

# Unsupervised Knowledge Graph Alignment by Probabilistic Reasoning and Semantic Embedding

Zhiyuan Qi<sup>1</sup>, Ziheng Zhang<sup>1</sup>, Jiaoyan Chen<sup>2</sup>, Xi Chen<sup>3\*</sup>,  
Yuejia Xiang<sup>3</sup>, Ningyu Zhang<sup>4</sup> and Yefeng Zheng<sup>1</sup>

<sup>1</sup>Tencent Jarvis Lab, Shenzhen, China

<sup>2</sup>Department of Computer Science, University of Oxford, UK

<sup>3</sup>Platform and Content Group, Tencent, Shenzhen, China

<sup>4</sup>Zhejiang University, Hangzhou, China

## Abstract

Knowledge Graph (KG) alignment is to discover the mappings (i.e., equivalent entities, relations, and others) between two KGs. The existing methods can be divided into the embedding-based models, and the conventional reasoning and lexical matching based systems. The former compute the similarity of entities via their cross-KG embeddings, but they usually rely on an ideal supervised learning setting for good performance and lack appropriate reasoning to avoid logically wrong mappings; while the latter address the reasoning issue but are poor at utilizing the KG graph structures and the entity contexts. In this study, we aim at combining the above two solutions and thus propose an iterative framework named PRASE which is based on probabilistic reasoning and semantic embedding. It learns the KG embeddings via entity mappings from a probabilistic reasoning system named PARIS, and feeds the resultant entity mappings and embeddings back into PARIS for augmentation. The PRASE framework is compatible with different embedding-based models, and our experiments on multiple datasets have demonstrated its state-of-the-art performance.

## 1 Introduction

A knowledge graph (KG) organizes entities, attributes, relations, and other information in a structured format [Hogan *et al.*, 2020]. One single KG is often incomplete while different KGs can complement each other to form a larger and more comprehensive KG via alignment, i.e., discovering equivalent entities, relations, and others across two KGs (a.k.a. mappings). Due to wide KG applications, KG alignment, especially entity alignment, has attracted massive attention.

With the development of machine learning, KG embeddings that encode entities, relations, and others into a low-dimensional vector space with their semantics preserved have become a powerful tool to manipulate and exploit KGs [Wang *et al.*, 2017]. Many embedding-based models have been proposed for entity alignment [Sun *et al.*, 2020], and usually

comply with the following paradigm. They first embed the to-be-aligned KGs into one vector space and then discover the mappings by calculating the vector distance or similarity.

Although the embedding-based models have achieved encouraging results, they are still limited in some aspects especially in industrial deployment. These models usually require a number of known mappings (a.k.a. alignment seeds) for training. However, seed annotation requires massive manual work, which may not be available in practice. An industrial evaluation study has shown the number and the sampling distribution of alignment seeds can dramatically influence the alignment performance [Zhang *et al.*, 2020]. The embedding-based models emphasize establishing expressive embeddings to capture entity features and then independently predict each mapping, ignoring the holistic analysis and logical consistency, which often leads to some false mappings.

In contrast, conventional KG or ontology alignment systems exploit various more traditional techniques such as logical reasoning and lexical matching. For example, the classic system LogMap [Jiménez-Ruiz and Grau, 2011] iteratively discovers mappings by lexical and graph matching, and repairs mappings by logical reasoning. PARIS [Suchanek *et al.*, 2012] is another representative conventional system that utilizes probabilistic reasoning and lexical matching. Specifically, after getting some initial mappings by matching with attributes such as names, PARIS expands the entity and relation mappings in each iteration by inferring the entity and relation equivalence with probabilistic reasoning. As no training is needed, these systems never rely on any alignment seeds, and are quite scalable and efficient. It is worth mentioning that PARIS and LogMap often outperform those embedding-based models according to the recent evaluation studies [Sun *et al.*, 2020; Zhang *et al.*, 2020]. On the other hand, these conventional systems use traditional lexical and graph matching techniques that are weak at exploiting and utilizing the graph structure and other contextual information.

In light of the complementarity between the embedding-based models and conventional systems, we propose to construct a unified framework that absorbs the advantages of both. The main challenge is to find the effective ways to make two completely different models work together. In this work, an unsupervised iterative framework named PRASE is proposed, which is composed of a probabilistic reason-

\*Corresponding author (jasonxchen@tencent.com)

ing (PR) module and a semantic embedding (SE) module. Specifically, the PR module initializes the mappings and infers logically consistent mappings with the entity embeddings from the SE module, while the SE module emphasizes learning high-quality cross-KG embeddings that encode the graph structures and the entity contexts. Note that the SE module is compatible with all kinds of embedding-based alignment models; while the PR module is currently developed based on the conventional system PARIS, but it can be extended to other reasoning-based systems such as LogMap.

The main contributions of this paper can be summarized as follows. First, an unsupervised KG alignment framework termed PRASE is proposed, which integrates probabilistic reasoning and semantic embedding using an iterative algorithm. To the best of our knowledge, this is the first to combine traditional reasoning techniques and state-of-the-art embedding techniques for KG alignment. Second, the PRASE framework has been implemented with PARIS and multiple different embedding-based models. Third, the PRASE framework has been evaluated on five widely used datasets and one industry dataset. The results show that PRASE achieves the state-of-the-art performance. On average, the F1-score of PRASE is 28.6% higher than the best embedding-based model and 5.96% higher than the best conventional system. Different settings of PRASE, such as the feedback from the SE module to the PR module and the iteration number, have also been studied.

## 2 Preliminaries

This section introduces the relevant background and related work. The problem formulation is first given, and then the conventional system PARIS and the embedding-based models are briefly introduced.

