RePS: Relation, Position and Structure aware Entity Alignment

Anil Surisetty*
Deepak Chaurasiya*
anil.surisetty@mastercard.com
deepak.chaurasiya@mastercard.com
AI Garage, Mastercard
Gurugram, Harayana, India

Nitish Kumar
Alok Singh
nitish.srivasatava@mastercard.com
alok.singh2@mastercard.com
AI Garage, Mastercard
Gurugram, Harayana, India

Gaurav Dhama
Aakarash Malhotra
Ankur Arora
gaurav.dhama@mastercard.com
aakarsh.malhotra@mastercard.com
ankur.arora@mastercard.com
AI Garage, Mastercard
Gurugram, Harayana, India

Vikrant Dey
ohm.vikrant2000@mastercard.com
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand, India

ABSTRACT

Entity Alignment (EA) is the task of recognizing the same entity present in different knowledge bases. Recently, embedding-based EA techniques have established dominance where alignment is done based on closeness in latent space. Graph Neural Networks (GNN) gained popularity as the embedding module due to its ability to learn entities' representation based on their local sub-graph structures. Although GNN shows promising results, limited works have aimed to capture relations while considering their global importance and entities' relative position during EA. This paper presents Relation, Position and Structure aware Entity Alignment (RePS), a multi-faceted representation learning-based EA method that encodes local, global, and relation information for aligning entities. To capture relations and neighborhood structure, we propose a relation-based aggregation technique – Graph Relation Network (GRN) that incorporates relation importance during aggregation. To capture the position of an entity, we propose Relation-aware Position Aggregator (RPA) to capitalize entities' position in a non-Euclidean space using training labels as anchors, which provides a global view of entities. Finally, we introduce a Knowledge Aware Negative Sampling (KANS) that generates harder to distinguish negative samples for the model to learn optimal representations. We perform exhaustive experimentation on four cross-lingual datasets and report an ablation study to demonstrate the effectiveness of GRN, KANS, and position encodings.

*Both authors contributed equally to this research.

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524720

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; • **Information systems** → *Data mining*.

KEYWORDS

Entity Alignment, Representation Learning, Knowledge Graph

ACM Reference Format:
Anil Surisetty, Deepak Chaurasiya, Nitish Kumar, Alok Singh, Gaurav Dhama, Aakarash Malhotra, Ankur Arora, and Vikrant Dey. 2022. RePS: Relation, Position and Structure aware Entity Alignment. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3487553.3524720

1 INTRODUCTION

The use of Knowledge Graphs ($\mathcal{KG}s$) to represent structured data has emerged as one of the primary tools in domains such as fraud detection [24] and dialogue generation [16]. Different $\mathcal{KG}s$ depict different information about an entity. Hence, it is only natural to fuse such complementary $\mathcal{KG}s$ and use the complete information of an entity. Due to the asynchronous nature of $\mathcal{KG}s$, Entity Alignment (EA) techniques have gained significant attention, enabling the fusion of such $\mathcal{KG}s$ by identifying identical entities in different $\mathcal{KG}s$. EA is the task of establishing an equivalence association between two nodes belonging to complementary $\mathcal{KG}s$. Such knowledge fusion can be used in applications such as intelligent cross-lingual question-answering [8, 12] and fact-checking [10]. Distinct schemata, different granularities of information, and the absence of linguistically similar aliases have inhibited the use of traditional EA methods, which rely on attribute matching [9, 18] or crowdsourcing [31]. Advancements in graph embedding generation methods and their ability to efficiently encode an entity's heterogeneous information has given rise to embedding-based EA methods.

Though embedding-based EA methods provide a significant increase in performance over their predecessors, there are certain

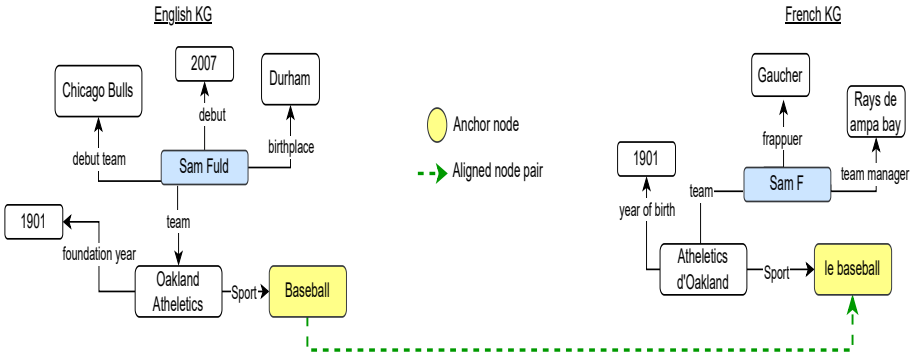

Figure 1: English and French $\mathcal{KG}$ sub-graph for entity Sam

aspects of information present in $\mathcal{KG}s$ which make this task challenging. (i) *Non-isomorphism*: Complementary $\mathcal{KG}s$ often have non-identical local structure for identical entities due to incomplete information present in different sources, thus leading to non-isomorphic subgraphs. For instance, in Fig. 1 Sam Fuld in English $\mathcal{KG}$ and Sam F in French $\mathcal{KG}$ represent the same real-world entity but have different degrees and non-identical neighborhoods. Since most of the embedding generation techniques primarily encode the local neighborhood structure, such representations can lead to sub-optimal alignment. (ii) *Relation-importance-inference*: Not all relations play an equal role in reducing search space and aligning entities. For instance, consider relation triples in Fig. 1. It is easier to identify and align Sam Fuld using $team \rightarrow Oakland\ Atheletics$ instead of $birthplace \rightarrow Durham$ as there might be many entities having the triple $birthplace \rightarrow Durham$ associated with them. Most of the existing embedding generation techniques utilize vector or matrix representations for representing a relation. Even though high-dimensional representation gives more representative power, it makes the final alignment result uninterpretable, i.e., it is difficult to reason why two entities got aligned. Since the actual alignment for entities is unknown, validation of results obtained is crucial in this task. Hence, the alignment architecture needs to be interpretable when attempting to align entities in an industrial setting.

