Unsupervised Knowledge Graph Alignment by Probabilistic Reasoning and Semantic Embedding

Zhiyuan Qi\textsuperscript{1}, Ziheng Zhang\textsuperscript{1}, Jiaoyan Chen\textsuperscript{2}, Xi Chen\textsuperscript{1,3*}, Yuejia Xiang\textsuperscript{1,3}, Ningyu Zhang\textsuperscript{1} and Yefeng Zheng\textsuperscript{1}

\textsuperscript{1}Tencent Jarvis Lab, Shenzhen, China
\textsuperscript{2}Department of Computer Science, University of Oxford, UK
\textsuperscript{3}Platform and Content Group, Tencent, Shenzhen, China
\textsuperscript{4}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 that 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. Due to wide KG applications, KG alignment, especially entity alignment, has attracted massive attention.

KG embeddings have become a powerful tool to exploit KGs by encoding entities, relations, and others into a low-dimensional vector space [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 (i.e., 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 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 system that utilizes probabilistic reasoning and lexical matching. Specifically, after getting 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 noting that PARIS and LogMap often outperform those embedding-based models according to the recent 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 Reasoning (PR) module and a Semantic Embedding (SE) module.
as the relation triples in PARIS, for convenience, we define the same real-world object. 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 contributions of this paper are threefold. 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 the state-of-the-art performance of PRASE. 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 the 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

$$ T^R = \{(h, r, t) | h, t \in E, r \in R\}, $$

$$ T^A = \{(e, a, v) | e \in E, a \in A, v \in V\}. $$

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.,

$$ F(r) := \frac{|\{(h, t) | (h, r, t) \in T^+\}|}{|\{(h, r, t) | h \in E^+, (h, r, t) \in T^+\}|}, \quad r \in R^+, $$

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

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., $F(`founder`) = 1$), it means that, given an organization, its founder can be uniquely determined. Note that the functionality and the inverse functionality of the relations are invariant for a given KG 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

$$ 1 - \prod_{(h, r, t) \in T^+} (1 - P(r') \subseteq r) P(t \equiv t') $$

$$ \times (1 - P(r \subseteq r') P^{-1}(r') P(t \equiv t')) $$

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

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

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., $0.1$, $P(v \equiv v')$ $(v \in V \subseteq E^+)$ 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 $\mathcal{Y}^p$, along with their probabilities (equivalence degrees), denoted by $P^p(e \equiv e')$ with $(e, e') \in \mathcal{Y}^p$ and superscript $p$ 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 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.

---

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

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

where $|| \cdot ||$ denotes the Euclidean norm operation; $h$, $r$, and $t$ 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., $e \equiv Me$. Finally, entity mappings, denoted by $\tilde{E}^k$, and their corresponding similarity scores in $[0, 1]$, denoted by $S(e \equiv e')$ with $(e, e') \in \tilde{E}^k$, can be obtained by performing nearest neighbor search in the embedding space.

There are some other embedding-based models. For example, GCN-Align embeds the KGs using graph convolutional networks [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; Zhang et al., 2020] and 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 $\tilde{E}^k$ with their probabilities $P^p(e \equiv e')$. Then, the highly confident entity mappings are selected as the alignment seeds $S$, and the PR 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 $\tilde{U}^p$). The resultant entity mappings $\tilde{E}^k$ with the similarity scores $S(e \equiv e')$ and the entity embeddings (denoted by $\tilde{E}^k = \{e|e \in E\} \cup \{e'|e' \in E'\}$) 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.

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 ($k = 1, \cdots, K$), the SE and the PR modules are alternately performed. In the $k$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 $(k - 1)$th iteration, i.e., $P_{k}(r \subseteq r') = P_{k-1}(r \subseteq r')$ with superscripts $^1$ and $^u$ indicating the meanings of input and output, respectively, and the probabilities of entity mappings are initialized as

$$P_{k}(e \equiv e') = \begin{cases} 
\alpha_1 P_{k-1}^p(e \equiv e') & \text{if } (e, e') \in \tilde{U}_{k-1}^p; \\
\alpha_2 S_{k}(e \equiv e') & \text{else if } (e, e') \in \tilde{E}_{k}^E \\
\beta_1 & \text{and } S_{k}(e \equiv e') > \delta_1; \\
0 & \text{otherwise},
\end{cases} \quad (4)$$

where $P^p$ and $P^u$ 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 $k$th and the $(k-1)$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

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

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
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 $\delta_t$ similarity function $\text{sim}(-, -)$