### 2.1 Problem Formulation

Let  $E$ ,  $R$ ,  $A$ , and  $V$  be the sets of entities, relations, attributes, and attribute values, respectively. A KG can be formulated as  $G = (E, R, A, V, T^R, T^A)$ , where  $T^R$  denotes the relation triples and  $T^A$  represents the attribute triples. Specifically,  $T^R$  and  $T^A$  are formalized as

$$\begin{aligned} T^R &= \{(h, r, t) | h, t \in E, r \in R\}, \\ T^A &= \{(e, a, v) | e \in E, a \in A, v \in V\}. \end{aligned}$$

Given two KGs  $G$  and  $G'$ , the problem of **Entity Alignment** is to discover the set of equivalent entity pairs (mappings) across  $G$  and  $G'$ , denoted as

$$\mathcal{Y} = \{(e, e') | e \equiv e', e \in E, e' \in E'\},$$

where the equivalence  $\equiv$  indicates that two entities refer to the same real-world object.

### 2.2 PARIS

Since the attribute triples are processed in a very similar way as the relation triples in PARIS, for convenience, we define  $E^+ = E \cup V$ ,  $R^+ = R \cup A$ , and  $T^+ = T^R \cup T^A$ . In order to derive mappings, PARIS measures the functionality

and inverse functionality of each relation, i.e.,

$$\begin{aligned} F(r) &:= \frac{|\{h | (h, r, t) \in T^+\}|}{|\{(h, t) | (h, r, t) \in T^+\}|}, \quad r \in R^+, \\ F^{-1}(r) &:= \frac{|\{t | (h, r, t) \in T^+\}|}{|\{(h, t) | (h, r, t) \in T^+\}|}, \quad r \in R^+, \end{aligned} \quad (1)$$

where  $|\cdot|$  denotes the set cardinality. The relation functionality and inverse functionality are used to determine the uniqueness of the head entity and tail entity, respectively. Take the relation *founder* as an example, if the relation functionality is equal to one (i.e.,  $F(\text{founder}) = 1$ ), it means that, given an organization, its founder can be uniquely determined. Note that the functionality and inverse functionality of the relations are invariant and can be computed in advance.

PARIS alternately computes the entity mappings and the subsumption relationships between relations. In computing the entity mappings, the probability of equivalence between two entities  $h$  and  $h'$ , denoted by  $P(h \equiv h')$ , is estimated as<sup>1</sup>

$$\begin{aligned} 1 - \prod_{(h, r, t) \in T^+, (h', r', t') \in T'^+} (1 - P(r' \subseteq r) F^{-1}(r) P(t \equiv t')) \\ \times (1 - P(r \subseteq r') F^{-1}(r') P(t \equiv t')), \end{aligned} \quad (2)$$

where  $P(r \subseteq r')$  represents the probability that  $r$  is a sub-relation of  $r'$ .  $P(r \subseteq r')$  is computed as

$$\frac{\sum_{h, t} (1 - \prod_{(h', r', t') \in T'^+} (1 - P(h \equiv h') P(t \equiv t')))}{\sum_{h, t} (1 - \prod_{h', t' \in E'^+} (1 - P(h \equiv h') P(t \equiv t')))}, \quad (3)$$

where  $(h, r, t) \in T^+$ . Similarly,  $P(r' \subseteq r)$  can also be computed. Note that the estimation of  $P(h \equiv h')$  relies on the subsumption relationships between relations, i.e.,  $P(r \subseteq r')$  and  $P(r' \subseteq r)$ , and vice versa. Therefore, PARIS adopts an iterative strategy for optimization. In the initialization phase,  $P(r \subseteq r')$  is set to a small value, e.g., 0.1;  $P(v \equiv v')$  ( $v \in V \subseteq E^+$  and  $v' \in V' \subseteq E'^+$ ) is set to 1 if  $v$  and  $v'$  are identical literals, and to 0 otherwise. Although this initialization method is simple, it has been shown quite effective. Other advanced methods, e.g., using the edit distance between text literals to score the equivalence of attribute values, can also be adopted. In each iteration, the equivalence probabilities of entities are computed based on Eq. (2), and then the probabilities for the subsumption relationships between relations are computed based on Eq. (3). The system self-iterates multiple times until convergence. Finally, the PARIS system outputs the entity mappings, denoted by  $\hat{\mathcal{Y}}^P$ , along with their probabilities (equivalence degrees), denoted by  $P^0(e \equiv e')$  with  $(e, e') \in \hat{\mathcal{Y}}^P$  and superscript  $^0$  indicating output. Please see [Suchanek *et al.*, 2012] for more details.

### 2.3 Embedding-based KG Alignment

Embedding-based KG alignment models usually work in the following two steps. First, the embeddings of KG components are learned based on some translational models (e.g., TransE [Bordes *et al.*, 2013]), graph neural networks [Kipf

<sup>1</sup>Since PARIS augments the to-be-aligned KGs with the inverse triples, the formula actually accounts for both head and tail entities.

and Welling, 2017] or other KG embedding algorithms [Guo *et al.*, 2019]. Entities of different KGs are embedded in the same vector space through strategies including parameter sharing, parameter swapping, embedding transformation, and embedding calibration. Then, entity mappings are predicted based on the similarity measure of the entity embeddings.

