Knowing the No-match: Entity Alignment with Dangling Cases

Zequn Sun[1] **Muhao Chen**[2,3] **Wei Hu**[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Department of Computer Science, University of Southern California, USA
[3]Information Sciences Institute, University of Southern California, USA
zqsun.nju@gmail.com, muhaoche@usc.edu, whu@nju.edu.cn

Abstract

This paper studies a new problem setting of entity alignment for knowledge graphs (KGs). Since KGs possess different sets of entities, there could be entities that cannot find alignment across them, leading to the problem of *dangling entities*. As the first attempt to this problem, we construct a new dataset and design a multi-task learning framework for both entity alignment and dangling entity detection. The framework can opt to abstain from predicting alignment for the detected dangling entities. We propose three techniques for dangling entity detection that are based on the distribution of nearest-neighbor distances, i.e., nearest neighbor classification, marginal ranking and background ranking. After detecting and removing dangling entities, an incorporated entity alignment model in our framework can provide more robust alignment for remaining entities. Comprehensive experiments and analyses demonstrate the effectiveness of our framework. We further discover that the dangling entity detection module can, in turn, improve alignment learning and the final performance. The contributed resource is publicly available to foster further research.

1 Introduction

Knowledge graphs (KGs) have evolved to be the building blocks of many intelligent systems (Ji et al., 2020). Despite the importance, KGs are usually costly to construct (Paulheim, 2018) and naturally suffer from incompleteness (Galárraga et al., 2017). Hence, merging multiple KGs through entity alignment can lead to mutual enrichment of their knowledge (Chen et al., 2020), and provide downstream applications with more comprehensive knowledge representations (Trivedi et al., 2018; Chen et al., 2020). Entity alignment seeks to discover identical entities in different KGs, such as English entity `Thailand` and its French counterpart `Thaïlande`. To tackle this important problem, literature has attempted with the embedding-based entity alignment methods (Chen et al., 2017; Wang et al., 2018; Cao et al., 2019; Fey et al., 2020; Wu et al., 2020a; Liu et al., 2020; Sun et al., 2020a). These methods jointly embed different KGs and put similar entities at close positions in a vector space, where the nearest neighbor search can retrieve entity alignment. Due to its effectiveness, embedding-based entity alignment has drawn extensive attention in recent years (Sun et al., 2020c).

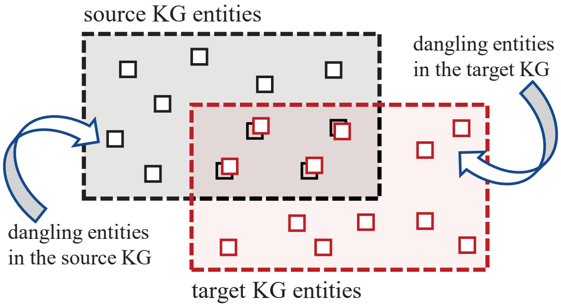

Figure 1: Illustration of entity alignment between two KGs with dangling cases. Paired red and black squares in the overlap region denote entity alignment while others are dangling entities without counterparts.

Nonetheless, to practically support the alignment of KGs as a real-world task, existing studies suffer one common problem of identifying entities without alignment across KGs (called *dangling entities*). Specifically, current methods are all built upon the assumption that any source entity has a counterpart in the target KG (Sun et al., 2020c), and are accordingly developed with learning resources that enforce the same assumption. Hence, given every entity in a source KG, a model always tends to predict a counterpart via the nearest neighbor search in the embedding space. However, since each KG may be independently created based on separate corpora (Lehmann et al., 2015) or contributed by different crowds (Speer et al., 2017; Carlson et al., 2010), it is natural for KGs to possess different sets

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 3582–3593
August 1–6, 2021. ©2021 Association for Computational Linguistics

of entities (Collarana et al., 2017), as illustrated in Fig. 1. Essentially, this problem overlooked in prior studies causes existing methods to fall short of distinguishing between matchable and dangling entities, hence hinders any of such methods to align KGs in a real-world scenario.

Towards more practical solutions of entity alignment for KGs, we provide a redefinition of the task with the incorporation of dangling cases (§2.1), as the *first contribution* of this work. Given a source entity, our setting does not assume that it must have a counterpart in the target KG as what previous studies do. Instead, conducting entity alignment also involves identifying whether the counterpart of an entity actually exists in another KG. Hence, a system to tackle this realistic problem setting of entity alignment is also challenged by the requirement for justifying the validity of its prediction.

To facilitate the research towards the new problem, the *second contribution* of this work is to construct a new dataset DBP2.0 for entity alignment with dangling cases (§2.2). As being discussed, existing benchmarks for entity alignment, including DBP15K (Sun et al., 2017), WK3L (Chen et al., 2017) and the more recent OpenEA (Sun et al., 2020c), are set with the constraint that any entity to be aligned should have a valid counterpart. We use the full DBpedia (Lehmann et al., 2015) to build a new dataset and the key challenge lies in that we need to guarantee the selected dangling entities actually do not have counterparts. We first extract two subgraphs with one-to-one entity alignment (i.e., all entities have counterparts). Then, we randomly remove some entities to make their left counterparts in the peer KG dangling.

