ICLEA: Interactive Contrastive Learning for Self-supervised Entity Alignment

Anonymous ACL submission

Abstract

Self-supervised entity alignment (EA) aims to link equivalent entities across different knowledge graphs (KGs) without seed alignments. The current SOTA self-supervised EA method draws inspiration from contrastive learning, originally designed in computer vision based on instance discrimination and contrastive loss, and suffers from two shortcomings. Firstly, it puts unidirectional emphasis on pushing sampled negative entities far away rather than pulling positively aligned pairs close, as is done in the well-established supervised EA. Secondly, KGs contain rich side information (e.g., entity description), and how to effectively leverage those information has not been adequately investigated in self-supervised EA. In this paper, we propose an interactive contrastive learning model for self-supervised EA. The model encodes not only structures and semantics of entities (including entity name, entity description, and entity neighborhood), but also conducts cross-KG contrastive learning by building pseudo-aligned entity pairs. Experimental results show that our approach outperforms previous best self-supervised results by a large margin (over 9% average improvement) and performs on par with previous SOTA supervised counterparts, demonstrating the effectiveness of the interactive contrastive learning for self-supervised EA.

1 Introduction

Knowledge Graphs (KGs) (e.g., DBpedia (Lehmann et al., 2015), YAGO (Pellissier-Tanon and et al.) and Wikidata (Vrandečić, 2014)) provide structural knowledge about the entities and relations in real world. These separately constructed KGs contain heterogeneous but complementary knowledge. Entity Alignment (EA) integrates the complementary knowledge in these KGs via identifying equivalent entities (Sun et al., 2020b), and thus benefits various knowledge-driven applications such as question answering (Cui et al., 2019) and recommendation system (Cao et al., 2019c).

The mainstream solutions in the literature are based on deep representation learning, which embed entities into a latent space, and then calculate the distance between embeddings as the evidence for EA. These approaches could be roughly classified into three lines: KG structure embedding-based methods (Chen et al., 2017; Sun et al., 2017a) utilize seed alignments to guide the representation learning of entity and relation; Graph-based methods (Wang et al., 2018a; Liu et al., 2020b) employ Graph Neural Networks (GNNs) (Kipf and Welling, 2016; Veličković et al., 2017) to enhance the entity embeddings by utilizing various entity information and propagating the seed alignments to the entire KG; BERT-based method (Tang et al., 2020) directly fine-tunes multilingual BERT and achieves the SOTA performance. Despite great performance achieved, they rely on seed alignments provided by humans during the training process. As the acquisition of seed alignments is usually time-consuming and labor-intensive (Liu et al., 2021c), it is hard to apply them to real-world EA scenarios.

Hence, self-supervised EA under the *unsupervised* setting (i.e., matching entities without any seed alignment) starts to attract research attention. One pioneer work is EVA (Liu et al., 2021b), which used visual semantic representations of entities to align entities in heterogeneous KGs in a fully unsupervised setting. Very recently, SelfKG (Liu et al., 2021c) drew inspirations from contrastive learning approaches, which originally targeted computer vision tasks by leveraging instance discrimination and contrastive loss (e.g., MoCo (He et al., 2020), SimCLR (Chen et al., 2020b)), and gave a contrastive learning framework for self-supervised EA. Despite their success, the following two critical issues still need further investigation.

How to find pivots for cross-KG interaction in a self-supervised EA framework? EA is essen-

tially building inter-KG links, and thus the cross-KG information interaction is critical. Existing methods achieve such interaction via explicit supervision (seed alignments) or implicit signals (images as visual pivots). SelfKG puts unidirectional emphasis on pushing sampled negative entities far away rather than pulling positively aligned pairs close, as is done in the well-established supervised EA. However, its negative sampling can only sample entities from the same source KG to avoid conflict, which blocks the direct cross-KG information interaction. Therefore, it is necessary to rethink how to design a self-supervised EA framework with direct cross-KG interaction.

How to learn comprehensive KG embeddings and jointly utilize entity structural and semantic information? Previous supervised methods have validated the effectiveness of various entity information (e.g., relation, attribute, description) (Wu et al., 2019a; Liu et al., 2020b; Tang et al., 2020). While in the un/self-supervised EA, limited features are focused, e.g., entity image (Liu et al., 2021b) and name label (Liu et al., 2021c). How to fully utilize and integrate the structural and semantic entity information in self-supervised EA remains another issue.

To address the above problems, we propose a model named Interactive Contrastive Learning for self-supervised Entity Alignment (**ICLEA**). We design a novel interactive contrastive learning mechanism in a self-supervised EA framework by constructing pseudo-aligned entity pairs to establish a direct information interaction channel for the two KGs. Inspired by previous supervised approaches, we introduce more important side information into self-supervised EA. Specifically, we separately encode entity name labels and descriptions by different pre-trained models, then organically combine them to provide powerful initial embeddings for entities. We propose a relation-aware neighborhood aggregator to better leverage the structural and semantic information brought by the KGs' relations.

The main contributions of our work are threefold: **1)** We design an interactive contrastive learning mechanism to achieve the direct cross-KG information interaction in a self-supervised EA framework. **2)** We propose appropriate modules to better utilize and integrate the structural and semantic entity information for self-supervised EA. **3)** Experimental results show that ICLEA outperforms the best self-supervised baseline by a large margin (over 9% average improvement), and performs on par with previous SOTA supervised methods while maintaining more stable model training. Our work significantly narrows the gap between supervised and self-supervised EA approaches[1].

2 Problem Definition

Definition 1 (Knowledge Graph)**.** *A knowledge graph is represented as* $\mathcal{G}=(\mathcal{E},\mathcal{R},\mathcal{T},\mathcal{S})$*, where each* $e \in \mathcal{E}$*,* $r \in \mathcal{R}$*,* $t=(e_i,r,e_j) \in \mathcal{T}$ *(* $e_i, e_j \in \mathcal{E}$ *) represent an entity, a relation and a fact respectively, and* $s(e)=\{n_e,d_e,a_e\} \in \mathcal{S}$ *denotes the side information of entity* e*, i.e., entity name, textual description and attribute value. We denote the set of one-hop neighbors of entity* e *as* $\mathcal{N}_e$ *of size* $|\mathcal{N}_e|$*, namely, the entities that are directly connected to* e *in KG* $\mathcal{G}$ *via fact triples, where* $\mathcal{N}_e = \{e' \mid (e,r,e') \in \mathcal{T} \wedge (e,e' \in \mathcal{E})\} \cup \{e' \mid (e',r,e) \in \mathcal{T} \wedge (e,e' \in \mathcal{E})\}$.

Knowledge graphs are often separately constructed for various goals, and thus contain heterogeneous but complementary knowledge. Entity alignment is to identify entities from different knowledge graphs (different languages or sources) that describe the same real-world object, and can be formally defined as follows.