Take the typical embedding-based model MTransE [Chen *et al.*, 2017] as an example. First, MTransE adopts TransE to learn embeddings by minimizing the following loss:

$$\sum_{(h,r,t) \in T^R} \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| + \sum_{(h',r',t') \in T'^R} \|\mathbf{h}' + \mathbf{r}' - \mathbf{t}'\|,$$

where  $\|\cdot\|$  denotes the Euclidean norm operation;  $\mathbf{h}$ ,  $\mathbf{r}$ , and  $\mathbf{t}$  denote the  $m$ -dimensional embeddings of  $h$ ,  $r$ , and  $t$ , respectively. To ensure the entities are embedded in the same vector space, an embedding transformation strategy is adopted. Let  $S = \{(e, e') | e \in E, e' \in E'\}$  be the alignment seeds, it minimizes the loss given by  $\sum_{(e,e') \in S} \|Me - e'\|$ , where  $e$  and  $e'$  denote the entity embeddings, and  $M \in \mathbb{R}^{m \times m}$  is a transformation matrix. After learning  $M$ , entity embeddings of  $G$  are transformed into the entity vector space of  $G'$ , i.e.,  $\mathbf{e} := M\mathbf{e}$ . Finally, entity mappings, denoted by  $\tilde{\mathcal{Y}}^E$ , and their corresponding similarity scores in  $[0, 1]$ , denoted by  $S(e \equiv e')$  with  $(e, e') \in \tilde{\mathcal{Y}}^E$ , can be obtained by performing nearest neighbor search in the embedding space.

There are some other embedding-based models. For example, IPTransE [Zhu *et al.*, 2017] uses PTransE [Lin *et al.*, 2015] as the translational model. GCN-Align embeds the KGs using a graph convolutional network [Wang *et al.*, 2018]. BootEA adopts a bootstrapping strategy with an alignment editing method to reduce error accumulation, so as to overcome the lack of training data [Sun *et al.*, 2018]. MultiKE embeds the entities with their names, relations, and attributes considered via multi-view learning [Zhang *et al.*, 2019]. Please see [Sun *et al.*, 2020] and [Zhang *et al.*, 2020] for more comprehensive reviews.

### 3 Framework

In this section, the overview of the PRASE framework is given at first, and then its probabilistic reasoning module and semantic embedding module are introduced with details.

#### 3.1 PRASE Overview

Figure 1 shows the workflow of the PRASE framework that includes the Probabilistic Reasoning (PR) module and the Semantic Embedding (SE) module. First, the PARIS-based PR module is performed on the input KGs to discover entity mappings  $\tilde{\mathcal{Y}}^P$  with their probabilities  $P^0(e \equiv e')$ . Then, the highly confident entity mappings are selected as the alignment seeds  $S$ , and the SE module is trained subsequently based on these seeds. After training, the SE module predicts mappings on the entities that have not been aligned by the PR module (denoted by  $\tilde{\mathcal{U}}^P$ ). The resultant entity mappings  $\tilde{\mathcal{Y}}^E$  with the similarity scores  $S(e \equiv e')$  and the entity embeddings (denoted by  $\tilde{\mathcal{E}}^E = \{e | e \in E\} \cup \{e' | e' \in E'\}$ ) are then fed back to the PR module. The above process can be iterated  $K$  times, and the PR module finally outputs the entity mappings. The iterative algorithm is also shown in Algorithm 1.

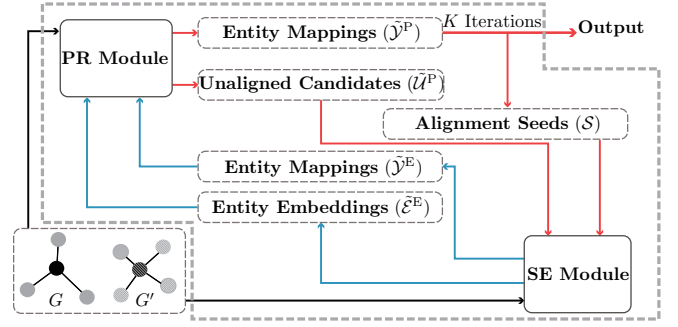


Figure 1: Overview of the PRASE framework.

#### 3.2 Probabilistic Reasoning Module

The PR module is constructed based on the PARIS system with the augmentation from KG embeddings. Let  $k$  be the iteration index of PRASE. In the initial iteration ( $k = 0$ ), the PR module first computes relation functionality and inverse functionality according to Eq. (1), and then alternatively computes mappings following Eqs. (2) and (3). In the subsequent iterations ( $k = 1, \dots, K$ ), the SE and the PR modules are alternately performed. In the  $k^{\text{th}}$  iteration ( $k > 0$ ), the PR module is initialized based on its previous output and the output of the SE module. Specifically, the probabilities of sub-relationships are set to the values of the  $(k - 1)^{\text{th}}$  iteration, i.e.,  $P_k^i(r \subseteq r') = P_{k-1}^o(r \subseteq r')$  with superscripts  $i$  and  $o$  indicating the meanings of input and output, respectively, and the probabilities of entity mappings are initialized as

$$P_k^i(e \equiv e') = \begin{cases} \alpha_1 P_{k-1}^o(e \equiv e') & \text{if } (e, e') \in \tilde{\mathcal{Y}}_{k-1}^P; \\ \alpha_2 S_k(e \equiv e') & \text{else if } (e, e') \in \tilde{\mathcal{Y}}_k^E \\ & \text{and } S_k(e \equiv e') > \delta_1; \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where  $P^i$  and  $P^o$  represent the input and the output mapping probabilities of the PR module, respectively;  $\alpha_1, \alpha_2 \in (0, 1]$  are two hyperparameters;  $\delta_1 \in [0, 1]$  is a threshold value; subscripts  $k$  and  $k-1$  indicate the variables in the  $k^{\text{th}}$  and the  $(k - 1)^{\text{th}}$  iterations, respectively. This customized initialization method for PARIS directly exploits the output of the SE module, through which the PR module could absorb the benefits captured by KG embeddings and further expand the mappings by reasoning. Note that the PR module is robust and can correct those unreliable entity mappings from the SE module, since it is based on probabilistic reasoning.