Although embedding-based entity alignment has been investigated for several years, handling with dangling entities has not been studied yet. As the *third contribution*, we present a multi-task learning framework for the proposed task (§3). It consists of two jointly optimized modules for *entity alignment* and *dangling entity detection*, respectively. While the entity alignment module can basically incorporate any existing techniques from prior studies (Sun et al., 2020c), in this paper, we experiment with two representative techniques, i.e., relational embedding based (Chen et al., 2017) and neighborhood aggregation based (Sun et al., 2020b) methods. For dangling entity detection, our framework incorporates an auxiliary learning objective, which seeks to learn a confidence metric for the inferred entity alignment. The principle to realize such metric learning is that the embeddings of dangling entities should be isolated and are distant from others. According to this principle, we exploit several techniques to distinguish between matchable and dangling entities based on their distance distribution with their neighbors (§3), including nearest neighbor classification, marginal ranking and background ranking (Dhamija et al., 2018).

We conduct comprehensive experiments on the new DBP2.0 dataset, which demonstrate the proposed techniques to solve the dangling entity detection problem to different extents. Moreover, we observe that training the dangling detection model (marginal ranking) provides an effective indirect supervision that improves the detection of alignment for matchable entities. We hope our task, dataset and framework can foster further investigation of entity alignment techniques in the suggested real scenario, leading to more effective and practical solutions to this challenging but important problem.

2 Task and Dataset

We hereby describe the problem setting of our task and introduce the new dataset.

2.1 Task Definition

A KG is a set of relational triples $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ denote vocabularies of entities and relations, respectively. Without loss of generality, we consider entity alignment between two KGs, i.e., a source KG $\mathcal{K}_1 = (\mathcal{T}_1, \mathcal{E}_1, \mathcal{R}_1)$ and a target KG $\mathcal{K}_2 = (\mathcal{T}_2, \mathcal{E}_2, \mathcal{R}_2)$. Given a small set of seed entity alignment $\mathcal{A}_{12} = \{(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2 \| e_1 \equiv e_2\}$ along with a small set of source entities $\mathcal{D} \subset \mathcal{E}_1$ known to have no counterparts as training data, the task seeks to find the remaining entity alignment. Different from the conventional entity alignment setting (Sun et al., 2017), a portion (with an anticipated quantity) of entities in $\mathcal{E}_1$ and $\mathcal{E}_2$ may have no counterparts. Our training and inference stages take such dangling entities into consideration.

2.2 Dataset Construction

As discussed, previous testbeds for entity alignment do not contain dangling entities (Sun et al., 2017; Chen et al., 2018; Sun et al., 2020c). Therefore, we first create a new dataset to support the study of the proposed problem setting. Same as the widely used existing benchmark DBP15K (Sun

Datasets		# Entities	# Rel.	# Triples	# Align.
ZH-EN	ZH	84,996	3,706	286,067	33,183
	EN	118,996	3,402	586,868	
JA-EN	JA	100,860	3,243	347,204	39,770
	EN	139,304	3,396	668,341	
FR-EN	FR	221,327	2,841	802,678	123,952
	EN	278,411	4,598	1,287,231	

Table 1: Statistics of the DBP2.0 dataset.

et al., 2017), we choose DBpedia 2016-10[1] as the raw data source. Following DBP15K, we also use English (EN), French (FR), Japanese (JA) and Chinese (ZH) versions of DBpedia to build three entity alignment settings of ZH-EN, JA-EN and FR-EN. For each monolingual KG, the triples are extracted from the Infobox Data of DBpedia, where relations are not mapped to a unified ontology. The reference entity alignment data is from the inter-language links (ILLs) of DBpedia across these three bridges of languages. Such reference data is later used as alignment labels for training and testing, and also serves as references to recognize dangling entities.

Construction. The key challenge of building our dataset lies in that we need to ensure the selected dangling entities are indeed without counterparts. Specifcally, we cannot simply regard entities without ILLs as dangling ones, since the ILLs are also incomplete (Chen et al., 2017). Under this circumstance, we use a two-step dataset extraction process, which first samples two subgraphs whose entities all have counterparts based on ILLs, and randomly removes a disjoint set of entities in the source and target graphs to make their counterparts dangling. For the first step, we iteratively delete unlinked entities and their triples from the source and target KGs until the left two subgraphs are one-to-one aligned. In the second step for entity removal, while the removed entities are disjoint in two KGs, the proportion of the removed entities also complies with the proportion of unaligned entities in each KG.

Statistics and evaluation. Tab. 1 lists the statistics our dataset. The three entity alignment settings have different data scales and each is much larger than the same setting in DBP15K, thus can benefit better scalability analysis of models. For dangling entity detection, we split 30% of dangling entities for training, 20% for validation and others for testing. The splits of reference alignment follow the same partition ratio, which is also consistent with that of DBP15K to simulate the weak alignment nature of KGs (Chen et al., 2017; Sun et al., 2017). We also compare the degree distribution of matchable and dangling entities in our dataset against DBP15K in Fig. 7 of Appx. §A. We find the matchable and unlabeled entities in DBP15K have biased degree distribution, which has an adverse effect on dangling entity detection and leads to unreal evaluation. By contrast, in DBP2.0, matchable and dangling entities have similar degree distribution.

3 Entity Alignment with Dangling Cases

We propose a multi-task learning framework for entity alignment with dangling cases, as illustrated in Fig. 2. It has two jointly optimized modules, i.e., entity alignment and dangling entity detection. The entity alignment module takes as input relational triples of two KGs (for KG embedding) and seed entity alignment (for alignment learning). As for the detection of dangling entities, the module uses a small number of labeled dangling entities to jump-start the learning of a confidence metric for distinguishing between matchable and dangling entities. In the inference stage for entity alignment, our framework is able to first identify and remove dangling entities, then predict alignment for those that are decided to be matchable.