Definition 2 (Entity Alignment)**.** *Given two different KGs* $\mathcal{G}_1$ *and* $\mathcal{G}_2$*, EA is to learn a ranking function* $f : \mathcal{E}_1 \times \mathcal{E}_2 \rightarrow \mathbb{R}$ *to calculate the similarity score between two entities, based on which we rank the correctly aligned entity* e_2 *as high as possible among all entities of* $\mathcal{E}_2$ *with a queried entity* $e_1 \in \mathcal{E}_1$*. Pre-aligned entity pairs* $\mathcal{I} = \{(e_1,e_2) \mid e_1 \in \mathcal{E}_1, e_2 \in \mathcal{E}_2, e_1 \leftrightarrow e_2\}$ *are provided, where* $\leftrightarrow$ *means that the* e_1 *from* $\mathcal{G}_1$ *and the* e_2 *from* $\mathcal{G}_2$ *are equivalent.*

According to the use of $\mathcal{I}$, the EA task is classified into (semi-)supervised and self-supervised or unsupervised settings. The (semi-)supervised setting leverages part of $\mathcal{I}$ as supervision signals for learning, while the self-supervised or unsupervised setting does not require any supervision.

3 The Proposed Approach

Although supervised EA methods achieve SOTA performance, their dependence on supervision signals limits real-world applications. In contrast, self-supervised methods show the ability to obtain competitive performance without any supervision, and

[1]We will release our code and data under an open source license after the review process.

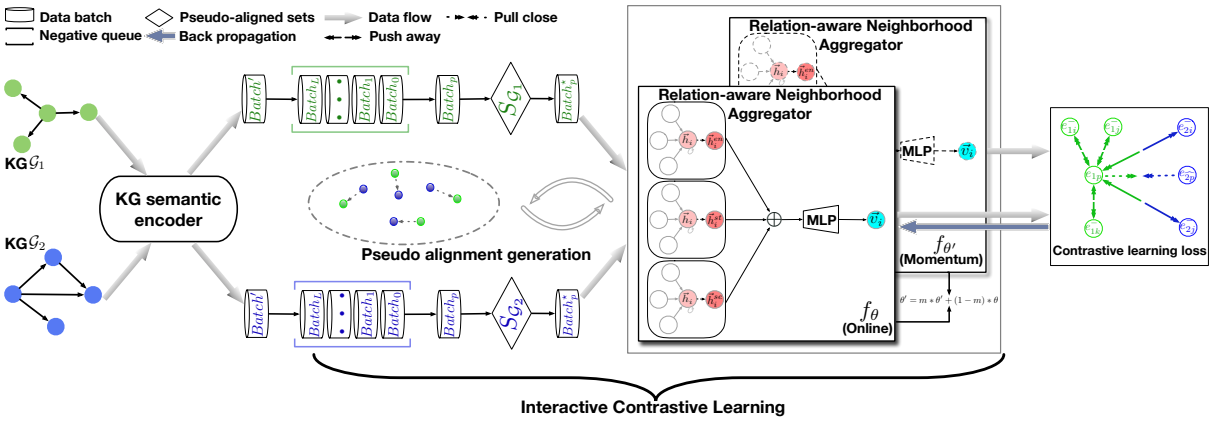


Figure 1: The overall framework of ICLEA. It consists of three main parts: (1) KG semantic encoder; (2) The relation-aware neighborhood aggregator; (3) Interactive contrastive learning mechanism.

have much better scalability. To fully exploit the potential of self-supervised EA approaches, two crucial issues remain to be addressed. 1.How to establish a direct cross-KG information interaction channel for a self-supervised EA framework instead of only performing intra-KG contrastive learning? 2.How to fully utilize and integrate the structural and semantic KG information?

To address the above issues, we introduce the interactive contrastive learning for self-supervised EA framework, which contains three parts: **1. A KG semantic encoder** is utilized to encode the names, descriptions of entities and relations for providing efficient initial embeddings. **2. A relation-aware neighborhood aggregator** is introduced to fully exploit relations' structural and semantic information so as to update entity representations. **3. An interactive contrastive learning mechanism** is proposed to perform contrastive learning between pseudo-aligned entity pairs in two KGs, thus promoting the direct learning of cross-KG interactions. Fig. 1 shows the overall framework of our proposed approach ICLEA.

3.1 KG Semantic Encoder

KG semantic encoder aims to fully capture the semantics of entities and relations by leveraging their names and descriptions. Since pre-trained language models (PLMs) have achieved remarkable progress in NLP, we decide to leverage LaBSE (Feng et al., 2020), a language-agnostic pre-trained sentence model, for encoding the name. Besides, we choose SentenceTransformers[2], a transformer-based sentence embedding framework, to obtain entity description embeddings. We choose them due to their outstanding empirical performance in capturing precise semantics for phrases and long sentences while helping us to cross the multilingual barrier.

The names of each entity e and relation r are usually composed of phrases. We utilize the tokenizer of LaBSE to obtain the tokens of the entity name n_e and relation name n_r, and get corresponding embeddings through LaBSE model $f_{\theta_{LaBSE}}$. Next, a mean pooling operation is applied to the embedding of each token, followed by an L_2 normalization.

$$\vec{h}_{n_x} = \left\| \text{Mean}(f_{\theta_{LaBSE}}(n_x)) \right\|_{L_2}, n_x \in \{n_e, n_r\}. \quad (1)$$

Moreover, entity description contains rich semantics that allows PLMs to encode with more context information. The entity description is usually composed of one or more sentences and contains multifaceted features related to the entity. For each description, we select the first L_{max} characters to feed into the SentenceTransformers model $f_{\theta_{ST}}$ and get the representation of entity description d_e,

$$\vec{h}_{d_e} = \left\| f_{\theta_{ST}}(d_e) \right\|_{L_2}. \quad (2)$$

Finally, we get the entity e's representation embedding $\vec{h}_e$ by concatenating entity name embedding $\vec{h}_{n_e}$ and entity description embedding $\vec{h}_{d_e}$, and more details are available in Appendix A.1,

$$\vec{h}_e = \text{Concat}(\vec{h}_{n_e}, \vec{h}_{d_e}). \quad (3)$$

3.2 Relation-Aware Neighborhood Aggregator

Relation-aware neighborhood aggregator aims to update entity embeddings by performing message passing with the help of KG relation information.

[2]https://github.com/UKPLab/sentence-transformers

Refer to 3.3, it has the online and momentum parts. These two parts have the same model structure, and they have different parameter updating strategies. We take the online part as an example to introduce this module. The relations can bring us two aspects of crucial information: ***structural*** — neighbor entities, which provide valuable context information for understanding the center node, and ***semantic*** — neighbor relations, which capture rich semantic information of edges adjacent to the given entity. In this work, we apply GAT (Veličković et al., 2017) as a backbone network due to its effectiveness.