In addition to directly utilizing the output from the SE module, we also seek to fully exploit the semantic and structural information contained in the embeddings during the self-iterations of the PR module. The probability of the equivalence between two entities is updated according to a modified estimation, which is given by

$$P(e \equiv e') := (1 - \beta) \cdot \text{sim}(\mathbf{e}, \mathbf{e}') + \beta \left( 1 - \prod_{(e,r,t) \in T^+, (e',r',t') \in T'^+} (1 - P(r' \subseteq r) F^{-1}(r) P(t \equiv t')) \times (1 - P(r \subseteq r') F^{-1}(r') P(t \equiv t')) \right), \quad (5)$$

**Algorithm 1** PARIS-based PRASE Implementation**Input:** two KGs  $G$  and  $G'$ **Parameter:** iteration number  $K$ , hyperparameters  $\alpha_1$ ,  $\alpha_2$ , and  $\beta$ , thresholds  $\delta_1$ ,  $\delta_2$ , and  $\delta_f$ , similarity function  $\text{sim}(\cdot, \cdot)$ 

- 1: Initialize the PR module using Eq. (1);
- 2: Perform the PR module using Eqs. (2) and (3);
- 3: Generate  $\tilde{\mathcal{Y}}_0^p$  and  $\tilde{\mathcal{U}}_0^p$ ;
- 4: **while**  $k = 1, \dots, K$  **do**
- 5:   Generate  $\mathcal{S}_k$  based on  $\tilde{\mathcal{Y}}_{k-1}^p$  and  $\delta_2$ ;
- 6:   Train the SE module on  $\mathcal{S}_k$ ;
- 7:   Test the SE module on  $\tilde{\mathcal{U}}_{k-1}^p$ ;
- 8:   Generate  $\tilde{\mathcal{Y}}_k^e$  and  $\tilde{\mathcal{E}}_k^e$ ;
- 9:   Initialize the PR module using Eq. (4);
- 10:   Perform the PR module using Eqs. (5) and (3);
- 11:   Generate  $\tilde{\mathcal{Y}}_k^p$  and  $\tilde{\mathcal{U}}_k^p$ ;
- 12: **end while**
- 13: Generate  $\tilde{\mathcal{Y}}_f$  based on  $\tilde{\mathcal{Y}}_K^p$  and  $\delta_f$ ;

**Output:**  $\tilde{\mathcal{Y}}_f$ 

where  $e$  and  $e'$  denote the embeddings of entities  $e$  and  $e'$ , respectively;  $\text{sim}(\cdot, \cdot)$  is a similarity function;  $\beta \in (0, 1)$  is a trade-off hyperparameter balancing the embedding similarity and the probability estimated by Eq. (2). Eq. (5) replaces Eq. (2) in the original PARIS. It complements the PR module with the deep structural information learned from the SE model, and it can help the PR module estimate the probability of entity mappings more accurately. The function  $\text{sim}(\cdot, \cdot)$  measures the similarity between two embeddings, of which the range should be  $[0, 1]$ . A simple choice of  $\text{sim}(\cdot, \cdot)$  is cosine similarity, i.e.,  $\text{sim}(e, e') = (e \cdot e') / (\|e\| \cdot \|e'\|)$ .

After several self-iterations, the PR module converges and outputs a new round of alignment results  $\tilde{\mathcal{Y}}_k^p$  as well as the unaligned candidates  $\tilde{\mathcal{U}}_k^p$ . Specifically,  $\tilde{\mathcal{U}}_k^p = \{e | e \in E, \forall e' \in E', (e, e') \notin \tilde{\mathcal{Y}}_k^p\} \cup \{e' | e' \in E', \forall e \in E, (e, e') \notin \tilde{\mathcal{Y}}_k^p\}$  is a set of unaligned entities, which is later used as the test data for the SE module to compute the similarity score  $S_{k+1}(e \equiv e')$ . In the last iteration ( $k = K$ ), the final output entity mappings of the PRASE framework are given by the PR module, denoted as  $\tilde{\mathcal{Y}}_f = \{(e, e') | (e, e') \in \tilde{\mathcal{Y}}_K^p, P_K^o(e \equiv e') > \delta_f\}$ , where  $\delta_f \in [0, 1]$  is a threshold value.

### 3.3 Semantic Embedding Module

In the  $k^{\text{th}}$  iteration,  $\tilde{\mathcal{Y}}_{k-1}^p$  is refined to generate reliable alignment seeds  $\mathcal{S}_k$  for training the SE module. A feasible method to obtain  $\mathcal{S}_k$  is to set a threshold  $\delta_2 \in [0, 1]$ , and  $\mathcal{S}_k = \{(e, e') | (e, e') \in \tilde{\mathcal{Y}}_{k-1}^p, P_{k-1}^o(e \equiv e') > \delta_2\}$ . Although in most cases, there are still some incorrect entity mappings in  $\mathcal{S}_k$ , the abundant correct mappings in  $\mathcal{S}_k$  can still provide useful information. As mentioned before, the SE module outputs (i) entity mappings via a nearest neighbour search among  $\tilde{\mathcal{U}}_{k-1}^p$ , and (ii) entity embeddings  $\tilde{\mathcal{E}}_k^e$ . Since almost all embedding-based models can output entity mappings and embeddings, PRASE can choose almost any existing embedding-based model as the SE module. Algorithm 1 shows the whole process of the PRASE framework.