3.1 Entity Alignment

Our framework can incorporate any entity alignment technique. For the sake of generality, we consider two representative techniques in our framework. One technique is based on MTransE (Chen et al., 2017), which is among the earliest studies for embedding-based entity alignment. It employs the translational model TransE (Bordes et al., 2013) to embed KGs in separate spaces, meanwhile jointly learns a linear transformation between the embedding spaces to match entity counterparts. Specifically, given an entity pair $(x_1, x_2) \in \mathcal{A}_{12}$, let $\mathbf{x}_1$ and $\mathbf{x}_2$ be their embeddings learned by the translational model. MTransE learns the linear transformation induced by a matrix $\mathbf{M}$ by minimizing $\|\mathbf{M}\mathbf{x}_1 - \mathbf{x}_2\|$, where $\|\cdot\|$ denotes the L_1 or L_2 norm.

The other technique is from AliNet (Sun et al., 2020b), which is one of the SOTA methods based on graph neural networks. AliNet encodes entities by performing a multi-hop neighborhood aggregation, seeking to cope with heteromorphism of

[1] Downloaded from `https://wiki.dbpedia.org/downloads-2016-10`. The latest 2020 version has not provided updated data for some languages other than English when this study is conducted.

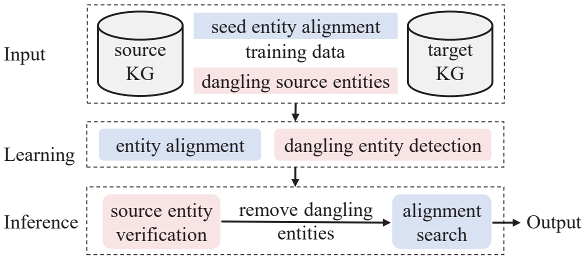

Figure 2: Framework of entity alignment w/ abstention.

their neighborhood structures. For alignment learning, different from MTransE that only minimizes the transformed embedding distance, AliNet additionally optimizes a margin-based ranking loss for entity counterparts with negative samples. Specifically, let x be a matchable source entity in the seed entity alignment, and x' is a randomly-sampled entity in the target KG, AliNet attempts to ensure $\|\mathbf{x} - \mathbf{x}'\| > \lambda_1 > 0$, where λ_1 is a distance margin.

3.2 Dangling Entity Detection

We propose three techniques to implement the dangling detection module based on the distribution of the nearest neighbor distance in embedding space.

3.2.1 NN Classification

This technique is to train a binary classifier to distinguish between dangling entities (labeled 1, i.e., $y = 1$) and matchable ones ($y = 0$). Specifically, we experiment with a feed-forward network (FFN) classifier. Given a source entity x, its input feature representation is the difference vector between its embedding $\mathbf{x}$ and its transformed NN embedding $\mathbf{x}_{\text{nn}}$ in the target KG embedding space[2]. The confidence of x being a dangling entity is given by $p(y = 1|x) = \text{sigmoid}(\text{FFN}(\mathbf{Mx} - \mathbf{x}_{\text{nn}}))$. Let $\mathcal{D}$ be the training set of dangling source entities and $\mathcal{A}$ denotes the set of matchable entities in the training alignment data. For every $x \in \mathcal{D} \cup \mathcal{A}$, we minimize the cross-entropy loss:

$$\begin{aligned}\mathcal{L}_x = -\big(&y_x \log(p(y = 1|x)) \\ &+ (1 - y_x)\log(1 - p(y = 1|x))\big),\end{aligned} \quad (1)$$

where y_x denotes the truth label for entity x. In a real-world entity alignment scenario, the dangling entities and matchable ones usually differ greatly in quantity, leading to unbalanced label distribution. In that case, we apply label weights (Huang et al., 2016) to balance between the losses for both labels.

3.2.2 Marginal Ranking

Considering that dangling entities are the noises for finding entity alignment based on embedding distance, we are motivated to let dangling entities have solitary representations in the embedding space, i.e., they should keep a distance away from their surrounding embeddings. Hence, we seek to put a distance margin between dangling entities and their sampled NNs. For every input dangling entity $x \in \mathcal{D}$, we minimize the following loss:

$$\mathcal{L}_x = \max(0, \lambda - \|\mathbf{Mx} - \mathbf{x}_{\text{nn}}\|), \quad (2)$$

where λ is a distance margin. This loss and the entity alignment loss (e.g., that of MTransE) conduct joint learning-to-rank, i.e., the distance between unaligned entities should be larger than that of aligned entities while dangling entities should have a lower ranking in the candidate list of any source entity.

3.2.3 Background Ranking

In the two aforementioned techniques, searching for the NN of an entity is time-consuming. Furthermore, selecting an appropriate value for the distance margin of the second technique is not trivial. Based on empirical studies, we find that the margin has a significant influence on the final performance. Hence, we would like to find a more efficient and self-driven technique. Inspired by the open-set classification approach (Dhamija et al., 2018) that lets a classifier equally penalize the output logits for samples of classes that are unknown to training (i.e. *background classes*), we follow a similar principle and let the model equally enlarge the distance of a dangling entity from any sampled target-space entities. This method is to treat all dangling entities as the "background" of the embedding space, since they should be distant from matchable ones. We also decrease the scale of the dangling entity embeddings to further provide a separation between the embeddings of matchable and dangling entities. For the dangling entity $x \in \mathcal{D}$, let X_x^v be the set of randomly-sampled target entities with size of v. The loss is defined as

$$\mathcal{L}_x = \sum_{x' \in X_x^v} \big|\lambda_x - \|\mathbf{Mx} - \mathbf{x}'\|\big| + \alpha\|\mathbf{x}\|, \quad (3)$$

where $|\cdot|$ denotes the absolute value and α is a weight hyper-parameter for balance. λ_x is the average distance, i.e., $\lambda_x = \frac{1}{v}\sum_{x' \in X_x^v} \|\mathbf{Mx} - \mathbf{x}'\|$. This objective can push the relatively close entities away from the source entity without requiring a pre-defined distance margin.

