Entity Alignment Across Knowledge Graphs Based on Representative Relations Selection

Conference Paper · November 2018

DOI: 10.1109/ICSAI.2018.8599288

Li Liu
Chongqing University of Posts and Telecommunications

Fujin Zhong
Chongqing University of Posts and Telecommunications

Entity Alignment Across Knowledge Graphs Based on Representative Relations Selection

Youmin Zhang
School of Software
Chongqing Institute of Engineering
Chongqing, China

Li Liu*, Shun Fu and Fujin Zhong
College of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China

***Abstract*—Entity alignment across knowledge graphs is an important task in web mining. The aligned entities can be used for transferring knowledge across knowledge graphs and benefit several tasks such as cross-lingual knowledge graph construction and knowledge reasoning. This paper propose a representation learning based algorithm for embedding knowledge graph and aligning entities. In particular, considering the multi-type relations in knowledge graph, we select the alignment-task driven representative relations based on the pre-aligned entity pairs. With the help of selected relations, we embed the entities across networks into a common space by modeling entities' head/tail are with corresponding context vectors. For entity alignment task, pre-aligned entities are adopted to facilitate the transfer of context information across the knowledges graphs. Through this way, the problem of entity embedding and alignment can be solved simultaneously under a unified framework.. Extensive experiments on two multi-lingual knowledge graphs demonstrate the effectiveness of the proposed model comparing with several state-of-the-art models.**

***Keywords*—Entity Alignment; Knowledge Graph Embedding; Representative Relation Selection.**

I. INTRODUCTION

Knowledge graph, a structural informative representation of human's knowledge, is an important resource in artificial intelligence era as it can benefit several applications such as intelligent search engine, question answering system, recommendation system, and etc. The construction of knowledge graph attracts more and more attention from both academia and industry nowadays.

Recently, several methods are conducted for building the knowledge graph including entity recognition and entity relation prediction. Some remarkable large-scale knowledge graphs such as DBpedia [1] and FreeBase [2] are obtained. These knowledge graphs are developed separately, which lead to inherent heterogeneous characteristic between them. On the contrary, they can also act as potential supplement resource to each other for integrating into a more complete knowledge. Suppose there are some pre-aligned entities (also called as anchors) across knowledge graphs, these entities can be act as "bridge" nodes not only benefit for cross-lingual knowledge construction but also for the knowledge reasoning across languages.

With the popularity of knowledge embedding, some embedding based models are proposed for the entity alignment. These methods can roughly be categorized into "parameter sharing" methods [3], [4] and "vector mapping" methods [5], [6]. The first class of methods tries to embed the entities into a common space by sharing the parameters which are provided by the anchor entities, while the second class of methods embed the entities in separated spaces for each knowledge graphs, then an anchor entities based vector mapping objective function is adopted for the alignment. Most of these embedding methods are based on translational methods [7], which can not capture the triangular structures with a single type of relation, This problem frequently appears in multi-relational networks frequently [8]. Focused on this problem, Li [9] proposes a non-translational model for aligning entities and relations of multi-relational knowledge graphs, but they fail in modeling the bi-directional relations and ignoring the importance of different types of relations. More head anchors sharing by an entity pair means they always act as objects in a triplets, and sharing more tail anchors means they always act as subjects. Intuitively, an entity pair sharing more similar sets of head and tail entities can be identified more easily than only considering one directional shared set. Another model which is similar with our work is IONE [10], they embed the social network with the followership/followeeship of each user explicitly modeled as input/output context vector representations. Different from social network, the knowledge have multi-types of relations which are ignored by IONE model. In this paper, we propose a representation learning based Entity Alignment model with Representative Relations (EARR), which not only considers the importance of relation types but also models the bi-directional relations, we summary the contributions of this paper as follows:

- For the multi-types of relations in knowledge graphs, we proposed an alignment-task based relations selection algorithm. In detail, we select the relations which are representative for alignment task using an information grain-like algorithms based on the anchor entity pairs.
- With the help of the representative relations set, we propose a probabilistic model which project all the knowledge graph entities into a common space. During the project process, we consider the bi-direction of relations, the head and tail entities are explicitly modeled with corresponding context vectors.