With these challenges in focus, we propose RePS, a novel framework that encodes multi-faceted information- Relation, Position, and Structure of an entity to learn optimal representation for EA. To tackle the challenge of *Non-isomorphism*, this work proposes a Relation-aware Position Aggregator (RPA). RPA generates a position based representation of entities with respect to global anchor nodes. We hypothesize that position of identical nodes in the two $\mathcal{KG}s$ relative to a pair of pre-aligned entities will be similar and invariant to non-isomorphic subgraph structures present in the $\mathcal{KG}s$.

Furthermore, to address the problem of *Relation-importance-inference*, we propose Graph Relation Network (GRN), which generates relation sensitive entity representations capturing structure. GRN learns a scalar for each relation rather than a vector or a matrix, making its global importance more interpretable for the task of EA. GRN is built on the hypothesis that not all relations play equal importance while aligning two entities, as illustrated previously with the example of *Sam Fuld* in Fig. 1.

Lastly, to effectively train RPA and GRN, this work proposes a novel negative sampling method - Knowledge Aware Negative Sampling (KANS), that generates *harder to distinguish* negative examples utilizing the structure and relations in the $\mathcal{KG}s$. The proposed study manifests a four-fold contribution, as summarized below:

- Propose a robust EA methodology that considers the position of an entity, relational importance, and local subgraph information while aligning entities.
- Design an interpretable relation-aware encoder to capture relation's relative information while aligning an entity.
- Propose a novel negative sampling technique that generates harder negative examples utilized to train the model.
- Empirically demonstrate RePS superiority over other SOTA methods by reporting their performance on four cross-lingual open-source datasets and a comprehensive ablation study showing the importance of each component.

2 PRELIMINARIES

2.1 Problem Definition

We define the i^{th} $\mathcal{KG}$ as $\mathcal{KG}_i$=($\mathcal{E}_i$,$\mathcal{R}_i$,$\mathcal{T}_i$) where $\mathcal{E}_i$, $\mathcal{R}_i$, and $\mathcal{T}_i$ denote the set of entities, relations, and relation triples, respectively. EA can be achieved using supervised or unsupervised manner. This study aims to address EA in supervised manner. EA is the problem of aligning identical entities $e_1^1 \in \mathcal{E}_1$ and $e_1^2 \in \mathcal{E}_2$ in $\mathcal{KG}_1$ and $\mathcal{KG}_2$, respectively, given a set of pre-aligned entity pairs $\mathcal{L}$ referred as training labels or seeds.

2.2 Related Work

Over the years, Entity Alignment (EA) has evolved as an application of graph-based methods. This section introduces literature around embedding-based EA methods. Such methods align entities using learned representations based on their distance in the embedding space. Generally, the EA methods are categorised into Trans-based [1] or GNN based class. We discuss methods from both these classes and categorize them into supervised and semi-supervised methods.

2.2.1 Supervised. Supervised approaches utilize only $\mathcal{L}$ and do not iteratively add alignments while training which include methods such as MTransE [3], JAPE [20], RGCN [17], GCN-Align [25], RDGCN [26], HGCN [27], RAGA [30], EASY [5], Alinet [22], RREA [15], Dual-AMN [13]. Techniques such as MTransE and JAPE use translation constraint in their embedding modules, whereas the latter additionally utilize attribute triples to learn attribute preserving embeddings jointly. Methods such as RGCN [17], GCN-Align [25] utilize GCN [11] variants to aggregate, where the former learns a matrix for every relation and the latter utilize attribute information as initial embeddings. Methods such as RDGCN [26], HGCN [27], RAGA [30], EASY [5] utilize BERT or Glove generated embeddings along with attention-based aggregation techniques for the task of EA. Techniques such as AliNet [22] attempt to resolve non-isomorphism by utilizing attention for distant neighbors as well. Whereas methods such as RREA [15] and Dual-AMN [13] utilize a Relation Reflection Aggregator with orthogonal transformations. DualAMN uses an additional proxy matching layer with RREA for capturing cross-graph attention and achieves the current SOTA performance.

2.2.2 Semi-supervised. Semi-Supervised techniques add alignments iteratively to the seed set, which includes algorithms such as IPTransE [29], BootEA [21]. Both the techniques utilize translation based constraints. IPTransE discusses two versions of iterative addition: (i) Hard alignment strategy: Entity pairs aligned above a confidence threshold are iteratively added. (ii) Soft alignment strategy: Entity pairs aligned above a certain threshold are added with an $\mathcal{L}_1$ penalization based on relational paths to inhibit error propagation. Improving on inhibiting the error propagation, BootEA adds new aligned pairs by maximizing a global objective function that follows maximum alignment likelihood and one-to-one mapping constraint.

All the methods mentioned above learn relation and structure enriched embeddings using more details of entities' neighborhoods or attributes. However, they miss out on global node features such as its position in the graph and are less interpretable on relation importance in aligning entities. You et al. [28] proposed PGNN that learns node embeddings by encoding node's position in the graph. Although PGNN finds its application in graphs, it has limited use in the case of $\mathcal{KG}s$ as it does not consider any relations while learning nodes' positional representation. To the best of our knowledge, this is the first application of position encodings in the task of EA. In this work, we propose a novel method to use the relation weighted position of a node in a $\mathcal{KG}$ for EA.