Dataset	KGs	#Ents.	Relation		Attribute	
			#Rels.	#Triples	#Attrs.	#Triples
EN-FR-100K	EN	100,000	379	649,902	364	503,922
	FR	100,000	287	561,391	468	431,379
EN-DE-100K	EN	100,000	323	622,588	326	560,247
	DE	100,000	170	629,395	189	793,710
D-W-100K	DB	100,000	318	616,457	328	467,103
	WD	100,000	239	588,203	760	878,219
D-Y-100K	DB	100,000	230	576,547	277	547,026
	YG	100,000	31	865,265	36	855,161
D-W-15K	DB	15,000	167	73,983	175	66,813
	WD	15,000	121	83,365	457	175,686
MED-BBK-9K	MED	9,162	32	158,357	19	11,467
	BBK	9,162	20	50,307	21	44,987

Table 1: Dataset statistics.

## 4 Evaluation

This section presents the evaluation of PRASE, whose code is available at <https://github.com/qizhyuan/PRASE-Python>.

### 4.1 Datasets

In the experiments, the following datasets are used, and the statistics of these datasets are presented in Table 1.

**OpenEA Datasets:** The OpenEA<sup>2</sup> datasets [Sun *et al.*, 2020] are constructed based on DBpedia, YAGO, and Wikidata. We use all their large-scale datasets of the version “V2” that has a more complex KG structure. They include two cross-lingual datasets (i.e., EN-FR-100K-V2 and EN-DE-100K-V2) and two cross-KG datasets (i.e., D-W-100K-V2 and D-Y-100K-V2). We also use a small dataset D-W-15K-V2, a relatively difficult dataset as reported by [Sun *et al.*, 2020]. In the following, the annotation “-V2” is omitted.

**Industry Dataset:** MED-BBK-9K is an industry dataset<sup>3</sup> proposed by [Zhang *et al.*, 2020], which is built from an authoritative medical KG and a KG extracted from Baidu Baike, a Chinese online encyclopedia.

### 4.2 Experimental Setting

The original implementation of PARIS is in Java.<sup>4</sup> We re-implemented PARIS in Python and updated it as the PR module such that it can easily work with the embedding-based models that are also implemented in Python. We adopt twelve competitive KG alignment methods as the baselines. They can be categorized into (i) embedding-based models that include MTransE [Chen *et al.*, 2017], IPTransE [Zhu *et al.*, 2017], GCNAlign [Wang *et al.*, 2018], BootEA [Sun *et al.*, 2018], RSN4EA [Guo *et al.*, 2019], IMUSE [He *et al.*, 2019], MultiKE [Zhang *et al.*, 2019], and RDGCN [Wu *et al.*, 2019], (ii) conventional systems that include PARIS and LogMap, and (iii) two simple matching models using either the edit distance (denoted by STR-Match) or the word embedding similarity (denoted by EMB-Match) between entity names. We adopt the implementations of the embedding-based models from OpenEA [Sun *et al.*, 2020] and also the same dataset

<sup>2</sup><https://github.com/nju-websoft/OpenEA><sup>3</sup><https://github.com/ZihengZZH/industry-eval-EA><sup>4</sup><http://webdam.inria.fr/paris/>

Model	EN-FR-100K			EN-DE-100K			D-W-100K			D-Y-100K			D-W-15K			MED-BBK-9K		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
MTransE	0.090	0.090	0.090	0.115	0.115	0.115	0.148	0.148	0.148	0.100	0.100	0.100	0.271	0.271	0.271	0.002	0.002	0.002
IPTransE	0.234	0.234	0.234	0.346	0.346	0.346	0.319	0.319	0.319	0.456	0.456	0.456	0.412	0.412	0.412	0.054	0.054	0.054
GCNAlign	0.257	0.257	0.257	0.375	0.375	0.375	0.353	0.353	0.353	0.620	0.620	0.620	0.506	0.506	0.506	0.057	0.057	0.057
BootEA	0.640	0.640	0.640	0.739	0.739	0.739	0.766	0.766	0.766	0.886	0.886	0.886	0.821	0.821	0.821	0.307	0.307	0.307
RSN4EA	0.495	0.495	0.495	0.639	0.639	0.639	0.634	0.634	0.634	0.841	0.841	0.841	0.723	0.723	0.723	0.195	0.195	0.195
IMUSE	0.461	0.461	0.461	0.457	0.457	0.457	0.431	0.431	0.431	0.629	0.629	0.629	0.581	0.581	0.581	0.186	0.186	0.186
MultiKE	0.642	0.642	0.642	0.661	0.661	0.661	0.319	0.319	0.319	0.853	0.853	0.853	0.495	0.495	0.495	0.410	0.410	0.410
RDGCN	0.715	0.715	0.715	0.766	0.766	0.766	0.421	0.421	0.421	0.911	0.911	0.911	0.623	0.623	0.623	0.301	0.301	0.301
PARIS	<b>0.981</b>	0.877	0.926	<u>0.988</u>	0.912	0.948	<b>0.931</b>	0.788	0.854	0.997	0.970	0.983	<b>0.950</b>	0.850	0.897	0.779	0.367	0.499
LogMap	0.541	0.709	0.614	0.729	0.729	0.729	-	-	-	0.954	0.912	0.933	-	-	-	<b>0.864</b>	0.441	<u>0.584</u>
STR-Match	0.754	0.720	0.737	0.790	0.760	0.775	0.419	0.306	0.354	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.606	0.419	0.495	0.545	0.495	0.519
EMB-Match	0.731	0.661	0.694	0.747	0.694	0.720	0.318	0.308	0.313	0.866	0.838	0.852	0.501	0.485	0.493	0.504	<u>0.504</u>	0.504
PRASE-BootEA	0.977	<b>0.932</b>	<b>0.954</b>	0.983	<u>0.948</u>	<u>0.965</u>	<u>0.927</u>	<b>0.855</b>	<b>0.890</b>	0.998	0.989	0.993	<u>0.948</u>	<b>0.900</b>	<b>0.923</b>	0.687	0.469	0.557
PRASE-MultiKE	<u>0.979</u>	<u>0.930</u>	<u>0.954</u>	<b>0.988</b>	<b>0.955</b>	<b>0.972</b>	0.922	0.804	<u>0.859</u>	<u>0.998</u>	<u>0.993</u>	<u>0.996</u>	<u>0.941</u>	<u>0.875</u>	<u>0.907</u>	<u>0.837</u>	<b>0.619</b>	<b>0.711</b>