* Corresponding author: Li Liu. Email: liliu@cqupt.edu.cn

978-1-7281-0120-0/18/$31.00 ©2018 IEEE

- We evaluate the EARR (the proposed model) with detailed experiments based on real-world dataset comparing with several the state-of-the-art methods.

II. Related Work

A. Knowledge Graph Embedding

Knowledge Graph Embedding has demonstrated its effectiveness in modeling the semantic information of Knowledge Graph. Translation based methods [7], [11], [12], [13] were widely used for this task. For instance, given the triplet (h, r, t), TransE [7] introduces relations as translating operations between head and tail entities, then projects both entities and relations into vector space by minimizing the objective function $||h + r - t||$. There are various mapping properties of relations in Knowledge Graph, such as reflexive, one-to-many, many-to-one, and many-to-many. In order to deal with these problems, its extensions including TransH[11], TransR[12], and pTransE[13] are proposed. Different with these methods, Liu[8] proposes a structural representation learning approach for multi-relational networks to address the weakness of trans-family that render them incapable of preserving the triangular structures.

B. Knowledge Graph Alignment

Recently, several embedding-based alignment models were proposed. MTransE [3] adapt TransE to encoded entities and relations of each language into separate vector spaces, and then learns transformation between them by utilizing anchor entities across knowledge for alignment. ITransE/IPTransE [4] proposes to utilize TransE/pTransE to jointly embed different KGs, it encode both entities and relations into a same low-dimensional subspace, then aligned entities according to their distance and update embeddings iteratively with newly-aligned entities. JE [5] and JAPE [6] learns embeddings for entities and relations of different KGs in a unified embedding space. JAPE [6] also embeds attributes and leverages attribute correlations to refine entity embeddings. BootEA [14] propose a bootstrapping approach for entity alignment. It iteratively labels likely entity alignment as training data for learning alignment-oriented KG embeddings. Furthermore, it employs an alignment editing method to reduce error accumulation during iterations. NTAM [9] applies a probabilistic model to project multiple networks onto a common embedded space. Instead of applying the trans-family, the probabilistic model explores the structural properties for network representation learning. And a more precise and robust alignments are achieved by the NTAM model.

III. Model Framework

Let $KG^{(X)} = (E^{(X)}, R^{(X)})$ be a knowledge graph X where $E := \{h_i^{(X)}\} \cup \{t_i^{(X)}\}$ is the set of head entities $\{h_i^{(X)}\}$ and tail entities $\{t_i^{(X)}\}$, $R := \{r_i^{(X)}\}$ is the set of directed edges representing the different relation types between the entities. For multiple knowledge graphs X and Y, we use $E_{anchor} := \{e_a \in E^X \ and \ e_a \in E^Y\}$ to denote the pre-aligned entities, and $R_{anchor} := \{r_a \in R^X \ and \ r_a \in R^Y\}$ to denote pre-aligned relations.

For entity alignment task, two main stages are conducted in our framework: 1) the relation selection, which selects the representative types of relations between entities for the alignment task, 2) the knowledge graph embedding, with the representative relations selected, it learns a common embedding for both the knowledge graphs by leveraging representative relations and preserving the proximity of entities with similar head/tails. Also, the anchor entities across knowledge graphs are used as hard constraints for facilitating the transfer of the contextual information.

