of parameters, it does not introduce additional relation-specific embeddings. The representation for r can be retrieved via its related entity embeddings:

$$r = \frac{1}{|T_r|} \sum_{(s,o)\in T_r} (h_s - h_o)$$

- minimize the following relation loss for refinement:

$$L_2 = \sum_{r\in R} \frac{1}{|T_r|} \sum_{(s,o)\in T_r} ||h_s - h_o - r||$$

AliNet

➢ Implementation

- objective:

$$L = L_1 + \alpha_2 L_2$$

- generalization to k-hop neighborhood :

$$h_i^{(l)} = \rho_{k-1}(\ldots \rho_2(\rho_1(h_{i,1}^{(l)}, h_{i,2}^{(l)}), h_{i,3}^{(l)}) \ldots)$$

- neighborhood augmentation: add balanced edge linking to mitigate the non-isomorphism(if two entities i and j of KG_1 have an edge while their counterparts i' and j' in KG_2
- alignment prediction : given a entity i in KG_1 , its counterpart in KG_2 is:

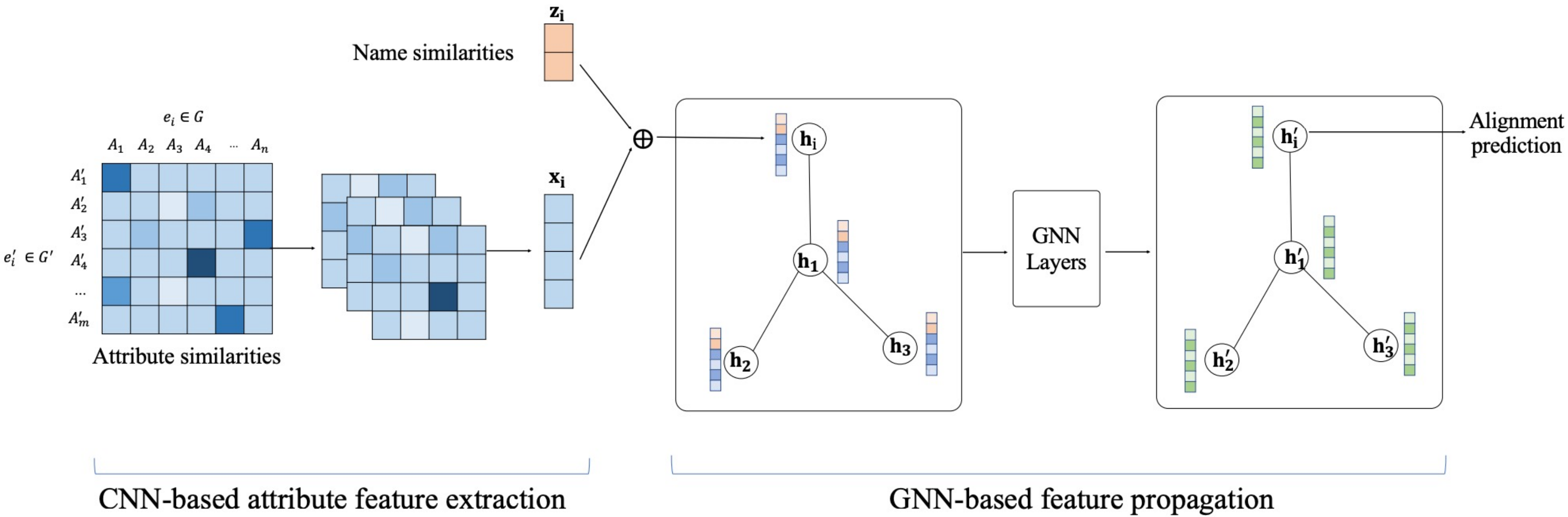
$$i' = argmin_{j \in \varepsilon_2} \pi(h_i, h_j)$$

Title of Paper

Knowledge Graph Alignment with Entity-Pair Embedding

Zhichun Wang, Jinjian Yang, Xiaoju Ye
School of Artificial Intelligence, Beijing Normal University, Beijing, China
Engineering Research Center of Intelligent Technology and Educational Application,
Ministry of Education, Beijing, China
zcwang@bnu.edu.cn {tecyyang,mryx}@mail.bnu.edu.cn