Table 2: The overall results of the PRASE models in comparison with the baselines.

division: 20%, 10%, and 70% of the entity mappings for training, validation, and testing, respectively. Except for embedding-based models, other models are performed in an unsupervised setting. STR-Match and EMB-Match compute the similarity between entity names with a threshold of 0.5. In the overall result analysis, BootEA and MultiKE are used as the SE module and the resultant models are denoted by PRASE-BootEA and PRASE-MultiKE; while in the ablation study, more embedding-based models are evaluated. We set  $\alpha_1 = \alpha_2 = 1$ ,  $\beta = 0.8$ ,  $\delta_1 = \delta_2 = \delta_f = 0.1$ , and choose cosine similarity as  $\text{sim}(\cdot)$ . Since a small value of  $K$  is found to be sufficient for PRASE to demonstrate its effectiveness, we set  $K = 1$  in the experiments unless specified.

Our experiments are conducted on the workstation with an Intel Xeon E5 2.40 GHz CPU, 128 GB memory, an NVIDIA Tesla M40 GPU, and CentOS 7.2. The average time cost of our PARIS implementation on the four 100K datasets is 1697 seconds<sup>5</sup>, while the average time costs of BootEA and MultiKE are 24727 and 3198 seconds, respectively. Therefore, the time cost of the PRASE framework is acceptable even when several iterations are executed. As a comprehensive metric, F1-score is used to evaluate different models in the following experiments with the corresponding precision (P) and recall (R) reported as supplementary metrics. Since embedding-based models output a list of matching candidates for each entity, their precision, recall, and F1-score are actually equivalent to Hits@1. The best performance is **bolded** and the second best is underlined in our experiments.

### 4.3 Overall Results

Table 2 presents the experimental results (here, we directly use the results from [Sun *et al.*, 2020] for embedding-based models and LogMap on OpenEA datasets). The results show that the two PRASE models consistently outperform all the baselines on all the datasets except D-Y-100K in terms of recall and F1-score. STR-Match reaches full scores and outperforms all other models on D-Y-100K. However, the recall of STR-Match is significantly lower than the proposed models on the other five datasets. Actually, D-Y-100K is an easy dataset on which STR-Match achieves a perfect align-

ment, while the proposed models are almost perfect. PRASE-BootEA performs best on EN-FR-100K, D-W-100K, and D-W-15K, while PRASE-MultiKE performs best on EN-DE-100K and MED-BBK-9K. Besides, the F1-score of PRASE-MultiKE reaches 0.711 on MED-BBK-9K, significantly surpassing PARIS by 0.212. Compared with PARIS, the precision of the proposed models is slightly decreased on EN-FR-100K, D-W-100K, and D-W-15K, but the recall significantly increases. It reflects that although the incorrect entity mappings predicted by the SE module can have a negative impact on the performance, the useful information provided by the SE module can still help the PR module find more potential alignments. Table 2 also shows that the unsupervised PRASE models significantly outperform all the supervised embedding-based models with an average improvement of 28.6% in F1-score, which further confirms the effectiveness of the PRASE framework.

### 4.4 Ablation Studies

**Impact of Embedding Feedback:** In PRASE, the PR module uses both the entity mappings and the embeddings as the feedback from the SE module. To explore the role of these two types of feedback, two additional PRASE frameworks are evaluated: one uses only the mapping feedback (denoted with suffix “-M”) and the other uses only the embedding feedback (denoted with suffix “-E”). Table 3 shows the comparison results. Generally, all three PRASE frameworks achieve good results, and the framework using both types of feedback performs the best on almost all datasets. Besides, the precision of the PRASE framework using only entity embeddings is higher than or close to the baseline PARIS, while the precision of the PRASE framework using only entity mappings is lower in most cases. Furthermore, the use of entity mappings can significantly improve the recall but compromise the precision; the use of entity embeddings tends to maintain the precision, while the recall improvement is relatively limited. Table 3 also shows that using both types of feedback can significantly improve the recall while maintaining the precision. **Impact of Different SE Modules:** To analyze the impact of using different SE modules, four additional PRASE models are constructed based on MTransE, IPTransE, GCNAlign, and IMUSE. Table 4 shows their performance. It can be observed that all the PRASE models achieve a higher F1-score

<sup>5</sup>Note that the average running time by the Java implementation can be reduced to 89 seconds.