[2] We use *transformed nearest neighbor (NN)* to denote the the NN of a source KG entity after it is transformed to the target embedding space.

3.3 Learning and Inference

The overall learning objective of the proposed framework is a combination of the entity alignment loss (e.g., MTransE's loss) and one of the dangling entity detection loss as mentioned above. The two losses are optimized in alternate batches. More training details are presented in §4.1.

Like the training phase, the inference phase is also separated into dangling entity detection and entity alignment. The way of inference for dangling entities differs with the employed technique. The NN classification uses the jointly trained FFN classifier to estimate whether the input entity is a dangling one. The marginal ranking takes the preset margin value in training as a confidence threshold, and decides whether an entity is a dangling one based on if its transformed NN distance is higher than the threshold. The inference of background ranking is similar to that of marginal ranking, with only the difference, by its design, to be that the confidence threshold is set as the average NN distance of entities in the target embedding space. After detecting dangling entities, the framework finds alignment in the remaining entities based on the transformed NN search among the matchable entities in the embedding space of the target KG.

Accelerated NN search. The first and second techniques need to search NNs. We can use an efficient similarity search library Faiss (Johnson et al., 2017) for fast NN retrieval in large embedding space. We also maintain a cache to store the NNs of entities backstage and update it every ten training epochs.

4 Experiments

In this section, we report our experimental results. We start with describing the experimental setups (§4.1). Next, we separately present the experimentation under two different evaluation settings (§4.2-§4.3), followed by an analysis on the similarity score distribution of the obtained representations for matchable and dangling entities (§4.4). To faciliate the use of the contributed dataset and software, we have incorporated these resources into the OpenEA benchmark[3] (Sun et al., 2020c).

4.1 Experimental Settings

We consider two evaluation settings. One setting is for the proposed problem setting with dangling entities, for which we refer as the *consolidated evaluation setting*. We first detect and remove the dangling source entities and then search alignment for the left entities. For this evaluation setting, we also separately assess the performance of the dangling detection module. The other simplified setting follows that in previous studies (Sun et al., 2017, 2020c) where the source entities in test set all have counterparts in the target KG, so no dangling source entities are considered. In this *relaxed evaluation setting*, we seek to evaluate the effect of dangling entity detection on entity alignment and make our results comparable to previous work.

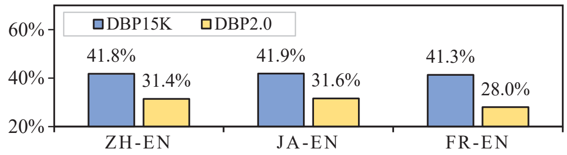

Figure 3: Average neighbor overlap ratio of aligned entities in DBP15K and our DBP2.0.

Evaluation Protocol. For the *relaxed evaluation setting*, given each source entity, the candidate counterpart list is selected via NN search in the embedding space. The widely-used metrics on the ranking lists are Hits@k ($k = 1, 10$, H@k for short) and mean reciprocal rank (MRR). Higher H@k and MRR indicate better performance.

For the *consolidated setting*, we report precision, recall and F1 for dangling entity detection. As for assessing the eventual performance of realistic entity alignment, since the dangling entity detection may not be perfect. it is inevitable for some dangling entities to be incorrectly sent to the entity alignment module for aligning, while some matchable ones may be wrongly excluded. In this case, H@k and MRR are not applicable for the consolidated entity alignment evaluation. Following a relevant evaluation setting for entity resolution in database (Mudgal et al., 2018; Ebraheem et al., 2018), we also use precision, recall and F1 as metrics. More specifically, if a source entity is dangling and is not identified by the detection module, the prediction is always regarded as incorrect. Similarly, if a matchable entity is falsely excluded by the dangling detection module, this test case is also regarded as incorrect since the alignment model has no chance to search for alignment. Otherwise, the alignment module searches for the NN of a source entity in the target embedding space and assesses if the predicated counterpart is correct.

Model Configuration. As described in §3.2, our dangling detection module has three variants, i.e.,

[3] https://github.com/nju-websoft/OpenEA

Methods	ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
	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	H@1	H@10	MRR
MTransE	.358	.675	.463	.353	.670	.461	.348	.661	.453	.342	.670	.452	.245	.524	.338	.247	.531	.342
w/ NNC	.350	.668	.457	.356	.664	.460	.340	.657	.441	.336	.630	.445	.253	.539	.343	.251	.536	.343
w/ MR	**.378**	**.693**	**.487**	**.383**	**.699**	**.491**	**.373**	**.686**	**.476**	**.374**	**.707**	**.485**	**.259**	**.541**	**.348**	**.265**	**.553**	**.360**
w/ BR	.360	.678	.468	.357	.675	.465	.344	.660	.451	.346	.675	.456	.251	.525	.342	.249	.531	.343
AliNet	.332	.594	.421	.359	.629	.451	.338	.596	.429	.363	.630	.455	.223	.473	.306	.246	.495	.329
w/ NNC	.321	.598	.415	.335	.608	.428	.330	.602	.422	.344	.627	.439	.212	.467	.294	.230	.476	.312
w/ MR	**.343**	**.606**	**.433**	**.364**	**.637**	**.459**	**.349**	**.608**	**.438**	**.377**	**.646**	**.469**	**.230**	**.477**	**.312**	**.252**	**.502**	**.335**
w/ BR	.333	.599	.426	.357	.632	.451	.341	**.608**	.431	.369	.636	.461	.214	.468	.298	.238	.487	.321