3 PROPOSED APPROACH

This section describes the working of the three components of RePS, namely, Relation-aware Position Aggregator (RPA), Graph Relational Network (GRN), and Knowledge Aware Negative Sampling technique (KANS). RePS encodes positional information of entities using RPA, while relational and structural information is captured using GRN. This multi-faceted information is then dynamically aggregated to generate optimal node representation for EA. Fig. 3 illustrates the complete RePS architecture along with the functioning of each component.

3.1 Relation-aware Position Aggregator (RPA)

This paper introduces the notion of "position-awar" representation for EA tasks. We now describe the proposed relation enriched position-aware representation, which generalises a node's representation by focusing on two key concepts - (1) Position-aware representation and (2) Relation enriched position formulation. Fig. 2 illustrates the functioning of RPA layer.

3.1.1 Position-aware representation. A node's "structure-aware" representation is characterised by its neighboring nodes information. In contrast, position-aware representation captures a node's position in the network relative to a randomly sampled subset of nodes (hereby referred to as anchor sets).

To define a position-aware representation of a node, we first sample k anchor sets. An i^{th} anchor set is a collection of l_i number of nodes called anchors. Since we require the two identical entities in the two $\mathcal{KG}s$ to have similar positions, it is essential to define their position relative to aligned anchor sets. Hence, we first generate k anchor sets for $\mathcal{KG}_1$ by randomly sampling from the seed entities set of $\mathcal{KG}_1$. Then, to generate the corresponding anchor set for $\mathcal{KG}_2$, entities in each of k anchor sets of $\mathcal{KG}_1$ are replaced by their counterparts from $\mathcal{KG}_2$. This ensures that the positions of two nodes in different $\mathcal{KG}s$ are comparable. This sampling methodology is detailed in Appendix A.1.

During training, position-aware feature messages for node v are generated w.r.t. every anchor set as shown in Eq. 1. Further, every such message for node v is aggregated as shown in Eq. 2 to update the position embedding of node v.

$$M(v, \psi_i) = \phi\left(\{d(v, \psi_{i,j}) \cdot h^{l-1}_{\psi_{i,j}}\}^{l_i}_{j=1}\right) \tag{1}$$

$$h^l_{v_p} = g\left(\frac{1}{k+1}\left(\sum_{i=1}^{k} M(v, \psi_i) + h^{l-1}_v\right)\right) \tag{2}$$

where h^l_x represents the embedding of node x from layer l and $d(*,*)$ is the distance function between two nodes. $\psi_{i,j}$ is the j^{th} node in i^{th} anchor set. ϕ denotes a composition function. Empirically, Min function performs the best. Moving forward, we assume ϕ to be Min function unless stated otherwise. For graphs with no relation information, [28] also suggested to use shortest path length function as $d(*,*)$. Lastly, $g(X) = \sigma(W_1 * X + b_1)$ with W_1 and b_1 as trainable parameters and σ as an activation function.

3.1.2 Relation enriched position formulation. Every relation in a $\mathcal{KG}$ has a different discriminative strength, i.e., each relation's ability to uniquely describe an entity is different. For instance, as shown in Fig 1, Sam in English $\mathcal{KG}$ has multiple unique relations connected to it like "birthplace", "team", etc. Comparing these two relations, "$birthplace \rightarrow Durham$" narrows down the search space for a head entity to the population of Durham, i.e., $\approx 50,000$. Whereas "$team \rightarrow OaklandAthletics$" narrows down the same to the number of players at Oakland Athletics, i.e., $\approx$ 40. This difference in discriminative power of relations raises the need for a relation weighted.

It is essential that the weight assigned to a relation present in both $\mathcal{KG}_1$ and $\mathcal{KG}_2$ should be similar. This ensures that similar entities in two graphs have similar aggregation and, consequently, similar representations. Hence, we use the Relative frequency (Rf)

of a relation in a $\mathcal{KG}$ as its weight since it is invariant to different $\mathcal{KG}$ sampling techniques. Rf of a relation is defined in Eq. 3. Further, the relation enriched distance function between two nodes is defined in Eq. 4

$$Rf(r, \mathcal{KG}_i) = \frac{|triples\ in\ \mathcal{KG}_i|}{|r\ relation\ triples\ in\ \mathcal{KG}_i|} \tag{3}$$

$$pd(u, v) = \min_{q} \left(\sum_{r \in P_q(u,v)} Rf(r, \mathcal{KG}_i) \right) \tag{4}$$

where $|*|$ denotes cardinality of set $*$, $u, v \in \mathcal{KG}_i$ and $P_q(u, v)$ is the list of relations in the q^{th} path between u and v.

This distance function used in the aggregation step mentioned in Eq. 6 generates relation enriched position-aware representation of entities. We hereby refer to this layer, using relation enriched distance function, as Relation-aware Position Aggregator (RPA). The message function and aggregation step of the RPA layer can be written as:

$$M'(v, \psi_i) = \phi \left(\{pd(v, \psi_{i,j}) \cdot h^{l-1}_{\psi_{i,j}}\}^{l_i}_{j=1} \right) \tag{5}$$

$$h^l_{v_p} = g \left(\frac{1}{k+1} \left(\sum_{i=1}^{k} M'(v, \psi_i) + h^{l-1}_v \right) \right) \tag{6}$$

3.2 Graph Relation Network (GRN)

Vanilla GNNs are incapable of capturing the heterogeneous nature of relations present in $\mathcal{KG}s$. This has led to the proposal of various relation-aware methods which embed the information of different neighboring relations of an entity. Recent methods such as [13] and [14] represent a relation as a diagonal matrix and use them during aggregation. Such methods do not fare well in scenarios where interpretability of relation representation is paramount. Representing a relation as a scalar enables comparison and helps build an interpretable EA architecture. For instance, in an industrial setting where the correct alignment of entities is unknown, it is helpful to know which set of relations played a significant role while aligning two entities. Such information can be used to validate the alignment results of a model. Therefore, we propose a relation–aware representation learning method which learns the relative information of a relation as a scalar and uses it during aggregation to generate a local representation of an entity.