A. Relation Selection

The relation selection aims to find the relation types which are representative for the entities alignment task. Given two knowledge graphs KG^X and KG^Y, we split the entity pairs across knowledge graphs as two sets: $Pair_{anchor}$ and $Pair_{non\_anchor}$. With the help of anchor entities, two different sets can be obtained: $AS_1 := \{e_i^X, e_i^Y, e_a, r_a\}$ where $\{e_i^X, e_i^Y\} \in Pair_{anchor}$, $e_a \in E_{anchor}$, $r_a \in R_{anchor}$ and $AS_2 := \{e_i^X, e_j^Y, e_a, r_a\}$ where $\{e_i^X, e_j^Y\} \in Pair_{non\_anchor}$, $e_a \in E_{anchor}$, $r_a \in R_{anchor}$

For relation selection, it is intuitive that relation r_a appears more times in AS_1 and less times in AS_2 can be selected as representative relation for the alignment-driven task. Based on this, we use a information gain-like algorithm for rating all the relations, which is shown in equation (1):

$$Score(r_i) = E(AS) - \frac{\#|r_i|}{\#|AS|}E(r_i) - \frac{\#|\bar{r_i}|}{\#|AS|}E(\bar{r_i}) \quad (1)$$

where $\#|AS| = (\#|AS_1| + \#|AS_2|)$ is the total number of the anchor-share sets, $\#|r_i|$ is the number of r_i appears in AS and $\#|\bar{r_i}| = (\#|AS| - \#|r_i|)$. Based on the entropy theory, $E(AS)$, $E(r_i)$ and $E(\bar{r_i})$ are defined as equation (2-4):

$$E(AS) = -\frac{\#|AS_1|}{\#|AS|}\log\frac{\#|AS_1|}{\#|AS|} - \frac{\#|AS_2|}{\#|AS|}\log\frac{\#|AS_2|}{\#|AS|} \quad (2)$$

$$\begin{aligned} E(r_i) \quad = \quad & -\frac{\#|r_i \ in \ AS_1|}{\#|r_i \ in \ AS|}\log\frac{\#|r_i \ in \ AS_1|}{\#|r_i \ in \ AS|} \\ & -\frac{\#|r_i \ in \ AS_2|}{\#|r_i \ in \ AS|}\log\frac{\#|r_i \ in \ AS_2|}{\#|r_i \ in \ AS|} \end{aligned} \quad (3)$$

$$\begin{aligned} E(\bar{r_i}) \quad = \quad & -\frac{\#|r_i \ not \ in \ AS_1|}{\#|r_i \ not \ in \ AS|}\log\frac{\#|r_i \ not \ in \ AS_1|}{\#|r_i \ not \ in \ AS|} \\ & -\frac{\#|r_i \ not \ in \ AS_2|}{\#|r_i \ in \ AS|}\log\frac{\#|r_i \ not \ in \ AS_2|}{\#|r_i \ in \ AS|} \end{aligned} \quad (4)$$

With all the scores $Score(r_i)$ of relations r_i are obtained, a top N representative relations set RR_n can be selected by ranking the scores of all relations.

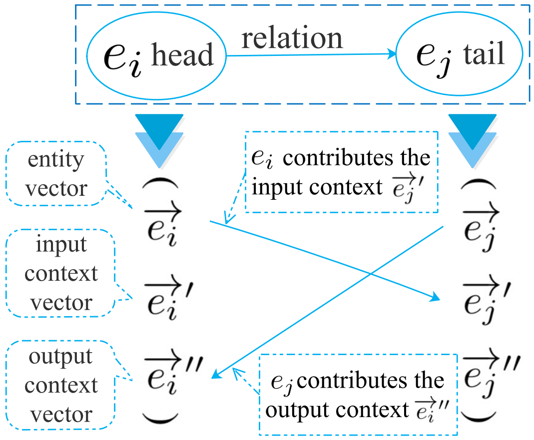

Fig. 1. An example of vector representation

B. Knowledge Graph Embedding With Representative Relations

With the help of selected relations, we try to embed all the entities into a common sub space for the entities alignment. In detail, we use a projection function $f : E \rightarrow R^d$ to represent every entity $e_i \in E$ as a d dimension vector. Since entities may act as different roles (head or tail) in triplets, we define the context of each entity of knowledge graph to be characterized by its own set of heads and tails. In general, we consider 1) the head of triplet as *input context* of the tail entity 2) the tail of triplet as *output context* of the head. Accordingly, every entity node e_i has three vector representations: an entity vector $\overrightarrow{e_i}$, an input context vector $\overrightarrow{e_i}'$ and an output context vector $\overrightarrow{e_i}''$. As illustrate in Fig. 1, e_i is the head node and e_j is the tail node, thus we have the rule that vector $\overrightarrow{e_i}$ contributes the input context vector $\overrightarrow{e_j}'$ as e_j's input context, meanwhile, $\overrightarrow{e_j}$ contributes the output context vector $\overrightarrow{e_i}''$.