PCG-Matcher



$\mathbf{z_i}$
Name similarities
$e_i \in G$
A_1 A_2 A_3 A_4 ... A_n
A'_1
A'_2
A'_3
$e'_i \in G'$
A'_4
...
A'_m
Attribute similarities
$\mathbf{x_i}$
$\oplus$
$\mathbf{h_i}$
$\mathbf{h_1}$
$\mathbf{h_2}$
$\mathbf{h_3}$
GNN Layers
$\mathbf{h'_i}$
$\mathbf{h'_1}$
$\mathbf{h'_2}$
$\mathbf{h'_3}$
Alignment prediction
CNN-based attribute feature extraction
GNN-based feature propagation

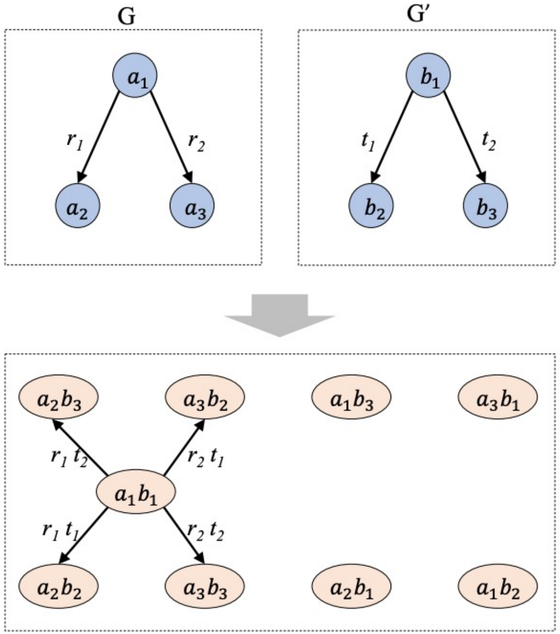
PCG-Matcher

➢ Generate PCG

- Using LSH to select entity-pairs having high equivalent possibilities as nodes in PCG.

$$< a, r, b >\in T \wedge < a', r', b' >\in T'$$

$$\Leftrightarrow < (a, a'), (r, r'), (b, b') >\in \mathcal{T}$$



PCG of G and G′

PCG-Matcher

➢ Attribute feature generation

- Compute similarity matrix $M_{m\times n}$ by comparing values of every attribute pair of two entities

$$Jaccard(s,t) = \frac{|NG(s) \cap NG(t)|}{NG(s) \cup NG(t)}$$

- Use CNN model to encode the sparse similarity matrix into a short and dense vector

$$X_k^{(l)} = ReLU(W_k^{(l)} \otimes X^{(l-1)} + b_k^{(l)})$$

- Name similarity feature

name or label of an entity is considered as an important clue for determining whether two entities are equivalent. The obtained name similarity will be concatenated with $X_k^{(l)}$:

$$\boldsymbol{z} = [z_1, z_2, z_3, z_4]$$

z_1:string equality z_2:edit distance z_3:Jaccrad similarity z_4:substring similarity

PCG-Matcher

➢ Attribute-based feature propagation

- Edge-aware attention mechanism: compute the attention α_{ij} based on the features of node i and j:

$$e_{ij} = LeakyReLU(a^T[Wh_i||Wh_j||t_{(i\rightarrow j)}])$$

- the vector of an edge-type $t_{(i\rightarrow j)}$ is computed based on the nodes' vectors connected by it

$$t_k = |\frac{1}{|S_k|}\sum_{i\in S_k} Wh_i - \frac{1}{|T_k|}\sum_{j\in T_k} Wh_j|$$

- normalized attention coefficients using a softmax function:

$$\alpha_{ij} = softmax_j(e_{ij})$$

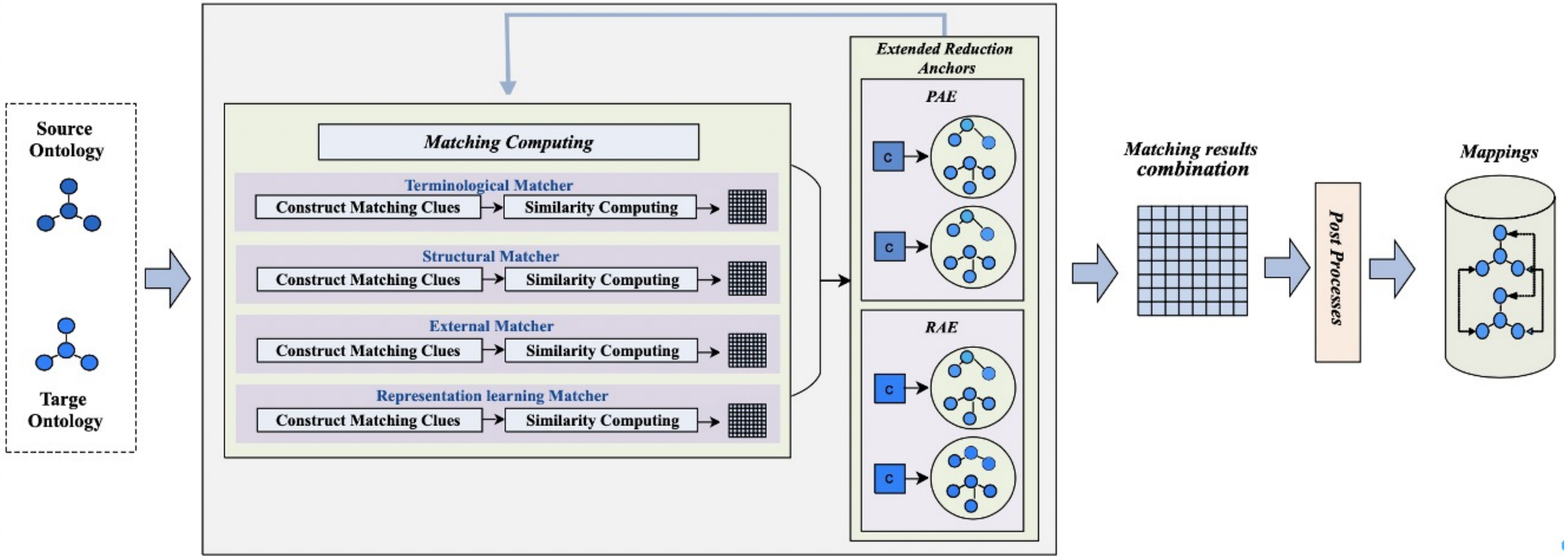
PCG-Matcher

➢ Residual connections in GNN

residual connections from the input features to the output layer of the GNN model To let the entity-pair embeddings memorize the original attribute features

- let $F = F'$, i.e. the sizes of input and output node vectors of each GNN layer are the same
- add s shortcut connection between the input and output layers
- Compute the final representation of a node by element-wise addition of h_i^0 and h_i^L

Matching Biomedical Ontologies: Clues, Approach, and Scalability



Matching Biomedical Ontologies: Clues, Approach, and Scalability

Table 1. Atomic clues for ontology matching.

Clues	Clues sources	Description
Terminological		
	Local name	Words in the local name of *e*.
	Label	Words in the rdfs:label of *e*.
	Comment	Words in the rdfs:comment of *e*
	Synonym	Words in the synonym statements such as {*owl: sameAs*} and {*rdfs: seeAlso*}.
Structural		
	Property	Property attributes of concepts: property name, domain, range and constraints.
	Hierarchy	Hierarchical context of concepts or properties, containing ancestors, descendants, siblings and disjoint elements.
External		
	General dictionary	Retrieval of alternative labels and synonyms from general dictionaries, such as WordNet, BabelNet, et al.
	Lexicon	Cross-searching synonyms as well as cross- references from specific-domain thesauri.
Representation learning		
	General model	The embeddings of elements via general pre-trained language models, such as Word2Vec and BERT.
	Specialized mode	The embeddings via domain-specific pre-trained models, such as BioBERT.

Matching Biomedical Ontologies: Clues, Approach, and Scalability

Table 3. The relations between clues and matchers.

Matcher		Clue											
		name	label	syn	prop	dh	lh	gh	WN	U_{dic}	BERT	BioBERT	fBio
Terminological													
	M_1	√											
	M_2		√										
	M_3		√	√									
Structural													
	M_4		√	√	√								
	M_5		√	√		√							
	M_6		√	√	√	√							
	M_7		√	√			√						
	M_8		√	√				√					
External													
	M_9		√	√					√				
	M_{10}		√	√						√			
Representation learning	M_{11}		√	√							√		
	M_{12}		√	√								√	
	M_{13}		√	√									√

धन्यवाद
Hindi
Hindi

多謝
Traditional Chinese

ขอบคุณ
Thai

Спасибо
Russian

Thank You
English

Gracias
Spanish

شكراً
Arabic

Obrigado
Brazilian Portuguese

Danke
German

Grazie
Italian

多谢
Simplified Chinese

Merci
French

நன்றி
Tamil
Tamil

ありがとうございました
Japanese

감사합니다
Korean