The layer capturing this relation-aware information will hereby be referred to as Graph Relation Network (GRN). GRN layer takes three metrics as inputs, h_v representing the initial node embedding for node v, c_v representing learnable coefficient for node v and $c_{r_{v,u}}$ representing the learnable coefficient for relation r connecting node v and u. During aggregation, the representation of nodes neighboring node v is weighed using the connecting relation's importance and summed as shown in Eq. 7. Further, The aggregation step of GRN at layer l is shown in Eq 8.

$$AGG(v) = \sum_{i \in N_v} \frac{h^{l-1}_i}{1 + c_{r_{v,i}}} \tag{7}$$

$$h^l_{v_r} = f \left((1 + c_v) \cdot h^{l-1}_v + AGG(v) \right) \tag{8}$$

where $f(X) = W_2 * X + b_2$ with W_2 and b_2 as learnable parameters and N_v denotes the neighborhood of node v.

The representation $\mathbf{h}^l_{v_p}$ (Eq. 6) holds information regarding position of an entity and $\mathbf{h}^l_{v_r}$ (Eq. 8) holds neighborhood information, both of which contains relation information. We chose not to concatenate them directly and utilize them for EA as these representations may have unequal importance in aligning entities. To learn such entity specific importance of these representations, inspired from [7] we obtain representation of v as

$$h^l_v = \mathbf{g}(h^l_{v_p}) \cdot h^l_{v_p} + (1 - \mathbf{g}(h^l_{v_p})) \cdot h_{v_r} \tag{9}$$

where $\mathbf{g}(h^l_{v_p}) = \sigma(W_3 * h^l_{v_p} + b_3)$ learns the aforementioned relative importance with W_3 and b_3 as trainable parameters and σ as the activation function.

3.3 Knowledge Aware Negative Sampling (KANS)

Most EA based methods are optimized using a margin-based loss function that aims to increase the plausibility of positive node pairs, i.e., pre-aligned entity pairs (seeds), and decrease the same of negative node pairs i.e., nodes which should not be aligned. Existing EA methods use either a uniform or ε-truncated sampling [21] to generate negative node pairs while training. Uniform sampling involves randomly replacing either the head or tail by other entities from the $\mathcal{KG}$ to generate the negative node pair set. To generate "harder to distinguish" positive and negative samples, BootEA proposed ε-truncated negative sampling, which samples negative samples from an entity's closest neighbors in the latent space. This approach requires the detection of an entity's closest neighbors as the possible set of negative node pairs every epoch, resulting in a significant overhead of time complexity.

Our work introduces a novel Knowledge Aware Negative Sampling technique (KANS) that generates harder negative samples by utilizing the structural and relational knowledge present in the $\mathcal{KG}$. Instances of such hard negative samples generated by KANS are listed in Table 6. For every tuple (v, v') present in seeds $\mathcal{L}$, the negative samples for v are sampled from set Φ_v, where Φ_v is a set of entities which share at least one relation-tail or relation-head with v'. KANS omits the redundancy of detecting possible negative node pairs every epoch, reducing the overall training time complexity.

We train our model by minimizing the following contrastive loss and aligning entities using CSLS [4] alignment strategy.

$$L = \sum_{(p,p') \in \mathcal{L}} \|\mathbf{p} - \mathbf{p}'\| + \beta \sum_{(p,q) \in \mathcal{L}'} [\,\gamma - \|\mathbf{p} - \mathbf{q}\|\,]_+ \tag{10}$$

where β is a weighing parameter and γ is the margin. $\mathcal{L}' = \{(p, q)\} \cup \{(p, s)\}$ where $p \in \mathcal{L}$, q is sampled from Φ_p and s is randomly sampled from other $\mathcal{KG}$. Number of negative samples from Φ_p is increased incrementally during training.

4 EXPERIMENT DESIGN

We demonstrate our results and analysis using two sparse cross-lingual datasets English-French (EN_FR_V1) and English-German (EN_DE_V1), and two dense cross-lingual datasets English-French (EN_FR_V2) and English-German (EN_DE_V2). These datasets are generated using Iterative Degree Sampling as detailed in [23], on the