In order to learn all vectors of entities, we define the probability that the head entity e_i contributes specially to tail entity e_j as its input context when compared with how e_i contributes to other tail entities, given as:

$$p_1(e_j|e_i) = \frac{\exp(\overrightarrow{e_j}'^T \cdot \overrightarrow{e_i})}{\Sigma_{k=1}^{|V|} \exp(\overrightarrow{e_k}'^T \cdot \overrightarrow{e_i})} \tag{5}$$

where $|V|$ is the number of tail entities which connect to head entity e_i. Similarly, we can define the probability that the tail e_j contributes specially to head entity e_i as its output context compared how with e_j contributes to other head entities:

$$p_2(e_i|e_j) = \frac{\exp(\overrightarrow{e_i}''^T \cdot \overrightarrow{e_j})}{\Sigma_{k=1}^{|V|} \exp(\overrightarrow{e_k}''^T \cdot \overrightarrow{e_j})} \tag{6}$$

Then given two knowledge graph X and Y, we use the pre-aligned (anchor) entities act as "bridge" between two graphs to learn a joint embedding space for the alignment. Specially, we try to achieve the following two objectives:

- **Objective 1**: Entities with selected representative relations have more contribution between each other and structural proximity of entities are preserved in their corresponding vectors as far as possible.
- **Objective 2**: The representations of the anchor entities coincide in the embedded space and those who are close in the embedded space can be considered as good candidates for alignment.

To formulate the objective 1, we define the empirical probabilities $\widehat{p}_1(e_j|e_i)$ and $\widehat{p}_2(e_i|e_j)$ which incorporate the representativeness of relation types for $p_1(e_j|e_i)$ and $p_2(e_i|e_j)$: $\widehat{p}_1(e_j|e_i) = w_{ij}/d_i^{out}$ and $\widehat{p}_2(e_i|e_j) = w_{ij}/d_j^{in}$, where w_{ij}, d_i^{out} and d_j^{in} are given as:

$$w_{ij} = \sum_{r_i, r_j \in (e_i \rightarrow e_j)} \alpha \# |r_i \in RR_n| + \# |r_j \notin RR_n| \tag{7}$$

where α is the representative weight factor of the relations in set RR_n.

$$d_i^{out} = \sum_n \sum_{\substack{r_i, r_j \\ \in (e_i \rightarrow e_n)}} \alpha \# |r_i \in RR_n| + \# |r_j \notin RR_n| \tag{8}$$

$$d_j^{in} = \sum_n \sum_{\substack{r_i, r_j \\ \in (e_n \rightarrow e_j)}} \alpha \# |r_i \in RR_n| + \# |r_j \notin RR_n| \tag{9}$$

Similar with IONE [10], we try to minimize the KL-divergence of p_1 and p_2 and their corresponding empirical probabilities $\widehat{p}_1(e_j|e_i)$ and $\widehat{p}_2(e_i|e_j)$. By properly setting coefficient of the KL-divergence, the the objective function can be given as:

$$\begin{aligned} O_1 = & -\sum_{k \in \{X,Y\}} \sum_{e_i, e_j \in E^k} w_{ij}^k \log p_1(e_j^k|e_i^k) \\ & -\sum_{k \in \{X,Y\}} \sum_{e_i, e_j \in E^k} w_{ij}^k \log p_2(e_i^k|e_j^k). \end{aligned} \tag{10}$$