Table 2: Entity alignment results (relaxed setting) of MTransE and AliNet on DBP2.0.

Methods		ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
MTransE	NNC	.604	.485	.538	.719	.511	.598	.622	.491	.549	.686	.506	.583	.459	.447	.453	.557	.543	.550
	MR	.781	.702	.740	.866	.675	.759	.799	.708	.751	.864	.653	.744	.482	.575	.524	.639	.613	.625
	BR	**.811**	**.728**	**.767**	**.892**	**.700**	**.785**	**.816**	**.733**	**.772**	**.888**	**.731**	**.801**	**.539**	**.686**	**.604**	**.692**	**.735**	**.713**
AliNet	NNC	.676	.419	.517	.738	.558	.634	.597	.482	.534	.761	.120	.207	.466	.365	.409	.545	.162	.250
	MR	.752	.538	.627	.828	.505	.627	.779	.580	.665	**.854**	.543	**.664**	**.552**	**.570**	**.561**	**.686**	.549	**.609**
	BR	**.762**	**.556**	**.643**	**.829**	**.515**	**.635**	**.783**	**.591**	**.673**	.846	**.546**	.663	.547	.556	.552	.674	**.556**	**.609**

Table 3: Dangling entity detection results on DBP2.0.

NN classification (NNC), marginal ranking (MR), and background ranking (BR). We report the implementation details of the entity alignment module (w/ MTransE or AliNet) in Appendices B and C. We initialize KG embeddings and model parameters using the Xavier initializer (Glorot and Bengio, 2010), and use Adam (Kingma and Ba, 2015) to optimize the learning objectives with the learning rate 0.001 for MTransE and 0.0005 for AliNet. Note that we do not follow some methods to initialize with machine translated entity name embeddings (Wu et al., 2020a). As being pointed out by recent studies (Chen et al., 2021; Liu et al., 2021, 2020), this is necessary to prevent test data leakage. Entity similarity is measured by cross-domain similarity local scaling (Lample et al., 2018) for reduced hubness effects, as being consistent to recent studies (Sun et al., 2020b; Chen et al., 2021). We use a two-layer FFN in NNC. For MR, the margin is set as $\lambda = 0.9$ for MTransE and 0.2 for AliNet. BR randomly samples 20 target entities for each entity per epoch and $\alpha = 0.01$. Training is terminated based on F1 results of entity alignment on validation data.

4.2 Relaxed Evaluation

We first present the evaluation under the relaxed entity alignment setting based on Tab. 2. This setting only involves matchable source entities to test entity alignment, which is an ideal (but less realistic) scenario similar to prior studies (Sun et al., 2020c). We also examine if jointly learning to detect dangling entities can indirectly improve alignment.

As observed, MTransE, even without dangling detection, can achieve promising performance on DBP2.0. The results are even better than those on DBP15K as reported by Sun et al. (2017). We attribute this phenomenon to the robustness of this simple embedding method and our improved implementation (e.g., more effective negative sampling). By contrast, although we have tried our best in tuning, the latest GNN-based AliNet falls behind MTransE. Unlike MTransE that learns entity embeddings from a first-order perspective (i.e., based on triple plausibility scores), AliNet represents an entity from a high-order perspective by aggregating its neighbor embeddings, and entities with similar neighborhood structures would have similar representations. However, the dangling entities in DBP2.0 inevitably become spread noises in entity neighborhoods. To further probe into this issue, we count the average neighbor overlap ratio of aligned entities in DBP15K and our DBP2.0. Given an entity alignment pair (x_1, x_2), let $\pi(x_1)$ and $\pi(x_2)$ be the sets of their neighboring entities respectively, where we also merge their aligned neighbors as one identity based on reference entity alignment. Then the neighbor overlap ratio of x_1 and x_2 is calculated as $|\pi(x_1) \cap \pi(x_2)|/|\pi(x_1) \cup \pi(x_2)|$. We average such a ratio for both DBP15K and DBP2.0 as given in Fig. 3. We can see that the three settings'

	Methods	ZH-EN			EN-ZH			JA-EN			EN-JA			FR-EN			EN-FR		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
MTransE	NNC	.164	.215	.186	.118	.207	.150	.180	.238	.205	.101	.167	.125	.185	.189	.187	.135	.140	.138
	MR	.302	.349	.324	.231	.362	.282	.313	**.367**	**.338**	.227	**.366**	.280	.260	**.220**	**.238**	.213	**.224**	.218
	BR	**.312**	**.362**	**.335**	**.241**	**.376**	**.294**	**.314**	.363	.336	**.251**	.358	**.295**	**.265**	.208	.233	**.231**	.213	**.222**
AliNet	NNC	.121	.193	.149	.085	.138	.105	.113	.146	.127	.067	.208	.101	.126	.148	.136	.086	.161	.112
	MR	**.207**	**.299**	**.245**	**.159**	**.320**	**.213**	**.231**	**.321**	**.269**	**.178**	**.340**	**.234**	**.195**	**.190**	**.193**	.160	**.200**	.178
	BR	.203	.286	.238	.155	.308	.207	.223	.306	.258	.170	.321	.222	.183	.181	.182	**.164**	**.200**	**.180**

Table 4: Entity alignment results on DBP2.0.