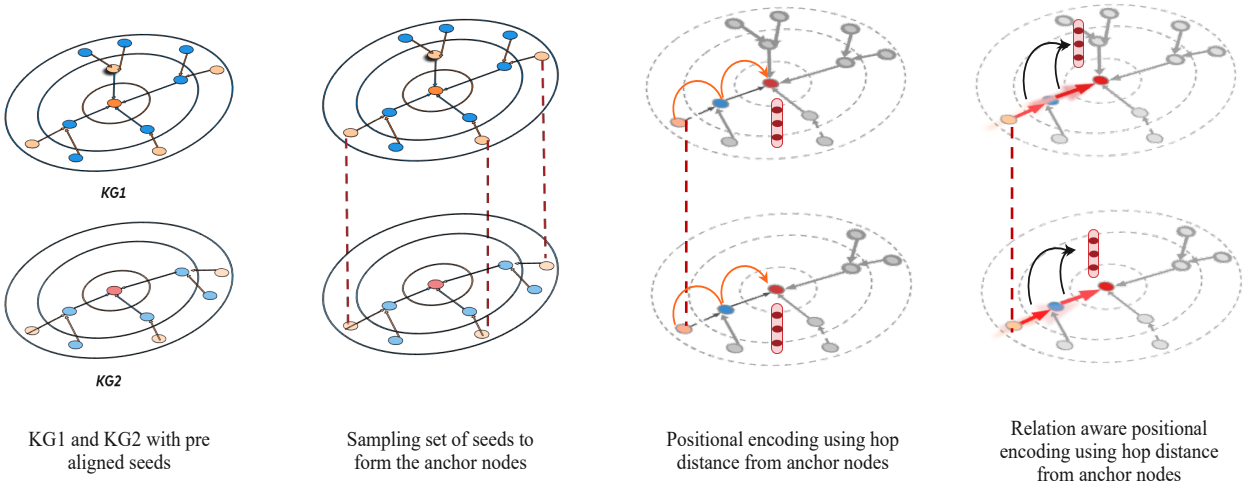

Figure 2: Method showing the different information encoded to learn the representation of an entity for EA.

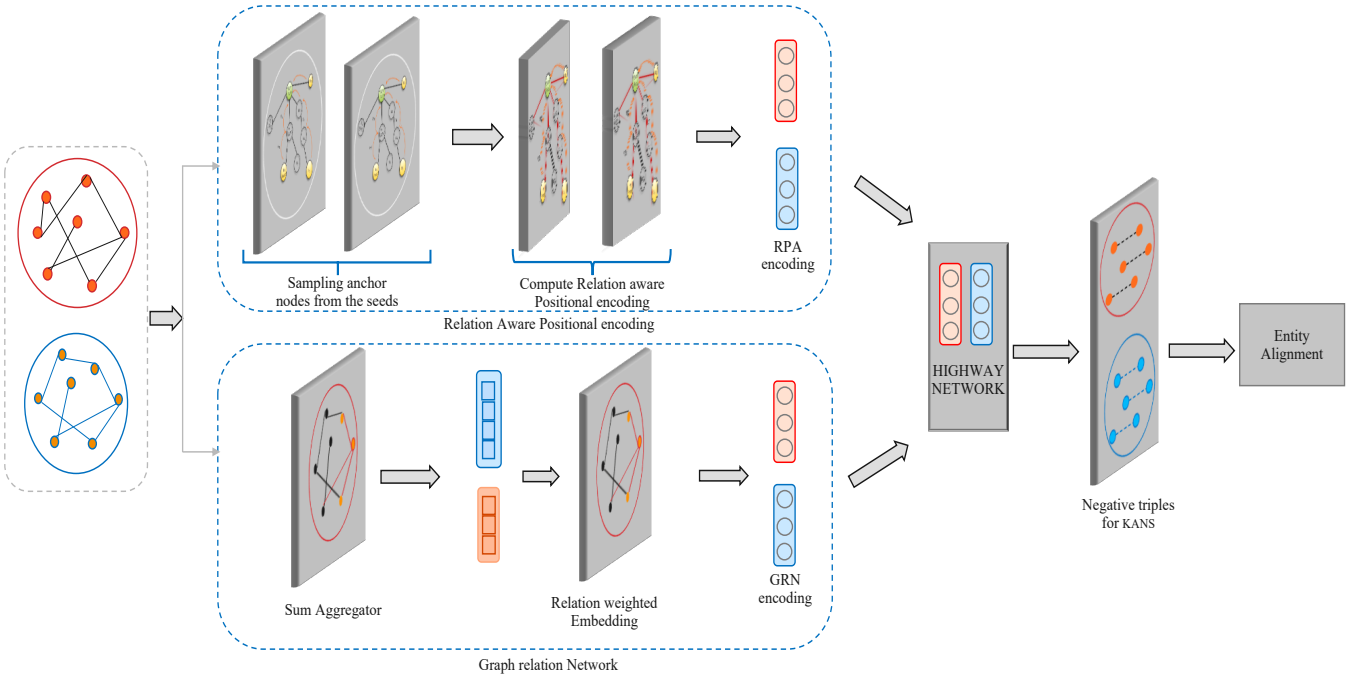

Figure 3: Architecture of RePS showing the different information encoded to learn the representation of an entity for EA

DBpedia knowledge base. Dataset statistics are shown in Appendix A.2.

All the methods use 20% seeds for training, 10% for validation, and 70% for testing. For a fair comparison, we used a batch size of 5000 on relation triples, maximum training epochs as 1000, and early stopping on validation set's Hits@1 with the patience of 15 epoch for all the baseline methods. For other method-specific hyperparameters such as the number of layers and margins in the alignment loss, values proposed by the respective authors are followed. Following Bourgain theorem [2], RPA layer samples 55 anchor sets ($O(\log^2 3000)$) for each $\mathcal{KG}$ and refreshes the same after every 5 epochs. For RePS, hyperparameter tuning is done over parameters values $\gamma = \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, epoch to start injecting hard negative samples $\{25, 50, 75, 100\}$, layer dimensions $\{100, 200, 300, 400, 500\}$ for initialisation of GRN. The source code of RePS and datasets used in the paper are available at https://github.com/Deepak-Work/RePS.