Structural Aggregator. The aggregation of neighbor entities is performed by considering both entities' importance and their relations to the center node. One vanilla GAT aggregates all neighbor entity embeddings for the center node to model the importance of different entities, which treats all relations equally, formally,

$$\begin{aligned}\vec{h}_i^{en} &= \|_{k=1}^{K}\ \sigma\left(\sum_{j\in\mathcal{N}_i\cup\{e_i\}}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right),\\ \alpha_{ij} &= \frac{\exp\left(\sigma\left(\overrightarrow{\mathbf{q}}^T\left[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_j\right]\right)\right)}{\sum_{k\in\mathcal{N}_i\cup\{e_i\}}\exp\left(\sigma\left(\overrightarrow{\mathbf{q}}^T\left[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_k\right]\right)\right)},\end{aligned} \tag{4}$$

where $\mathbf{W}$ is the linear transformation's weight matrix. The attention mechanism is a single-layer feedforward neural network parametrized by a weight vector $\overrightarrow{\mathbf{q}}$. K is the multi-head attention number. σ is a nonlinear activation function such as $\mathrm{LeakyReLU}$. α_{ij} is the normalized attention coefficient. $\|$ is vector concatenation.

Another extended GAT is proposed to model the importance of different adjacent relations. It uses relation specific attention heads as relation-wise gates to control information flow from neighbor entities and obtains updated entity embedding $\vec{h}_i^{st}$,

$$\begin{aligned}\vec{h}_i^{st} &= \|_{k=1}^{K}\ \sigma\left(\sum_{j\in\mathcal{N}_i}\beta_{ij}^k\mathbf{W}^k\vec{h}_j\right),\\ \beta_{ij} &= \frac{\exp\left(\gamma_{ij}\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\gamma_{ik}\right)},\\ \gamma_{ij} &= \sigma\left(\left(\vec{r}_{ij}\mathbf{W}_{r_{ij}^1}+\mathbf{B}_{r_{ij}^1}\right)\mathbf{W}_{r_{ij}^2}+\mathbf{B}_{r_{ij}^2}\right),\end{aligned} \tag{5}$$

where $\vec{r}_{ij}$ is trainable relation specific embedding between entity i and entity j. $\mathbf{W}_{r_{ij}}$ and $\mathbf{B}_{r_{ij}}$ are relation's transformation and bias weight matrix.

Semantic Aggregator. Relation's name in KGs usually contains certain semantic information, which is helpful to obtain better entity representations. Semantic aggregator aims to aggregate and fuse neighbor relations' semantic information into the center entity. In this work, we apply a GAT that is same as Eq. 4 to fuse neighbor relation name embeddings into corresponding entity embeddings. Specifically, for each $\vec{h}_j$ in Eq. 4, we replace it with a average neighbor relation embedding $\widehat{h}_j = \frac{1}{K_{ij}}\sum_{x=0}^{K_{ij}}\vec{h}_{n_{r_x}}$, where n_{r_x} is the x-th relation's name between entity i and entity j, K_{ij} is the total number of relations between enitty i and entity j, and $\vec{h}_{n_{r_x}}$ is the relation name embedding, which can be calculated from Eq. 1. For $\vec{h}_k$ in Eq. 4, we replace it with $\widehat{h}_k$ using a similar method. In this way, we can obtain $\vec{h}_i^{se}$ of entity i based on its neighbor relation names.

Finally, we use a fully-connected layer to fuse three aspects of embeddings $\vec{h}_i^{en}$, $\vec{h}_i^{st}$, and $\vec{h}_i^{se}$ to obtain the final entity representation $\vec{v}_i$,

$$\vec{v}_i = \mathrm{MLP}(\mathrm{Concat}(\vec{h}_i^{en},\vec{h}_i^{st},\vec{h}_i^{se})). \tag{6}$$

3.3 Interactive Contrastive Learning

Interactive contrastive learning strategy is designed to learn direct cross-KG interactions for self-supervised EA. It mainly consists of three parts:(1) Momentum contrastive learning mechanism samples negative entities and pushes them far away from the positive one, thus aligned ones are relatively drawn close. (2) Negative sample queues store previous encoded batches as negative samples for the positive batch. (3) Interactive contrastive learning mechanism constructs pseudo-aligned sets during training and establishes cross-KG direct information interaction. It pulls the positive samples and pseudo-aligned entities closer, while pushing them far away from the negative samples in KGs.

Momentum Contrastive Learning Mechanism. Given a training KG $\mathcal{G}_1$'s initial entity embeddings $\mathbf{H_1} = \left\{\vec{h}_1,\vec{h}_2,\ldots,\vec{h}_{|\mathcal{E}_1|}\right\}$, self-supervised EA representation learning aims to learn an online embedding tranformation function f_θ that maps $\mathbf{H_1}$ to $\mathbf{V_1} = \left\{\vec{v}_1,\vec{v}_2,\ldots,\vec{v}_{|\mathcal{E}_1|}\right\}$ with $\vec{v}_x = f_\theta\left(\vec{h}_x\right)$, such that $\vec{v}_x$ best describes $\vec{h}_x$. Instance-wise contrastive learning achieves this objective by optimizing a contrastive loss. Following SelfKG, we use the Noise Contrastive Estimation (NCE) loss(Gutmann and Hyvärinen, 2010). In practice, we jointly optimize the NCE loss on both source KG $\mathcal{G}_1$ and target KG $\mathcal{G}_2$, defined as,

$$\mathcal{L}_{\mathrm{nce}_{\mathcal{G}_1}} = \sum_{x=1}^{|\mathcal{E}_1|} -\log\frac{\exp\left(\vec{v}_x\cdot\vec{v}'_x/\tau\right)}{\exp\left(\vec{v}_x\cdot\vec{v}'_x/\tau\right)+\sum_{k=1}^{\tau}\exp\left(\vec{v}_x\cdot\vec{v}'_k/\tau\right)}, \tag{7}$$

where, v_x and v'_x are positive embeddings for entity e_x, and v'_k includes r negative samples' embeddings, and τ is a temperature hyperparameter. These embeddings are obtained by feeding $\vec{h}_x$ to a momentum encoder parameterized by θ', $\vec{v}'_x = f_{\theta'}(\vec{h}_x)$, where θ' is a moving average of θ,

$$\theta' \leftarrow m \times \theta' + (1-m) \times \theta, m \in [0,1). \quad (8)$$

Negative Sample Queues. While performing momentum contrastive learning for EA, we need to sample negative entities from the same source KG to avoid the conflict by simply excluding the positive one. We maintain two negative queues for both KGs that store previous encoded batches as negative samples. As shown in the middle part of Fig. 1, when a new batch $Batch'$ arrives, we add it to the corresponding queue tail, and the head $Batch_0$ is dequeued as a positive sample batch $Batch_p$. In the early stage of training, we do not perform any parameter update until one of the negative queues reaches the predefined length $L+1$, where L for the number of previous batches used as negative samples and "1" for the dequeued positive batch. Let the numbers of the entities in KGs $|\mathcal{E}_1|$, $|\mathcal{E}_2|$, the batch size B and L is constraint by,

$$(L+1) \times B \leqslant \min(|\mathcal{E}_1|, |\mathcal{E}_2|). \quad (9)$$

Finally, the real number of negative samples used for each positive batch's entity is $(L+1) \times B - 1$.