Model	EN-FR-100K			EN-DE-100K			D-W-100K			D-Y-100K			D-W-15K			MED-BBK-9K		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PARIS	0.981	0.877	0.926	0.988	0.912	0.948	<b>0.931</b>	0.788	0.854	0.997	0.970	0.983	<b>0.950</b>	0.850	0.897	0.779	0.367	0.499
PRASE-BootEA-M	0.976	0.912	0.943	0.983	0.934	0.958	0.928	<u>0.846</u>	<u>0.885</u>	0.997	0.984	0.991	0.948	<u>0.898</u>	<u>0.923</u>	0.692	0.458	0.552
PRASE-BootEA-E	<u>0.982</u>	0.912	0.945	0.987	0.933	0.960	0.928	0.821	0.871	0.997	0.978	0.988	0.949	0.876	0.911	0.761	0.435	0.554
PRASE-BootEA	0.977	<b>0.932</b>	<b>0.954</b>	0.983	<u>0.948</u>	0.965	0.927	<b>0.855</b>	<b>0.890</b>	0.998	0.989	0.993	0.948	<b>0.900</b>	<b>0.923</b>	0.687	0.469	0.557
PRASE-MultiKE-M	0.977	0.914	0.945	0.987	0.946	0.966	0.923	0.800	0.857	<u>0.998</u>	<u>0.990</u>	<u>0.994</u>	0.943	0.877	0.909	0.825	<u>0.593</u>	0.690
PRASE-MultiKE-E	<b>0.984</b>	0.903	0.942	<b>0.989</b>	0.930	0.959	<u>0.930</u>	0.798	0.859	0.997	0.978	0.988	0.949	0.863	0.904	<b>0.837</b>	0.493	0.621
PRASE-MultiKE	0.979	<u>0.930</u>	<u>0.954</u>	<u>0.988</u>	<b>0.955</b>	<b>0.972</b>	0.922	0.804	0.859	<b>0.998</b>	<b>0.993</b>	<b>0.996</b>	0.941	0.875	0.907	<u>0.837</u>	<b>0.619</b>	<b>0.711</b>

Table 3: The results of PRASE with different feedback settings from the SE module to the PR module.

Model	EN-FR-100K			EN-DE-100K			D-W-100K			D-Y-100K			D-W-15K			MED-BBK-9K		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PARIS	<b>0.981</b>	0.877	0.926	<u>0.988</u>	0.912	0.948	<b>0.931</b>	0.788	0.854	0.997	0.970	0.983	<b>0.950</b>	0.850	0.897	<u>0.779</u>	0.367	0.499
PRASE-MTransE	0.970	0.908	0.938	0.980	0.933	0.956	0.914	0.821	0.865	0.995	0.984	0.989	0.944	<u>0.886</u>	<u>0.914</u>	0.668	0.381	0.485
PRASE-IPTransE	0.979	0.918	0.947	0.985	0.938	0.961	0.927	0.825	0.873	0.997	0.986	0.992	0.945	0.879	<u>0.910</u>	0.650	0.429	0.517
PRASE-GCNAlign	<u>0.981</u>	0.900	0.939	0.986	0.924	0.954	0.927	0.803	0.861	0.997	0.975	0.986	<u>0.950</u>	0.863	0.904	0.676	0.418	0.517
PRASE-IMUSE	0.973	0.912	0.941	0.982	0.938	0.959	0.923	<u>0.829</u>	<u>0.873</u>	0.997	0.987	0.992	0.945	0.883	0.913	0.616	0.422	0.501
PRASE-BootEA	0.977	<b>0.932</b>	<b>0.954</b>	0.983	<u>0.948</u>	<u>0.965</u>	0.927	<b>0.855</b>	<b>0.890</b>	<u>0.998</u>	<u>0.989</u>	<u>0.993</u>	0.948	<b>0.900</b>	<b>0.923</b>	0.687	<u>0.469</u>	<u>0.557</u>
PRASE-MultiKE	0.979	<u>0.930</u>	<u>0.954</u>	<b>0.988</b>	<b>0.955</b>	<b>0.972</b>	0.922	0.804	0.859	<b>0.998</b>	<b>0.993</b>	<b>0.996</b>	0.941	0.875	0.907	<b>0.837</b>	<b>0.619</b>	<b>0.711</b>

Table 4: The comparison results of the PRASE models using different embedding-based models.

Dataset	MTransE	IPTransE	GCNAlign	IMUSE	BootEA	MultiKE
	Hits@1					
EN-FR-100K	0.281	0.439	0.328	0.382	<b>0.629</b>	<u>0.445</u>
EN-DE-100K	0.288	0.442	0.338	0.425	<b>0.576</b>	<u>0.575</u>
D-W-100K	0.283	0.328	0.318	<u>0.345</u>	<b>0.522</b>	0.117
D-Y-100K	0.574	0.806	0.745	0.811	<u>0.858</u>	<b>0.906</b>
D-W-15K	0.457	0.413	0.361	<u>0.481</u>	<b>0.614</b>	0.364
MED-BBK-9K	0.009	0.139	0.070	0.120	<u>0.233</u>	<b>0.433</b>

Table 5: The performance of different SE modules.