Table 1: Comparative results of RePS on cross-lingual (English-French and English-German) sparse (V1) and dense datasets (V2)

Dataset	EN_FR_V1			EN_DE_V1			EN_FR_V2			EN_DE_V2		
Method	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
MTransE [3]	24.3	45.5	0.34	27.7	49.0	0.38	23.5	42.8	0.33	19.64	35.41	0.28
IPTransE [29]	17.4	32.9	0.25	30.5	49.6	0.40	19.9	40.9	0.30	46.29	67.11	0.56
JAPE [20]	26.4	49.3	0.37	29.8	53.3	0.41	30.3	52.7	0.41	15.95	31.42	0.24
RSN4EA [6]	39.2	59.1	0.48	59.2	75.8	0.6	57.1	74.6	0.65	80.21	89.61	0.84
GCNAlign [25]	34.4	60.4	0.46	55.2	73.8	0.64	41.9	70.7	0.55	54.12	72.52	0.63
AliNet [22]	36.9	59.3	0.47	60.1	75.8	0.67	54.5	80.6	0.65	79.24	90.09	0.84
HyperKA [19]	43.5	66.7	0.54	59.8	80.4	0.71	**62.3**	85.4	**0.76**	83.64	91.21	0.86
RePS	**45.5**	**73.1**	**0.55**	**68.2**	**86.4**	**0.74**	62.2	**90.5**	0.72	**84.18**	**92.34**	**0.87**

5 RESULTS AND ANALYSIS

We compare RePS with seven existing methods on all the four datasets as shown in Table 1. RePS achieves a gain of 8.6% on Hits@1 compared with AliNet, and 6.3% against RSN4EA. For a fair comparison, we exclude methods that use name-based linguistic information or entity attributes from our baselines since RePS, in its current form, does not utilize that information. We also limit our comparisons to supervised EA methods since iterative addition introduces additional time complexity (sampling anchor sets) and error propagation whose handling is left as future work.

As shown in Table 1, GNN based EA methods consistently outperform Trans based methods. This can be attributed to the more expressive architecture of such GNN based methods that capture an entity's rich structural information. GNN based methods like AliNet improved EA accuracy by capturing the local neighborhood information of an entity and its few global properties. RePS aims to further improve the performance on EA task by capturing an entities' relation-aware local information and its global position based topological information. It can be seen that RePS outperforms other benchmarking EA methods in terms of MRR and Hits% metrics.

5.1 Analysis

5.1.1 Comparison of RePS with less interpretable models: GRN block of RePS learns the relative importance of a relation as a scalar and captures a relation-aware representation of an entity's local neighborhood. When aligning entities in real-world KGs, the correct alignment for all the entities is often unknown. Hence, there is a need for a more interpretable model that can validate the plausibility of aligned entities and GRN attempts to do the same. For instance, Table 4 lists out the three most important relations captured by GRN while aligning the entity "Nevin Ashley". Now, to validate if "Nevin Ashley" has been correctly aligned with node x, it is recommended to check for the "Team", "Manager" and "Position" relation of node x and compare them with those of "Nevin Ashley". Other relation-aware aggregators do not provide such a flexibility and are often unsuitable for industrial EA tasks.

Table 2 compares the performance of GRN with other relation-aware aggregators. We note that, even with a significantly less number of parameters, GRN achieves competitive performance. In scenarios where the interpretability of a model is not essential, it is advised to use a more expressive relation-aware aggregator in the place of GRN. Hence, we compare the performance of the current SOTA method, DualAMN (RREA+Proxy Matching Layer), with that of RePS with GRN replaced with RREA aggregator (uninterpretable setting). Figure 4 shows this comparison in terms of Hits@1 w.r.t. number of training epochs. Thus, even in the uninterpretable setting, RePS achieves SOTA performance for the task of EA.

Table 2: Results of relation-aware aggregators on EN_FR_V1

Agg.	H@1	H@5	Parameters
RGCN [17]	12.9	34.7	$O(n_r \times d \times d)$
GRN (Ours)	26.7	48.7	$O(n_r)$
MRAEA [14]	27.7	48.3	$O(n_r \times d)$
RREA [15]	33.4	55.1	$O(n_r \times d)$

Table 3: Ablation study of RePS

Method	Hits@1	Hits@5	MRR
RePS w/o GRN & RPA	34.02	52.68	0.42
RePS w/o GRN	41.23	71.17	0.52
RePS	45.50	73.08	0.55

5.1.2 Ablation Study. We show the effectiveness of each component of RePS by conducting experiments on the EN_FR_V1 dataset. Empirical results of the ablation are listed in Table 3.

RePS w/o GRN & RPA Without GRN and RPA, RePS performs equivalent to multiple layers of GCNs that capture only an entity's local structural properties. We observe 11.48% decrease in accuracy, which empirically shows the deficiencies in such information for aligning entities.

RePS w/o GRN - Supplementing the node's rich local neighborhood information with its global position for EA shows a significant improvement in terms of accuracy, supporting our hypothesis that RPA better aligns entities by utilizing their similarity in positions relative to anchor sets.

RePS - Not all relations are equally important while aligning two entities. To establish the same, we use the RePS architecture as detailed in Fig. 3 which uses a relation importance-based aggregator GRN rather than GCN. We observe a 4.27% improvement on Hits@1, supporting our argument for learning the relative importance of relations to align entities.

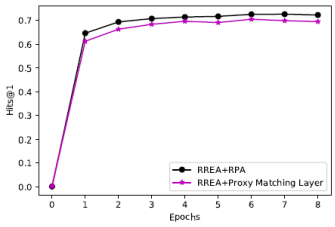

Figure 4: RPA efficacy over Proxy layer of DualAMN

Table 4: Weights learnt for associated relations of Nevin Ashley in the French $\mathcal{KG}$. #Head Entites denotes the count of distinct number of entities connected to "Nevin Ashley → Relation"

#Head Entities	Coeff	Relation	Tail Entity
6	0.28	Team	*Mets de New York*
27	0.25	Manager	*Brewers de Milwaukee*
42	0.15	Position	*Receveur (baseball)*

6 CONCLUSION

Existing EA techniques capture the local structure and attributes of the entities very well. However, limited research is present detailing the use of relation information along with entities' network-related global properties for EA. This paper proposes a novel framework that encodes local structural information and the global importance of relation during aggregation along with relation-sensitive position information. We focus on encoding this multi-faceted information to generate an optimal representation for aligning entities. The improved alignment performance provides empirical validation for the use of position during EA. We further propose a novel negative sampling technique that generates harder to distinguish negative examples for an entity to train our model. In future, this work can be enhanced by representing the shortest path in RPA as a relation vector composition or by incorporating attributes during EA.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems* 26.