Interactive Contrastive Learning Mechanism. This mechanism aims to build an direct information interaction channel for two KGs during training. We construct pseudo-aligned entity sets $S_{\mathcal{G}_1}$ and $S_{\mathcal{G}_2}$ for each source KG to the corresponding target KG at the beginning of each training epoch. Given initial embeddings of source and target KGs $\mathcal{G}_1$ and $\mathcal{G}_2$, $\mathbf{H_1}$ and $\mathbf{H_2}$, we feed them into the online encoder f_θ to obtain the corresponding embeddings $\mathbf{V_1}$ and $\mathbf{V_2}$. For each entity e_{1p} in $\mathcal{G}_1$ and e_{2q} in $\mathcal{G}_2$, we match the most similar entities $e_{\widehat{2p}}$ and $e_{\widehat{1q}}$ from the corresponding $\mathcal{G}_2$ and $\mathcal{G}_1$. We predefine a L_2 distance threshold λ, if $\mathrm{Dis}(e_{1p}, e_{\widehat{2p}})$ or $\mathrm{Dis}(e_{2q}, e_{\widehat{1q}})$ is less than λ, we add the pair of entities $\left\langle e_{1p}, e_{\widehat{2p}} \right\rangle$ or $\left\langle e_{2q}, e_{\widehat{1q}} \right\rangle$ to the corresponding pseudo-aligned set $S_{\mathcal{G}_1}$ or $S_{\mathcal{G}_2}$. We apply Faiss[3] for obtaining pseudo-aligned sets efficiently. It is worth noting that distinct from the previous bootstrapping strategy (Sun et al., 2018) that maintains high-confidence aligned entity pairs iteratively, our approach focuses on generating larger pseudo-aligned sets automatically, which liberates us from the caution of introducing a few noises. To incorporate pseudo-aligned sets into the contrastive learning process we introduce an additional NCE loss $\mathcal{L}_{\text{icl}}$ into the training process, and in Eq. 10 we take the example of the KG $\mathcal{G}_1$, where $\vec{v_x} = f_\theta(\vec{h_{1p}})$, $\vec{v_x}' = f'_\theta(\vec{h_{\widehat{2p}}})$, $\vec{h_{1p}}$ and $\vec{h_{\widehat{2p}}}$ is entity embeddings. The hyperparameter β balances the sensitiveness of the model to the negative sample $\vec{v}'_{1_k}$ and $\vec{v}'_{2_k}$ from different KGs. A pseudo-code of our algorithm is given in Appendix B,

$$\begin{aligned}\mathcal{L}_{\text{icl}_{\mathcal{G}_1}} = \sum_{x=1}^{|\mathcal{E}_1|} \beta \cdot \left[-\log \frac{\exp\left(\vec{v_x} \cdot \vec{v_x}'/\tau\right)}{\exp\left(\vec{v_x} \cdot \vec{v_x}'/\tau\right) + \sum_{k=1}^{r} \exp\left(\vec{v_x} \cdot \vec{v}'_{1_k}/\tau\right)} \right] \\ +(1-\beta) \cdot \left[-\log \frac{\exp\left(\vec{v_x} \cdot \vec{v_x}'/\tau\right)}{\exp\left(\vec{v_x} \cdot \vec{v_x}'/\tau\right) + \sum_{k=1}^{r} \exp\left(\vec{v_x} \cdot \vec{v}'_{2_k}/\tau\right)} \right]. \end{aligned} \quad (10)$$

Based on the above discussion, we can get the overall optimization goal $\mathcal{L}$ of ICLEA,

$$\mathcal{L} = \sum_{i \in \{1,2\}} \mathcal{L}_{\text{nce}_{\mathcal{G}_i}} + \mathcal{L}_{\text{icl}_{\mathcal{G}_i}}. \quad (11)$$

4 Experiments

In this section, we evaluate our proposed approach on DBP15K, a widely used benchmark for EA. We first introduce the experimental settings, then report the overall results, and finally conduct ablation studies as well as parameter sensitivity analyses.

4.1 Experimental Settings

Dataset. The DBP15K dataset is originally built by (Sun et al., 2017b), which includes three cross-lingual datasets extracted from DBpedia[4]. Each contains $15,000$ reference alignments between English (EN) and one of the other languages, i.e., Chinese (ZH), Japanese (JA) and French (FR). Tab. 2 presents the detailed statistics. To make wide comparisons, we also report the results on its translated version (Xu et al., 2019). Moreover, to make the comparisons fair, we use Google Translate to translate all non-English entity descriptions into English when using translated DBP15K in our experiments.

Evaluation Metrics. We use **Hits@N** as the evaluation metric. **Hits@N** means the proportion of correct entities that rank no larger than N (**N** is 1 and 10), and higher Hits@N indicates better performance. We further calculate the average Hits@1 of three subtasks to measure the overall performance.

[3] https://github.com/facebookresearch/faiss

[4] http://downloads.dbpedia.org/2016-04/

Model	DBP15K$_{ZH_EN}$		DBP15K$_{JA_EN}$		DBP15K$_{FR_EN}$		AVG Hits@1
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10	
Supervised							
MTransE	30.8	61.4	27.9	57.5	24.4	55.6	27.7
JAPE	41.2	74.5	36.3	68.5	32.4	66.7	36.6
MuGNN	49.4	84.4	50.1	85.7	49.5	87.0	49.7
BootEA	62.9	84.8	62.2	85.4	65.3	87.4	63.5
MRPEA	68.1	86.7	65.5	85.9	67.7	89.0	67.1
JEANS	71.9	89.5	73.7	91.4	76.9	94.0	74.2
GM-Align*	67.9	78.5	74.0	87.2	89.4	95.2	77.1
HGCN*	72.0	85.7	76.6	89.7	89.2	96.1	79.3
AttrGNN*	79.6	92.9	78.3	92.1	91.9	97.8	83.3
RNM*	84.0	91.9	87.2	94.4	93.8	95.4	88.3
EPEA*	88.5	95.3	92.4	96.9	95.5	98.6	92.1
CEAFF	79.5	-	86.0	-	96.4	-	87.3
HMAN	87.1	98.7	93.5	99.4	97.3	99.8	92.6
BERT-INT	96.8	99.0	96.4	99.1	99.2	99.8	97.5
Unsupervised or Self-supervised							
MultiKE	50.9	57.6	39.3	48.9	63.9	71.2	51.4
EVA	75.2	89.5	73.7	89.0	73.1	90.9	74.0
SelfKG	74.5	86.6	81.6	91.3	95.7	99.2	84.0
SelfKG*	82.9	91.9	89.0	95.3	95.9	99.2	89.3
ICLEA	**88.4**	**97.2**	**92.4**	**97.8**	**99.1**	**99.9**	**93.3**
ICLEA*	92.1	98.1	95.5	98.8	99.2	99.9	95.6