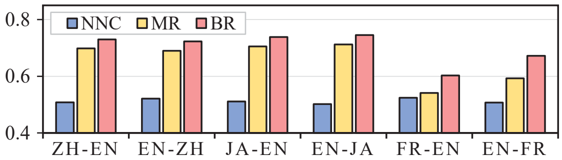

Figure 4: Accuracy of dangling entity detection.

overlap ratios in DBP2.0 are all much lower than those in DBP15K. Thus, DBP2.0 poses additional challenges, as compared to DBP15K, specifically for those methods relying on neighborhood aggregation. Based on results and analysis, we argue that methods performing well on the previous synthetic entity alignment dataset may not robustly generalize to the more realistic dataset with dangling cases. The performance of both MTransE and AliNet is relatively worse on FR-EN, which has more entities (i.e., larger candidate search space) and a low neighborhood overlap ratio (therefore, more difficult to match entities based on neighborhood similarity).

Meanwhile, we find that the dangling detection module can affect the performance of entity alignment. In details, MR consistently leads to improvement to both MTransE and AliNet. BR can also noticeably boost entity alignment on most settings. This shows that learning to isolate dangling entities from matchable ones naturally provides indirect help to discriminate the counterpart of a matchable entity from irrelevant ones. On the other hand, such indirect supervision signals may be consumed by the additional trainable parameters in NNC, causing its effect on entity alignment to be negligible. Overall, the observation here calls for more robust entity alignment methods and dangling detection techniques, and lead to further analysis (§4.3).

4.3 Consolidated Evaluation

We now report the experiment on the more realistic consolidated evaluation setting. Tab. 3 gives the precision, recall and F1 results of dangling entity detection, and the final entity alignment performance is presented in Tab. 4. In addition, Fig. 4 shows the accuracy of dangling entity detection. We analyze the results from the following aspects.

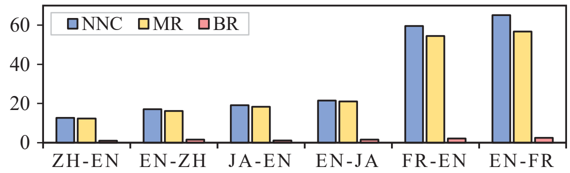

Figure 5: Average training time (seconds) of one epoch for dangling entity detection (MTransE variants).

Dangling entity detection. Regardless of which alignment module is incorporated, NNC performs the worst (e.g., the low recall and accuracy around 0.5) among the dangling detection techniques, whereas BR generally performs the best. NNC determines whether an entity is dangling based on the difference vector of the entity embedding and its NN, instead of directly capturing the embedding distance which is observed to be more important based on the results by the other two techniques. By directly pushing dangling entities away from their NNs in the embedding space, both MR and BR offer much better performance. Besides, BR outperforms MR in most cases. By carefully checking their prediction results and the actual distance of NNs, we find that the induced distance margin in BR better discriminates dangling entities from matchable ones than the pre-defined margin.

Efficiency. We compare the average epoch time of training the three dangling detection modules for MTransE in Fig. 5. We conduct the experiment using a workstation with an Intel Xeon E5-1620 3.50GHz CPU and a NVIDIA GeForce RTX 2080 Ti GPU. Since NNC and MR need to search for NNs of source entities, both techniques spend much more training time that is saved by random sampling in BR. Overall, BR is an effective and efficient technique for dangling entity detection.

Entity alignment. Generally, for both MTransE and AliNet variants, MR and BR lead to better entity alignment results than NNC. MR and BR

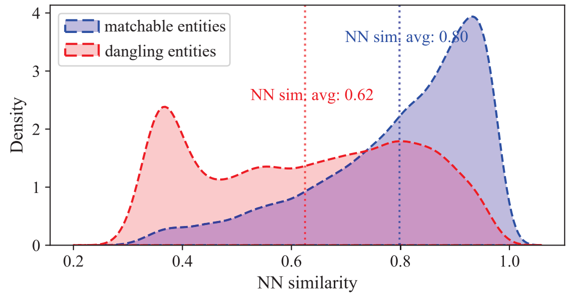

Figure 6: Kernel density estimate plot of the test matchable and dangling entities' similarity distribution with their nearest target neighbors in ZH-EN.

obtain higher precision and recall performance on detecting dangling entities as listed in Tab. 3, resulting in less noise that enters the entity alignment stage. By contrast, NNC has a low accuracy and thus introduces many noises. As BR outperforms MR in dangling detection, it also achieves higher entity alignment results than MR on most settings. We also notice that MR in a few settings, MR offer comparible or slightly better performance than BR. This is because MR can enhance the learning of alignment modules (see §4.2 for detailed analysis), thus delivering improvement to the final performance. MTransE variants generally excels AliNet variants in both entity alignment (see Tab. 2) and dangling entity detection (see Tab. 3) than AliNet, similar to the observation in §4.2.