[2] Jean Bourgain. 1985. On Lipschitz embedding of finite metric spaces in Hilbert space. In *Israel Journal of Mathematics*, Vol. 52. 46–52.

[3] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multi-lingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. In *International Joint Conference on Artificial Intelligence*. 1511–1517.

[4] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word Translation Without Parallel Data. In *International Conference on Learning Representations*.

[5] Congcong Ge, Xiaoze Liu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2021. Make It Easy: An Effective End-to-End Entity Alignment Framework. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. 777–786.

[6] Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs. In *International Conference on Machine Learning*. 2505–2514.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[8] Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning Knowledge Graphs for Question Answering through Conversational Dialog. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 851–861.

[9] Wei Hu, Jianfeng Chen, and Yuzhong Qu. 2011. A Self-training Approach for Resolving Object Coreference on the Semantic Web. In *International conference on World Wide Web*. 87–96.

[10] Viet-Phi Huynh and Paolo Papotti. 2019. Buckle: Evaluating Fact Checking Algorithms Built on Knowledge Bases. In *International Conference on Very Large Data Bases*, Vol. 12. 1798–1801.

[11] Thomas N Kipf and Max Welling. 2016. Semi-supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[12] Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019. XQA: A Cross-lingual Open-domain Question Answering Dataset. In *Annual Meeting of the Association for Computational Linguistics*. 2358–2368.

[13] Xin Mao, Wenting Wang, and Yuanbin Wu. 2021. Boosting the Speed of Entity Alignment 10*: Dual Attention Matching Network with Normalized Hard Sample Mining. In *International conference on World Wide Web*. 821–832.

[14] 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 *ACM International Conference on Web Search and Data Mining*. 420–428.

[15] Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational Reflection Entity Alignment. In *ACM International Conference on Information and Knowledge Management*. 1095–1104.

[16] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Association for Computational Linguistics*. 845–854.

[17] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *European Semantic Web Conference*. 593–607.

[18] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. In *International Conference on Very Large Data Bases*.

[19] Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020. Knowledge Association with Hyperbolic Knowledge Graph Embeddings. In *Empirical Methods in Natural Language Processing*. 5704–5716.

[20] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*. 628–644.

[21] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding.. In *International Joint Conference on Artificial Intelligence*, Vol. 18. 4396–4402.

[22] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, and Wei Zhang. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Association for the Advancement of Artificial Intelligence*, Vol. 34. 222–229.

[23] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, and Farahnaz Akrami. 2020. A benchmarking study of embedding-based entity alignment for knowledge graphs. In *International Conference on Very Large Data Bases*, Vol. 13.

[24] Weikang Wang, Jiajun Zhang, Qian Li, Chengqing Zong, and Zhifei Li. 2019. Are You for Real? Detecting Identity Fraud via Dialogue Interactions. In *Empirical Methods in Natural Language Processing*. 1762–1771.

[25] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Empirical Methods in Natural Language Processing*. 349–357.

[26] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware Entity Alignment for Heterogeneous Knowledge Graphs. In *International Joint Conference on Artificial Intelligence*.

[27] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. Jointly Learning Entity and Relation Representations for Entity Alignment. In *Empirical Methods in Natural Language Processing*. 240–249.

[28] Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware Graph Neural Networks. In *International Conference on Machine Learning*. 7134–7143.

[29] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings. In *International Joint Conference on Artificial Intelligence*. 4258–4264.

[30] Renbo Zhu, Meng Ma, and Ping Wang. 2021. RAGA: Relation-aware Graph Attention Networks for Global Entity Alignment. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 501–513.

[31] Yan Zhuang, Guoliang Li, Zhuojian Zhong, and Jianhua Feng. 2017. Hike: A Hybrid Human-Machine Method for Entity Alignment in Large-Scale Knowledge Bases. In *ACM International Conference on Information and Knowledge Management*. 1917–1926.

A RESEARCH METHODS

Algorithm 1: Sampling Anchor Sets

Input: Seeds: $\mathbb{L}$, **Number of seeds:** n
Let $\phi_1 = \phi_2 = \emptyset$,$m = [log^2 n], i = 0$.
while $i < m$ **do**
  $j = \frac{m}{2^{(i+1)}}$ $\phi = randomsample(\mathbb{L}, size = j)$
  $\phi_1.add(\mathcal{KG}_1\ entities\ from\ \phi)$
  $\phi_2.add(\mathcal{KG}_2\ entities\ from\ \phi)$ $i++$;
end
Output: ϕ_1, ϕ_2

A.1 Sampling methodology of Anchor sets

Linial et al. [2] proposed Bourgain theorem which provides insights in the selection of the values k & l_i. First, only O(log2n) anchor sets are required to guarantee an insignificant distortion in position-aware embeddings. Second, the anchor sets need to be of different sizes. Here, we define the size of anchor sets to be exponentially distributed i.e., $l_i/l_{i-1} = \alpha$ where $\alpha \in (1, \infty)$.