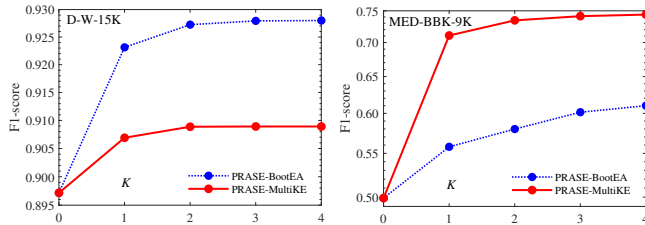


Figure 2: F1-score of PRASE models w.r.t.  $K$  on two datasets.

than PARIS on all datasets except for PRASE-MTransE on MED-BBK-9K, which indicates that the PRASE framework is robust and not sensitive to the selection of the SE module. However, different SE modules may bring different degrees of improvement. Table 5 shows the Hits@1 of the entity mappings predicted by different SE modules on  $\tilde{\mathcal{U}}^P$ . Their performance is positively correlated to the corresponding PRASE models. Specifically, BootEA and MultiKE outperform the other SE modules, and their corresponding PRASE models also perform better on most datasets. Besides, the Hits@1 of MTransE is only 0.9%, which is consistent with the degraded performance of PRASE-MTransE compared with PARIS. We additionally perform PRASE-MTransE-E on MED-BBK-9K, and the F1-score reaches 0.511, which shows that the entity embeddings can still provide useful information and benefit the PR module, even if the SE module has a poor alignment performance. Briefly, the benefit of the SE module is closely

related to the selected embedding-based model. It is recommended to choose an advanced embedding-based model such as BootEA or MultiKE. If the SE module has a poor alignment performance, it would be better to only use entity embeddings as the feedback to the PR module.

**Impact of Iteration Number:** To analyze the impact of the iteration number  $K$ , we additionally perform PRASE-BootEA and PRASE-MultiKE on D-W-15K and MED-BBK-9K with  $K = 2, 3$ , and 4. The results are shown in Figure 2, where the F1-score increases as  $K$  increases. Therefore, more iterations may help to improve the PRASE model. However, in practice, since the framework converges quickly and the embedding learning in each iteration costs much more time, it is suggested to set  $K$  to a small value.

## 5 Conclusion and Discussion

In this work, an unsupervised KG alignment framework PRASE has been proposed, which consists of a probabilistic reasoning module, a semantic embedding module, and an iterative algorithm for the interaction of the two modules. PRASE is compatible with most existing embedding-based models. Extensive experiments have been conducted on six datasets and verified the state-of-the-art performance of PRASE. More importantly, this work has shed light on the potential of unifying probabilistic reasoning and semantic embedding for KG alignment. It is therefore necessary to call for such hybrids for academic research and industrial applications. As side results, PRASE outputs relation mappings through the relations' subsumption relationships by PR module. This will be further studied in the future. Meanwhile, we plan to expand PRASE with other reasoning-based systems (e.g., LogMap) and enhance the interaction between the PR and SE modules by, e.g., injecting prior knowledge defined by the KGs' ontologies [Chen *et al.*, 2021]. We will also utilize the alignment of KGs to address KG refinement problems such as error detection and correction [Chen *et al.*, 2020].



## References

- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Annual Conference on Advances in Neural Information Processing Systems*, pages 2787–2795, Lake Tahoe, United States, December 2013.
- [Chen *et al.*, 2017] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1511–1517, Melbourne, Australia, August 2017.
- [Chen *et al.*, 2020] Jiaoyan Chen, Xi Chen, Ian Horrocks, Erik B. Myklebust, and Ernesto Jimenez-Ruiz. Correcting knowledge base assertions. In *Proceedings of The Web Conference 2020*, pages 1537–1547, April 2020.
- [Chen *et al.*, 2021] Jiaoyan Chen, Ernesto Jimenez-Ruiz, Ian Horrocks, Denvar Antonyrajah, Ali Hadian, and Jaehun Lee. Augmenting ontology alignment by semantic embedding and distant supervision. In *Proceedings of the 18th Extended Semantic Web Conference*, pages 1–16, Heraklion, Greece, June 2021.
- [Guo *et al.*, 2019] Lingbing Guo, Zequn Sun, and Wei Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2505–2514, Long Beach, United States, June 2019.
- [He *et al.*, 2019] Fuzhen He, Zhixu Li, Yang Qiang, An Liu, Guanfeng Liu, Pengpeng Zhao, Lei Zhao, Min Zhang, and Zhigang Chen. Unsupervised entity alignment using attribute triples and relation triples. In *Proceedings of the 24th International Conference on Database Systems for Advanced Applications*, pages 367–382, Chiang Mai, Thailand, April 2019.
- [Hogan *et al.*, 2020] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *arXiv preprint arXiv:2003.02320*, 2020.
- [Jiménez-Ruiz and Grau, 2011] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *Proceedings of the 10th International Semantic Web Conference*, pages 273–288, Bonn, Germany, October 2011.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, pages 1–14, Toulon, France, April 2017.
- [Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 705–714, Lisbon, Portugal, September 2015.
- [Suchanek *et al.*, 2012] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. PARIS: Probabilistic alignment of relations, instances, and schema. In *Proceedings of the 38th International Conference on Very Large Databases*, pages 157–168, Istanbul, Turkey, August 2012.
- [Sun *et al.*, 2018] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4396–4402, Stockholm, Sweden, July 2018.
- [Sun *et al.*, 2020] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. In *Proceedings of the 46th International Conference on Very Large Databases*, pages 2326–2340, Tokyo, Japan, August 2020.
- [Wang *et al.*, 2017] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.
- [Wang *et al.*, 2018] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 349–357, Brussels, Belgium, October 2018.
- [Wu *et al.*, 2019] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5278–5284, Macao, China, August 2019.
- [Zhang *et al.*, 2019] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Multi-view knowledge graph embedding for entity alignment. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5429–5435, Macao, China, August 2019.
- [Zhang *et al.*, 2020] Ziheng Zhang, Jiaoyan Chen, Xi Chen, Hualuo Liu, Yuejia Xiang, Bo Liu, and Yefeng Zheng. An industry evaluation of embedding-based entity alignment. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 179–189, Barcelona, Spain, December 2020.
- [Zhu *et al.*, 2017] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4258–4264, Melbourne, Australia, August 2017.