Table 1: Overall results on DBP15K. Methods marked with "*" use a translated version of DBP15K. The best results in supervised/unsupervised or self-supervised settings are marked in underline/bold.

Datasets		Ent.	Rel.	R-Tri.	Ent Alignments.
ZH_EN	ZH	19,388	1,700	70,414	15,000
	EN	19,572	1,322	95,142	15,000
JA_EN	JA	19,814	1,298	77,214	15,000
	EN	19,780	1,152	93,484	15,000
FR_EN	FR	19,661	902	105,998	15,000
	EN	19,993	1,207	115,722	15,000

Table 2: Statistics of the simplified DBP15K datasets.

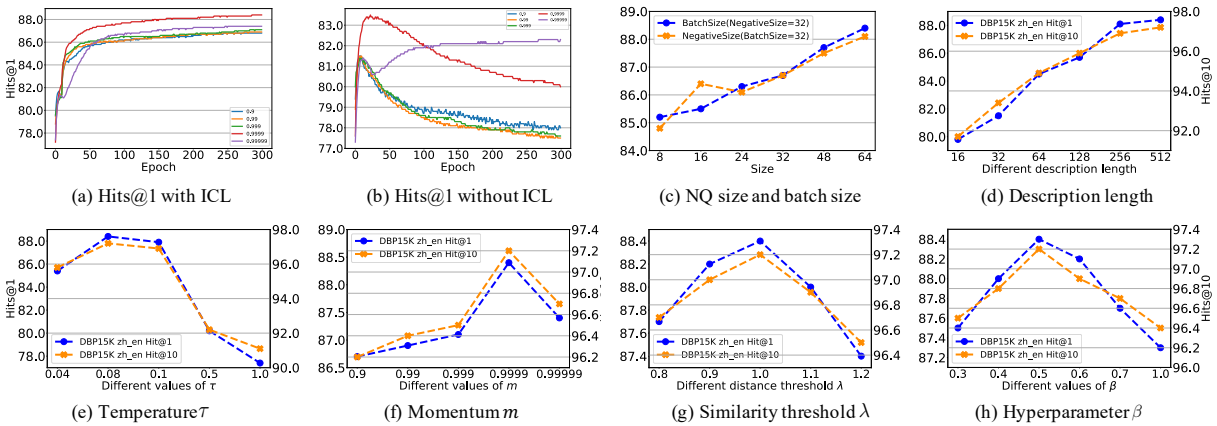
Baselines. We compare our **ICLEA** against two groups of baselines, i.e., semi-supervised and un/self-supervised EA approaches. The latter include MultiKE (Zhang et al., 2019), EVA and SelfKG, and the former are further divided into three branches: **1)** KG structure embedding-based methods that only use the KGs' structure information: MTransE (Chen et al., 2017), JAPE (Sun et al., 2017b), BootEA (Sun et al., 2018), MuGNN (Cao et al., 2019a), MRPEA (Shi and Xiao, 2019) and JEANS (Chen et al., 2020a), **2)** Graph-based methods that leverage GNNs to utilize various KG information: GM-Align(Xu et al., 2019), CEAFF (Zeng et al., 2019), HGCN(Wu et al., 2019b), AttrGNN (Liu et al., 2020b), RNM (Zhu et al., 2020b) and EPEA (Wang et al., 2020), **3)** BERT-based methods that directly use multilingual BERT to deal with EA: HMAN (Yang et al., 2019) and BERT-INT (Yang et al., 2019).

Implementation Details. Our model is implemented with Pytorch 1.7.0. We employ Adam as our optimizer with a learning rate $1e{-}6$ and gradually reduce it to maintain the stability of training. The number of training epochs is 300, the batch size is 64, momentum m is 0.9999, similarity threshold λ is 1.0, temperature τ is 0.08 and the size of negative queue is 32. We choose L_2 distance as entity embedding distance metric for pseudo alignment generation and set β to 0.5 in interactive contrastive learning. More implementation details are available in Appendix A.

4.2 Main Results

Tab. 1 lists the overall performance. For all baselines, we take the reported results from the original papers or SelfKG (Liu et al., 2021c). From the results, we have the following observations:

Comparisons with semi-supervised methods. ICLEA outperforms almost all previous supervised models on both original and translated DBP15K, and achieves results comparable to BERT-INT (the strongest baseline). They use multi-lingual BERT directly to help with entity alignment. ICLEA achieves the same excellent results as BERT-INT on the FR_EN subtask, where the BERT-INT's HIT@1 is 0.1% higher than that of ICLEA, while the ICLEA's HIT@10 is 0.1% higher than that of BERT-INT. The gap between ICLEA and BERT-INT is only 4.2% in average Hits@1, and it is further narrowed to 1.9% on the translated dataset (though BERT-INT does not translate dataset di-



(a) Hits@1 with ICL (b) Hits@1 without ICL (c) NQ size and batch size (d) Description length

(e) Temperature τ (f) Momentum m (g) Similarity threshold λ (h) Hyperparameter β

Figure 2: Study on negative queue (NQ) size, batch size, description length, temperature τ, momentum m, L_2 distance threshold λ and hyperparameter β on DBP15K$_{\text{ZH_EN}}$. (a), (b) present the test Hit@1 curve throughout the training epochs with and without ICL.

rectly, it uses multi-lingual BERT). It is worth mentioning that BERT-INT actually fine-tunes the whole multi-lingual BERT, whose parameter scale is 10 times larger than ICLEA (110M vs 11M). In a nutshell, ICLEA significantly narrows the gap between supervised and self-supervised EA.

Comparisons with un/self-supervised methods. In the unsupervised or self-supervised setting, ICLEA notably outperforms all those baselines with a large margin. Even compared with the strongest baseline SelfKG, ICLEA improves the average Hits@1 score by 9.3%/6.3% on the original/translated datasets. Note that the improvement is more significant in original datasets, especially in ZH_EN subtask, indicating that ICLEA is less language dependent and suitable for dealing with the cross-lingual entity alignment tasks.