We further describe the intuition behind the latter. For the sake of discussion, let us assume that the distance between an anchor set and a node v is defined as the minimum shortest path length between node v and all the anchor set S nodes. If all the anchor sets have only one anchor node, the distance between v and S would be very certain, i.e., one would determine precisely where a node is located w.r.t. an anchor set. However, the distance between node v and an anchor set will always be higher due to the fundamental nature of graphs (the number of nodes at a distance d from a node increases exponentially with an increase in d). This results in a skewed distribution of representation. On the other hand, if all the anchor sets have a large number of nodes, it results in a less certain distance calculation, but the distribution of representation is more uniform. Hence, anchor sets of exponentially increasing cardinality are sampled as a trade-off between the two extremes. Psuedo code for the sampling methodology is shown in Algorithm 1.

A.2 Dataset Statistics

Statistical properties of the dataset used in this study is listed in Table 5.

Table 5: Dataset description

Dataset	$\mathcal{KG}s$	#Ent.	#Rel.	#Rel. triples
EN_FR_V1	EN	15K	267	47,334
	FR	15K	210	40,864
EN_DE_V1	EN	15K	215	47,676
	DE	15K	131	50,419
EN_FR_V2	EN	15K	193	96,318
	FR	15K	166	80,112
EN_DE_V2	EN	15K	169	84,867
	DE	15K	96	92,632

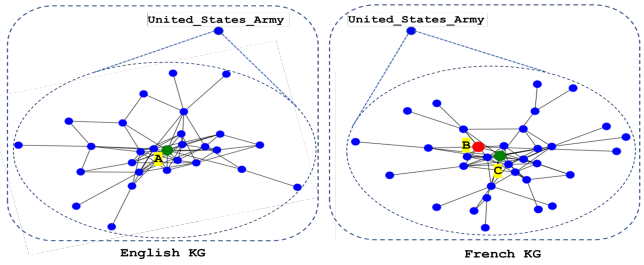

Figure 5: Increased distinction due to position. Node A is an entity in the English $\mathcal{KG}$, Node B and C are the entities matched by AliNet and RePS, respectively.

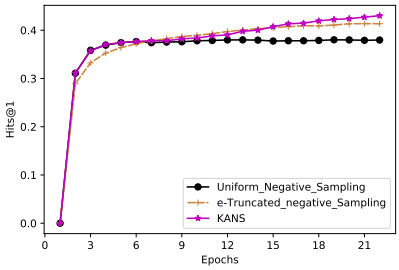

Figure 6: Comparison of KANS, ϵ-truncated, and Uniform negative sampling on EN_FR_V1

.

B EFFICIENCY OF METHODS

B.1 Efficacy of position embedding generated by RPA:

RPA layer aims to capture distinguishable information for entities lying in very close proximity in a $\mathcal{KG}$. Structure and relation-aware aggregators struggle to differentiate between such entities as these entities have a highly overlapping neighborhood. The position of two entities lying in close proximity is not the same as their relation-weighted shortest path from an anchor node can be very dissimilar. Hence, we utilize the notion of position and hypothesize that an entity appearing in two $\mathcal{KG}s$ will have a similar position with respect to an anchor node. Fig. 5 describes a scenario where Node B (*Navy_Distinguished_Service_Medal*) and C (*Navy_Cross*) in French $\mathcal{KG}$ appear at a 2-hop distance and have a very similar local neighborhood. Node A (*Navy_Cross*) in English $\mathcal{KG}$ is the correct alignment of Node C. Although the structure of nodes B and C are very similar, we note that their relation-weighted distance from *United_States_Army* (French anchor) is 27.79 and 42.5, respectively. Further, the distance of Node A relative to *United_States_Army* (English anchor) is 56.85, which is more similar to Node C's position than that of Node B. This difference in position of the two entities enables RePS to align such entities correctly.

The novel notion of position can be paired with any existing aggregator to learn richer representations. In scenarios where an aggregator's simplicity is not paramount, the RPA layer can be coupled with SOTA aggregators to achieve a better alignment accuracy.

Table 6: Negative samples generated using KANS v/s ε-truncated.

Snoop dog		Attack(Political Party)	
KANS	**ϵ -Truncated Negative**	**KANS**	**ϵ -Truncated Negative**
T-pain	David J.Lesar	Plaid Cymru	AC Omonia
Daz dillinger	Foreign Air Supply Company	Common Man's Front	Olympique Lyonnais
Ryan leslie	Thomas Hunt Morgan	Democratic Unionist Party	DC Virunga
Warren G	Moesha	Homeland Union	AL Sadd SC
Fall Out Boy	George Davis Snell	Democrats 66	Real Madrid C.F.

While comparing RePS with the current SOTA, DualAMN (RREA + Proxy matching layer), we note that the RPA layer encompasses the Proxy matching layer used by DualAMN. RPA layer mimics the latter when path lengths between all entities are considered unity and pre-aligned entities are initialized with the same representation. Further, Fig. 4 shows that substituting the Proxy matching layer of DualAMN with RPA achieves SOTA performance. Hence, we conclude that the RPA layer can be an additional block with any embedding module to obtain a richer node representation.

B.2 Effectiveness of KANS:

KANS aims to generate harder negative examples by perturbing the head or tail in $\mathcal{KG}$ triple utilizing relation information in the $\mathcal{KG}$. Our work shows that the negative samples obtained are very similar to the original entity, making it difficult for the network to distinguish. Table 6 shows entities and their negative samples. We note that the negative samples for entity *Snoop Dog*, a rapper, are also rappers. Similarly, negative samples of *Attack (Political party)* resemble other political parties. The effectiveness of KANS can be further observed from Fig. 6 where the RePS framework is trained on EN_FR_V1 with uniform negative sampling and ϵ-Truncated negative sampling.