1. Initialize the PR module using Eq. (1);
2. Perform the PR module using Eqs. (2) and (3);
3. Generate $\hat{Y}_k^p$ and $U_k^p$;
4. while $k = 1, \ldots, K$ do
5. Generate $S_k$ based on $\hat{Y}_{k-1}^p$ and $\delta_2$;
6. Train the SE module on $S_k$;
7. Test the SE module on $\hat{U}_{k-1}^p$;
8. Generate $D_k^p$ and $\delta_k^e$;
9. Initialize the PR module using Eq. (4);
10. Perform the PR module using Eqs. (5) and (3);
11. Generate $\hat{Y}_k^p$ and $U_k^p$;
12. end while
13. Generate $\hat{Y}_t$ based on $\hat{Y}_K^p$ and $\delta_t$;

**Output:** $\hat{Y}_t$

---

### Table 1: Dataset statistics.

| Dataset     | KGs | #Ents. | Relation | Attribute | #Rel. | #Triples | #Atts. | #Triples |
|-------------|-----|--------|----------|-----------|-------|----------|--------|----------|
| EN-FR-100K  | EN  | 100,000| 379      | 649,018   | 364   | 503,922  |
|             | FR  | 100,000| 287      | 561,391   | 468   | 167,379  |
| 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      | 73,982    | 175   | 66,813   |
|             | WD  | 15,000 | 121      | 83,365    | 457   | 175,686  |
| MED-BBBK-9K | MED | 9,162  | 32       | 158,357   | 19    | 11,467   |
|             | BBK | 9,162  | 20       | 50,307    | 21    | 44,987   |

---

4 Evaluation

This section presents the evaluation of PRASE, and the 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 datasets are constructed based on DBpedia, YAGO, and Wikidata [Sun et al., 2020]. We use all their large-scale datasets of the version "V2" that has more complex KG structures. 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-BBBK-9K is an industry dataset 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. We reimplemented 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 [Wang et al., 2019], (ii) conventional systems including 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] with the same dataset 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.

---

3.3 Semantic Embedding Module

In the $k^{th}$ iteration, $\hat{Y}_{k-1}^p$ is refined to generate reliable alignment seeds $S_k$ for training the SE module. A feasible method to obtain $S_k$ is to set a threshold $\delta_2 \in [0, 1)$, and $\bar{S}_k = \{(e, e') | (e, e') \in \hat{Y}_{k-1}^p, P_k^p(e \equiv e') > \delta_t\}$. Although in most cases, there are still some incorrect entity mappings in $\bar{S}_k$, the abundant correct mappings in $\bar{S}_k$ can still provide useful information. As mentioned before, the SE module outputs (i) entity mappings via a nearest neighbour search among $\hat{U}_{k-1}^p$, and (ii) entity embeddings $\hat{e}_k^p$. 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.
Table 2: The overall results of the PRASE models in comparison with the baselines.

| 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         |
| PRASE-BootEA-M  | 0.981      | 0.877      | 0.926    | 0.986    | 0.921   | 0.948      |
| PRASE-BootEA-E  | 0.981      | 0.877      | 0.926    | 0.986    | 0.921   | 0.948      |
| PRASE-MultiKE-M | 0.977      | 0.932      | 0.954    | 0.983    | 0.948   | 0.965      |
| PRASE-MultiKE-E | 0.979      | 0.930      | 0.954    | 0.988    | 0.955   | 0.972      |

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         |
| PRASE-BootEA-M  | 0.976      | 0.912      | 0.943    | 0.983    | 0.934   | 0.958      |
| PRASE-BootEA-E  | 0.982      | 0.912      | 0.945    | 0.987    | 0.933   | 0.960      |
| PRASE-MultiKE-M | 0.977      | 0.932      | 0.954    | 0.983    | 0.948   | 0.965      |
| PRASE-MultiKE-E | 0.979      | 0.930      | 0.954    | 0.988    | 0.955   | 0.972      |

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 alignment, 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

---

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 \( \alpha_1 = \alpha_2 = 1, \beta = 0.8, \delta_1 = \delta_2 = \delta_3 = 0.1 \), and choose cosine similarity as \( \alpha(\cdot) \). 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 a workstation with an Intel Xeon E5 CPU and an NVIDIA Tesla M40 GPU. The average time cost of our PARIS implementation on the four 100K datasets is 1697 seconds\(^4\), 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 with several iterations 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.

\(^4\)The average running time by the Java implementation is 89 s.
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, IPrTransE, GCNAAlign, and IMUSE. Table 4 shows their performance. It can be observed that all the PRASE models achieve a higher F1-score 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 \( U^P \). 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 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 use only 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 on six datasets have 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. For future work, 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 [Chen et al., 2020].