Alignment direction. We find that the alignment direction makes a difference in both dangling entity detection and entity alignment. Using EN KG as the source is coupled with easier dangling detection than in other languages, as the most populated EN KG contributes more dangling entities and triples to training than other KGs. As for entity alignment, we find the observation to be quite the opposite, as using the EN KG as a source leads to noticeable drops in results. For example, the precision of MTransE-BR is 0.312 on ZH-EN, but only 0.241 on EN-ZH. This is because the EN KG has a larger portion of dangling entities. Although the dangling detection module performs well on the EN KG than on others, there are still much more dangling entities entering the alignment search stage, thus reducing the entity alignment precision. This observation suggests that choosing the alignment direction from a less populated KG to the more populated EN KG can be a more effective solution.

4.4 Similarity Score Distribution

To illustrate how well the BR technique distinguishes between matchable and dangling entities, we plot in Fig. 6 the distribution of similarity scores of each test entity and its NN. The plot illustrates BR has the expected effect to isolate dangling entities from their NNs, whereas matchable entities are generally placed closer to their NNs. Yet, we can still see a modest overlap between the two NN similarity distributions of dangling and matchable entities, and a number of dangling entities still have a quite large NN similarity. This also reveals the fact that the proposed problem setting of entity alignment with dangling cases has many remaining challenges that await further investigation.

5 Related Work

We discuss two topics of relevant work.

5.1 Entity Alignment

Embedding-based entity alignment is first attempted in MTransE (Chen et al., 2017), which jointly learns a translational embedding model and a transform-based alignment model for two KGs. Later studies generally follow three lines of improvement. (i) The first line improves the embedding technique to better suit the alignment task, including contextual translation techniques (Sun et al., 2019), long-term dependency techniques (Guo et al., 2019) and neighborhood aggregation (or GNN-based) ones (Wang et al., 2018; Cao et al., 2019; Li et al., 2019; Sun et al., 2020b,a; Fey et al., 2020). (ii) The second line focuses on effective alignment learning with limited supervision. Some leverage semi-supervised learning techniques to resolve the training data insufficiency issue, including self-learning (Sun et al., 2018; Mao et al., 2020) and co-training (Chen et al., 2018). (iii) Another line of research seeks to retrieve auxiliary or indirect supervision signals from profile information or side features of entities, such as entity attributes (Sun et al., 2017; Trisedya et al., 2019; Zhang et al., 2019; Pei et al., 2019), literals (Wu et al., 2019, 2020b; Liu et al., 2020), free text (Chen et al., 2021), pre-trained language models (Yang et al., 2019; Tang et al., 2020) or visual modalities (Liu et al., 2021). Due to the large body of recent advances, we refer readers to a more comprehensive summarization in the survey (Sun et al., 2020c).

5.2 Learning with Abstention

Learning with abstention is a fundamental machine learning, where the learner can opt to abstain from making a prediction if without enough decisive confidence (Cortes et al., 2016, 2018). Related techniques include thresholding softmax (Stefano et al., 2000), selective classification (Geifman and El-Yaniv, 2017), open-set classification with background classes (Dhamija et al., 2018) and out-of-distribution detection (Liang et al., 2018; Vyas et al., 2018). The idea of learning with abstention also has applications in NLP, such as unanswerable QA, where correct answers of some questions are not stated in the given reference text (Rajpurkar et al., 2018; Zhu et al., 2019; Hu et al., 2019).

To the best of our knowledge, our task, dataset, and the proposed dangling detection techniques are the first contribution to support learning with abstention for entity alignment and structured representation learning.

6 Conclusion and Future Work

In this paper, we propose and study a new entity alignment task with dangling cases. We construct a dataset to support the study of the proposed problem setting, and design a multi-learning framework for both entity alignment and dangling entity detection. Three types of dangling detection techniques are studied, which are based on nearest neighbor classification, marginal ranking, and background ranking. Comprehensive experiments demonstrate the effectiveness of the method, and provide insights to foster further investigation on this new problem. We further find that dangling entity detection can, in turn, effectively provide auxiliary supervision signals to improve the performance of entity alignment.

For future work, we plan to extend the benchmarking on DBP2.0 with results from more base models of entity alignment as well as more abstention inference techniques. Extending our framework to support more prediction tasks with abstention, such as entity type inference (Hao et al., 2019) and relation extraction (Alt et al., 2020), is another direction with potentially broad impact.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work is supported by the National Natural Science Foundation of China (No. 61872172), and the Collaborative Innovation Center of Novel Software Technology & Industrialization. Muhao Chen's work is supported by the National Science Foundation of United States Grant IIS-2105329.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1558–1569.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2787–2795.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1452–1461.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*.

Muhao Chen, Weijia Shi, Ben Zhou, and Dan Roth. 2021. Cross-lingual Entity Alignment with Incidental Supervision. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

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 *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3998–4004.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1511–1517.

Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Uppunda, Yizhou Sun, and Carlo Zaniolo. 2020. Multilingual knowledge graph completion via ensemble knowledge transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3227–3238.

Diego Collarana, Mikhail Galkin, Ignacio Traverso Ribón, Christoph Lange, Maria-Esther Vidal, and Sören Auer. 2017. Semantic data integration for knowledge graph construction at query time. In *Proceedings of the 11th IEEE International Conference on Semantic Computing (ICSC)*, pages 109–116.

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. 2018. Online learning with abstention. In *Proceedings of the 35th Inter-*

national Conference on Machine Learning (ICML), pages 1067–1075.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. 2016. Boosting with abstention. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1660–1668.

Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult. 2018. Reducing network agnostophobia. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 9175–9186.

Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467.

Matthias Fey, Jan Eric Lenssen, Christopher Morris, Jonathan Masci, and Nils M. Kriege. 2020. Deep graph matching consensus. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. 2017. Predicting completeness in knowledge bases. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 375–383.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 4878–4887.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2505–2514.

Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1709–1719.

Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Dongsheng Li. 2019. Read + Verify: Machine reading comprehension with unanswerable questions. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 6529–6537.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5375–5384.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *CoRR*, abs/2002.00388.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2723–2732.

Shiyu Liang, Yixuan Li, and R Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021. Visual Pivoting for (Unsupervised) Entity Alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

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

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. 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 (WSDM)*, pages 420–428.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, pages 19–34.

Heiko Paulheim. 2018. How much is a triple? estimating the cost of knowledge graph creation. In *Proceedings of the 17th International Semantic Web Conference (ISWC)*.

Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Proceedings of the World Wide Web Conference (WWW)*, pages 3130–3136.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 784–789.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, volume 31.

Claudio De Stefano, Carlo Sansone, and Mario Vento. 2000. To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 30(1):84–94.

Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020a. Knowledge association with hyperbolic knowledge graph embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5704–5716.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of the 16th International Semantic Web Conference (ISWC)*, pages 628–644.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4396–4402.

Zequn Sun, JiaCheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *Proceedings of the 18th International Semantic Web Conference (ISWC)*, pages 612–629.

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

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020c. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11):2326–2340.

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 *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3174–3180.

Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 297–304.

Rakshit Trivedi, Bunyamin Sisman, Xin Luna Dong, Christos Faloutsos, Jun Ma, and Hongyuan Zha. 2018. LinkNBed: Multi-graph representation learning with entity linkage. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 252–262.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. 2018. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 349–357.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. Jointly learning entity and relation representations for entity alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 240–249.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2020a. Neighborhood matching network for entity alignment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6477–6487.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2020b. Neighborhood matching network for entity alignment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6477–6487.

Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4431–4441.

Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. In *Proceedings of the 28th International Joint Conference (IJCAI)*, pages 5429–5435.

Haichao Zhu, Li Dong, Furu Wei, Wenhui Wang, Bing Qin, and Ting Liu. 2019. Learning to ask unanswerable questions for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4238–4248.

Appendices

A Degree Distribution

Fig. 7 shows the degree distribution of the matchable and dangling entities in our dataset against DBP15K. Although DBP15K contains some entities that are not labeled to have counterparts, by checking the ILLs in the recent update of DBpedia, we find many of these entities to have counterparts in the target KG. Hence, these entities in DBP15k cannot act as dangling entities that are key to the more realistic evaluation protocol being proposed in this work. From the comparison, we can see that these unlabeled entities in DBP15K have much fewer triples than matchable entities. This biased degree distribution will have an adverse effect on dangling entity detection and lead to unreal evaluation. By contrast, in our dataset, matchable and dangling entities have similar degree distribution.

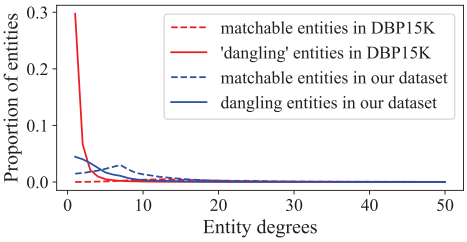

Figure 7: Degree distribution of matchable and dangling entities in DBP15K FR-EN and our FR-EN.

B Configuration of MTransE and AliNet

For entity alignment, we experiment with MTransE (Chen et al., 2017) and the SOTA method AliNet (Sun et al., 2020b). The implementation of our framework is extended based on OpenEA (Sun et al., 2020c). We adopt the truncated negative sampling method by BootEA (Sun et al., 2018) to generate negative triples for MTransE and negative alignment links for AliNet, which leads to improved performance. The embedding size is 128 for MTransE and 256 for AliNet. The batch size of MTransE is $20,480$ on ZH-EN and JA-EN, and $102,400$ on FR-EN. The batch size of AliNet is $8,192$ on ZH-EN and JA-EN, and $20,480$ on FR-EN. $\lambda_1 = 1.4$ in AliNet.

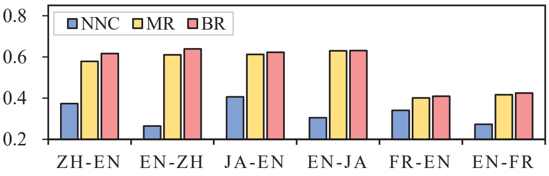

Figure 8: Recall@10 results of entity alignment.

C Hyper-parameter Settings

We select each hyper-parameter setting within a wide range of values as follows:

- Learning rate: $\{0.0001, 0.0002, 0.0005, 0.001\}$
- Embedding dimension: $\{64, 128, 256, 512\}$
- Batch size: $\{4096, 8192, 10240, 20480, 102400\}$
- # FNN layers: $\{1, 2, 3, 4\}$
- # Random targets: $\{1, 10, 20, 30, 40, 50\}$
- λ: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

D Recall@10 of Entity Alignment

Fig. 8 gives the recall@10 results of the MTransE variants with dangling entity detection in the consolidated evaluation setting. We can see that the recall@10 results on FR-EN are lower than that on ZH-EN and JA-EN, which is similar to the observation in entity alignment §4.3. From the results, we think existing embedding-based entity alignment methods are still far from being usable in practice.