Multi-lingual bias phenomenon. ICLEA exhibits multi-lingual bias consistent with most previous methods, i.e., performing best on the FR_EN and worst on the ZH_EN. Note that EVA has a relatively balanced performance on three datasets due to the introduction of language-independent entity images. This inspires us that multi-modal entity information utilization in self-supervised EA would be a promising future direction.

4.3 Ablation Study

We perform ablation study to evaluate the effectiveness of all model components. Accordingly, we implement four variants of ICLEA by removing entity description encoder (*w/o Des.*), relation-aware neighborhood aggregator (*w/o Rel.*), momentum contrastive learning (*w/o MCL*) and interactive contrastive learning (*w/o ICL*). Tab. 3 presents the results, which show that the removal of each component has a negative impact on the performance.

Specifically, *w/o Des.* has the largest negative impact on performance, demonstrating the effectiveness of entity description. *w/o Rel.* causes relatively larger decreases in Hits@1 than Hits@10, indicating the importance of relation and structure in distinguishing more ambiguous cases. *w/o ICL* brings a more notable performance degradation, illustrating the importance of the interactive contrastive learning mechanism for self-supervised EA. Besides, we further investigate how the performance changes during the model training with and without ICL. As shown in Fig. 2a and 2b, without ICL, the Hits@1 score reaches the peak after tens of epochs and then shows a sharp drop, while using ICL brings a more stable model training.

Model	DBP15K$_{\text{ZH_EN}}$		DBP15K$_{\text{JA_EN}}$		DBP15K$_{\text{FR_EN}}$	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
ICLEA	**88.4**	**97.2**	**92.4**	**97.8**	**99.1**	**99.9**
w/o Des.	80.4	91.4	87.3	94.1	97.3	99.5
w/o Rel.	87.0	96.7	91.5	97.7	98.8	99.9
w/o MCL	86.7	96.5	90.2	97.1	98.6	99.7
w/o ICL	83.6	93.7	88.8	96.2	97.5	99.8

Table 3: Ablation study of ICLEA on DBP15K.

4.4 Hyper-parameters Analyses

Negative queue size and batch size. For these two parameters, we perform grid search from 8 to 64. Fig. 2c shows that the performance exhibits a fluctuating upward trend when fixing batch size to 64 and increasing the size of the negative queue. Meanwhile, Fig. 2c also shows the performance increases steadily with larger batch size when setting

queue size to 64. While both expanding negative queue and batch size can improve the performance, a larger batch size usually brings more computational cost than the negative queue.

Entity description length. As shown in Fig. 2d, a longer description brings better performance because it can provide more semantic information. Considering the computation cost and the description length distribution in the datasets (see Appendix for details), we set the length to 512.

Temperature τ and momentum coefficient m. The temperature τ regulates the degree of attention to difficult samples (Wang and Liu, 2021) and the momentum coefficient m prevents sensitive update. We empirically choose both parameters from finite sets and present the results in Fig. 2e and Fig. 2f, which show that a relatively larger m (e.g., 0.9999) leads to better performance and $\tau = 0.08$ is a good cut-off point.

Distance threshold λ and hyperparameter β. Distance threshold λ controls the selection of pseudo alignments. A larger λ means more pseudo alignments for cross-KG interaction, while possibly introducing more noise. As shown in Fig. 2g, it is a good balance point when $\lambda = 1.0$. The hyperparameter β is the trade-off factor between the positive samples and negative samples from different knowledge graphs. As shown in Fig. 2h, we find that the sensitivity of the model is almost the same for both, and thus set β to 0.5.

5 Related Work

In this section, we briefly review representative efforts of the EA task to present the trends and promising directions in this domain.

EA task aims to identify the equivalent entities from heterogeneous knowledge graphs for better curation of them (Tang et al., 2006). The early efforts mostly consider this task under *semi-* or *fully supervised* setting (Tang et al., 2006; Li et al., 2008). From early KG structure embedding-based methods (Chen et al., 2017; Sun et al., 2018; Li et al., 2019) to GNNs-based methods (Wang et al., 2018b; Cao et al., 2019b; Zhu et al., 2020b,a; Mao et al., 2020a; Wong et al., 2020; Sun et al., 2020a; Mao et al., 2020b; Yu et al., 2021), a series of explorations have proven that structural information is indispensable in EA tasks. When the research comes into the era of pre-trained language models (PLMs), the multilingual PLMs further improve the SOTA performance of semantic EA models (Wu et al., 2019b; Tang et al., 2020) and attracted more attention for this field. Besides the high performance achieved by these (semi-) supervised models, researchers begin to consider the efficiency and practicality with new methods such as adversarial learning, active learning, etc (Pei et al., 2020; Zeng et al., 2021; Mao et al., 2021; Liu et al., 2021a).

However, the imperative need for labeled data still bridges the gap between these models and real-world applications, for annotating the data of entity alignment is naturally time-consuming and labor-intensive (Liu et al., 2020a). Therefore, several pioneer approaches (Zhang et al., 2019; Liu et al., 2021b, 2020a) attempt to solve the problem under *unsupervised* settings, i.e., matching the entities without any seed alignment. Strikingly, the success of *self-supervised* training in computer vision community (He et al., 2020; Chen et al., 2020b) provides new directions for unsupervised entity alignment models. As there are few attempts that launch self-supervised training in this task (Liu et al., 2021c), we recognize that it is urgent and purposeful to set up a paradigm so that the experience of existing EA methods can be adapted to this new direction. Therefore, we propose ICLEA, which jointly considers the structural and semantic information in the self-training process, and calls for more efforts in self-supervised EA.

6 Conclusion and Future Work

In this work, we propose a model named Interactive Contrastive Learning for self-supervised Entity Alignment — ICLEA. To better jointly utilize the entity's structural and semantic information, we separately encode entity name labels and descriptions with the help of PLMs and propose a relation-aware neighborhood aggregator to better leverage the structural and semantic information brought by KGs' relations. We design an innovative interactive contrastive learning mechanism by constructing pseudo-aligned entity pairs to establish a direct information interaction channel for the two KGs. Experimental results show that ICLEA performs on par with previous SOTA supervised counterparts and outperforms previous best self-supervised results by a large margin while maintaining more stable model training. We also present several promising directions, including the utilization of entity attributes, solving multi-lingual bias problem and applying prototypical contrastive learning, whose detailed discussions are in Appendix C.

References

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. 2019a. Multi-channel graph neural network for entity alignment. *arXiv preprint arXiv:1908.09898*.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. 2019b. Multi-channel graph neural network for entity alignment.

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019c. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*, pages 151–161.

Muhao Chen, Weijia Shi, Ben Zhou, and Dan Roth. 2020a. Cross-lingual entity alignment with incidental supervision. *arXiv preprint arXiv:2005.00171*.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2019. Kbqa: learning question answering over qa corpora and knowledge bases. *arXiv*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Jens Lehmann, Robert Isele, and Jakob et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *EMNLP*, pages 2723–2732.

Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. 2008. Rimom: A dynamic multistrategy ontology alignment framework. *TKDE*, 21(8):1218–1232.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*.

Bing Liu, Harrisen Scells, Guido Zuccon, Wen Hua, and Genghong Zhao. 2021a. Activeea: Active learning for neural entity alignment. *arXiv preprint arXiv:2110.06474*.

Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021b. Visual pivoting for (unsupervised) entity alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. 2021c. A self-supervised method for entity alignment. *arXiv preprint arXiv:2106.09395*.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020a. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 1(2).

Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020b. Exploring and evaluating attributes, values, and structure for entity alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6355–6364.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Are negative samples necessary in entity alignment? an approach with high performance, scalability and robustness. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1263–1273.

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020a. Mraea: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 420–428.

Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020b. Relational reflection entity alignment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1095–1104.

Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2020. Rea: Robust cross-lingual entity alignment between knowledge graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2175–2184.

Thomas Pellissier-Tanon and Weikum et al. Yago 4: A reason-able knowledge base.

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

M Sun, H Zhu, R Xie, and Z Liu. 2017a. Iterative entity alignment via joint knowledge embeddings [c]. In *International Joint Conference on Artificial Intelligence. AAAI Press*.

Zequn Sun, Wei Hu, and Chengkai Li. 2017b. Cross-lingual entity alignment via joint attribute-preserving embedding. pages 628–644.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI, vol.18*, pages 4396–4402.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020a. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 222–229.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020b. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11).

Jie Tang, Juanzi Li, Bangyong Liang, Xiaotong Huang, Yi Li, and Kehong Wang. 2006. Using bayesian decision for ontology mapping. *Journal of web semantics*, 4(4):243–262.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. Bert-int: A bert-based interaction model for knowledge graph alignment. In *IJCAI*, pages 3174–3180.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Vrandečić. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018a. Cross-lingual knowledge graph alignment via graph convolutional networks. pages 349–357.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018b. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357.

Zhichun Wang, Jinjian Yang, and Xiaoju Ye. 2020. Knowledge graph alignment with entity-pair embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1672–1680.

Chi Man Wong, Qiang Chen, Suhui Wu, and Wei Zhang. 2020. Global structure and local semantics-preserved embeddings for entity alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, volume 20, pages 3658–3664.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019a. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019b. Jointly learning entity and relation representations for entity alignment. *arXiv preprint arXiv:1909.09317*.

Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual knowledge graph alignment via graph matching neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3156–3161.

Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. *arXiv preprint arXiv:1910.06575*.

Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. 2021. Knowledge embedding based graph convolutional network. In *Proceedings of the Web Conference 2021*, pages 1619–1628.

Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Changjun Fan. 2021. Reinforced active entity alignment. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2477–2486.

Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2019. Collective embedding-based entity alignment via adaptive features.

Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. *arXiv preprint arXiv:1906.02390*.

Qi Zhu, Hao Wei, Bunyamin Sisman, Da Zheng, Christos Faloutsos, Xin Luna Dong, and Jiawei Han. 2020a. Collective multi-type entity alignment between knowledge graphs. In *Proceedings of The Web Conference 2020*, pages 2241–2252.

Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2020b. Relation-aware neighborhood matching model for entity alignment. *arXiv preprint arXiv:2012.08128*.

A Implementation and Training Details

We train and evaluate all our models on a machine with the specifications listed in Table 4.

Table 4: Hardware specifications of the used machine.

hardware	specification
RAM	252 GB
CPU	AMD® EPYC 7402 24-core 48-thread
GPU	NVIDIA® GeForce RTX 3090 (24 GB) × 8

A.1 Dataset

We process the data and conduct an experimental study on the DBP15K datasets, with the original datasets from (Sun et al., 2017b) and (Tang et al., 2020).

First, we simplify the names of entities and relations. The names of entities or relations in DBP15K usually have a string of useless prefixes, such as "http://dbpedia.org/resource/Jay_Chou" and "http://dbpedia.org/property/nation", we remove the useless prefixes and replace the underscores with empty spaces, leaving meaningful entity and relation names. (Tang et al., 2020) provides the description data of entities in DBP15K, and distribution of the number of entities based on description length in DBP15K dataset is shown in Figure 3, the description length of most entities is less than 256, but in order to fully consider all entities, we set the extracted description length to 512, while 512 is also the maximum length that can be extracted from most PLMs. After processing, the new entity names, relation names, and entity descriptions allow PLMs to handle them in an efficient and unambiguous way.

Then, we need to map each entity separately to a unique index in each pair of KGs. We use the DBP15K dataset provided in (Tang et al., 2020) as our original dataset and follow the indexes they created in our experiments, since they have already completed this processing step.

In terms of obtaining 1-hop neighbors, we treat KGs as undirected graphs, that means we use the relational triples in the datasets to find all entities connected to a given entity, regardless of the direction of the connection.

Finally, we rebuild the dataset and use the DataLoader of Pytorch's torch.utils.data package to package our data and create a batch process, with inputs containing the names of the central and adjacent entities, their descriptions and adjacency matrix.

The DBP15K dataset we used can be downloaded via the following links:

- https://github.com/nju-websoft/JAPE
- https://github.com/kosugi11037/bert-int

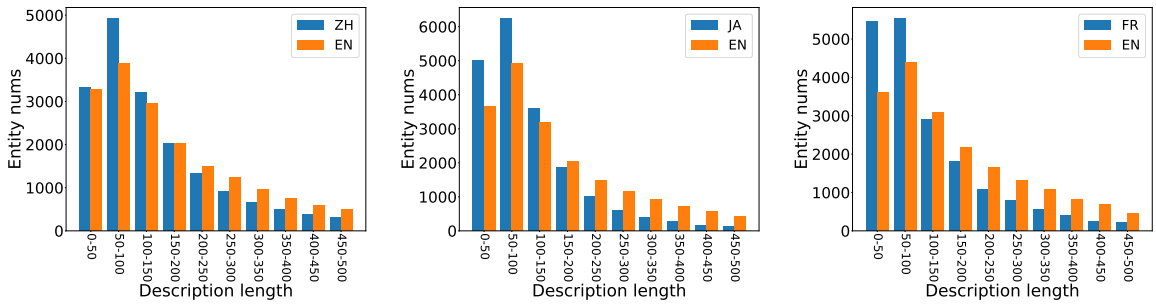
A.2 Training Details

Our model is implemented by Pytorch with version1.7.0, trained with following parameters:batch size=64, negative queue size=32, training epochs=300, description length=512, max neighbor num=15, m=0.9999, τ=0.08, warm-up epochs=24 and random seed=37. We use only NCE loss in the first 20-30 epochs to warm up the network, within this range, there is not much difference in the performance of the model so we set warm-up epochs to 24. The results prove that the model can acquire more accurate pseudo-entity pairs after the warm-up epochs.

We employ Adam as our optimizer with a small learning rate $1e{-}6$ and gradually reduce our learning rate to maintain the stability of training.

To extract information at the semantic level, we use LaBSE to obtain entity names and relation embeddings, LaBSE is a SOTA language-agnostic sentence embedding PLM which is trained on 109 different languages. SentenceTransformers are used to obtain entity description embeddings as input for subsequent layers. SentenceTransformers is a Python framework for SOTA sentence, text and image embeddings. We use the $paraphrase{-}multilingual{-}mpnet{-}base{-}v2$ model provided by SentenceTransformers to extract sentence representations, which trained on parallel data for 50+ languages, is selected to extract sentence representations because of its best performance tested in our experiments. In neighbor aggregator layer, both input_size and output_size =LaBSE_DIM + DESC_DIM, and in fully-connected layer, input_size=LaBSE_DIM × 5 and output_size = LaBSE_DIM × 3, where LaBSE_DIM and DESC_DIM are equal to 768.

Each batch consists of center entity embedding, neighbor embeddings, relation embeddings and their adjacency matrix. We use neighbor embeddings, relation embeddings to obtain two types of entity embeddings as the input of vanilla GAT. Extended GAT is used to obtain relation-aware entity representations. Finally, we use a fully-connected



(a) DBP15K$_{ZH_EN}$ (b) DBP15K$_{JA_EN}$ (c) DBP15K$_{FR_EN}$

Figure 3: Distribution of the number of entities based on description length in DBP15K.

layer to fuse three aspects of embeddings to obtain the final entity representation. It is worth noting that we perform normalization at the end of both GAT and fully-connected layer.

We randomly select 5% from the original training set as the validation set in order to early stopping. All experiments are conducted on a Ubuntu server with GPU(GeForceRTX 3090). Each evaluation is repeated five times with the same random seed and averaged results are reported.

A.3 Nearest Neighbor Search

To quickly obtain pseudo-entity pairs and evaluate the model, we use Faiss, a library for efficient similarity search and clustering of dense vectors.

We first apply IndexFlatL2 based on L2 distance as an indexer to construct indexes for the entities in source KG and target KG. Once the indices are built, the kd-tree algorithm used in Faiss allows for a fast similarity search task.

In the entity pair acquisition period, we treat the two KGs as equivalent and perform two-direction searches. If the L2 distance between an entity and its nearest neighbor is less than the threshold λ, we treat each entity and its top 1 nearest neighbor as a pair of pseudo-aligned entities.

In the model evaluation period, we search the top 1 and top 10 nearest entities in the target KG for each entity in the source KG, which is used to calculate Hits@1, Hits@10.

B Pseudo-code for Interactive Contrastive Learning

Algorithm 1 gives the overall technological process of our interactive contrastive learning for self-supervised entity alignment.

Algorithm 1: Interactive Contrastive Learning.

Input: encoder f_θ, training dataset X of source KG KG_1, target dataset Y of target KG KG_2, Similarity threshold λ, momentum m,

$\theta' = \theta$. // initialize momentum encoder as the encoder

while not MaxEpoch **do**

  /* Get pseudo-aligned entities */

  $V_i = f_\theta(X), V_j = f_\theta(Y)$ // get features for all data

  $Pair_{1\to2} = Faiss.search(V_i, V_j, 1, \lambda)$

  $Pair_{2\to1} = Faiss.search(V_j, V_i, 1, \lambda)$

  // Find the top 1 nearest entity with L2 distance less than λ for each entity

  $Pair_{all} = merge(Pair_{1\to2}, Pair_{2\to1})$

  /* Training */

  for x **in** Dataloader(X) **do** // load a minibatch x

    $v = f_\theta(x), v' = f'_\theta(x)$

    $v'' = f_{\theta'}(Pair_{all}[x])$ // forward pass through encoder and momentum encoder, Using the pseudo-aligned entity representation of x as positive example

    $neg\_value = f_{\theta'}(neg\_queue)$

    $Loss = \mathcal{L}_{\text{nce}}(v, v', neg\_value) + \mathcal{L}_{\text{icl}}(v, v'', neg\_value)$

    // calculate loss with Equation 10

    $\theta = \text{Adam}(Loss, \theta)$

    // update encoder parameters

    $\theta' = \text{m} * \theta' + (1 - \text{m}) * \theta$

    // update momentum encoder

C Future Works

Utilization of Entity Attributes Attributes are a series of attribute-value pairs, which are similar to relation-entity pairs. Aligned entities usually have similar attributes and values in different KGs. Some previous work has demonstrated that constructing representations for attributes can improve the accuracy of entity alignment. AttrGNN (Liu et al., 2020b) uses graph partition and attributed value encoder to deal with various types of attribute triples. But attribute values are of various types and are numerous, so it is difficult to construct attribute-related representations for entities in the context

of a self-supervised framework. How to combine attributes with a self-supervised EA framework to construct better representations for entities and perform self-supervised learning is one of the future research directions.

Solving Multi-lingual bias problem Multi-lingual bias can be interpreted as:A model prefers one language or language pair over others. There is some multi-lingual bias in the representation of sentences extracted using the cross-linguistic model, so Hits@1 varies in different language datasets, usually the best in FR_EN and the worst in ZH_EN. The earlier models that rely only on KG structure features to deal with EA such as JAPE, MTransE, etc., almost all models show the highest Hits@1 value in ZH_EN dataset, which shows on the other hand that multi-lingual bias does exist. EVA solves this problem by introducing image information and achieves a balanced performance on the three datasets. It uses image information to generate an initial seed dictionary and incorporates image features into entity representation. Designing a multi-modal fusion algorithm for EA, which organically fuses image, side information, and graph structures for our self-supervised learning, is also a work worthy of research.

Prototypical Contrastive Learning For a given entity, contrastive learning of EA treats different entities of the same knowledge graph as negative examples, regardless of their semantic similarity. This leads to some entities with similar semantics being pushed too far in the representation space. Some previous work learns semantic relationships between different samples by introducing prototype information, and has achieved significant improvements on CV tasks. PCL (Li et al., 2020) introduces prototypes as latent variables to help find the maximum-likelihood estimation of the network parameters in an Expectation-Maximization framework. Knowledge graph entities also have potential concept information, such as people, buildings, etc. How to let the model learn this information and assist in self-supervised EA is also a challenge.