To formulate the objective 2, we set the vector representations of corresponding anchor entities to be identical, there anchors can act as "bridge" for across graph embedding. With the help of "bridge" entities, the entity e_i^X in graph X can contribute as its input context to e_k^Y's tails in graph Y if e_i^X and e_k^Y are pre-aligned anchor entities and vice versa. Based on this idea, we can write the objective function O_2 similar with O_1:

$$\begin{aligned} O_2 = & -\sum_{e_i \in X} \sum_{(e_k, e_j) \in E^Y} w_{kj}^Y I(e_i^X = e_k^Y) \log p_1(e_j^Y|e_i^X) \\ & -\sum_{e_i \in X} \sum_{(e_k, e_j) \in E^Y} w_{kj}^Y I(e_i^X = e_k^Y) \log p_2(e_i^X|e_j^Y) \\ & -\sum_{e_i \in Y} \sum_{(e_k, e_j) \in E^X} w_{kj}^X I(e_i^Y = e_k^X) \log p_1(e_j^X|e_i^Y) \\ & -\sum_{e_i \in Y} \sum_{(e_k, e_j) \in E^X} w_{kj}^X I(e_i^Y = e_k^X) \log p_2(e_i^Y|e_j^X) \end{aligned} \tag{11}$$

where $I(e_i^Y = e_k^X)$ means e_i^Y and e_k^X are pre-aligned anchors.

Thus the final framework can be computed by minimizing a combined objective function $O = O_1 + O_2$ w.r.t. $\{\vec{e_x}^X, \vec{e_x}'^X, \vec{e_x}''^X, \vec{e_y}^Y, \vec{e_y}'^Y, \vec{e_y}''^Y\}$

C. Model Inference

Stochastic gradient descent algorithm is used for learning the representation of two knowledge graphs. To update the node vector of e_i and e_j in knowledge X, i.e., $\vec{e_i}^X$, $\vec{e_j}^X$, the gradient can be computed as:

$$\begin{aligned}\frac{\partial O}{\partial \vec{e_i}^X} &= \frac{\partial O_1}{\partial \vec{e_i}^X} + \frac{\partial O_2}{\partial \vec{e_i}^X} \\ &= \sum_{e_i,e_j\in E^X} w_{ij}^X * \frac{\partial \log p_1(e_j^X|e_i^X)}{\partial \vec{e_i}^X} \\ &+ \sum_{e_i\in E^X}\sum_{e_k,e_j\in E^Y} w_{kj}^Y * I(e_i^X = e_k^Y)\frac{\partial \log p_1(e_j^Y|e_i^X)}{\partial \vec{e_i}^X}\end{aligned} \tag{12}$$

The partial derivatives w.r.t. the input and output context vectors can be obtained similarly.

Here we adopt the negative sampling algorithm for reducing the calculation complexity for p_1 and p_2. The equivalent counterparts of objective function can ben given as:

$$\begin{aligned}\log p_1(e_j^X|e_i^X) \quad \propto \quad & \log\sigma(\vec{e_j}'^{X^T}\cdot \vec{e_i}^X) \\ & + \Sigma_{m=1}^K E_{v_n\sim p_n(v)} \log\sigma(-\vec{e_n}'^{X^T}\cdot \vec{e_i}^X)\end{aligned} \tag{13}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, K is the number of negative samples e_n which is sampled from the "noisy distribution" of $p_n(e) = d_e^{3/4}$ as in [15], and d_e is the output degree.

With the negative sampling adopted, once that e_i, e_j in one triple and a negative node e_n are sampled in two graphs, the partial derivative of Eq.(12) w.r.t. $\vec{e_i}$ can be rewritten as:

$$\begin{aligned}\frac{\partial O}{\partial \vec{e_i}^X} \quad = \quad & [1-\sigma(\vec{e_j}'^{X^T}\cdot \vec{e_i}^X)]\vec{e_j}'^X - \sigma(\vec{e_n}'^{X^T}\cdot \vec{e_i}^X)\vec{e_n}'^X \\ & + \sum_{e_k,e_j\in E^Y} I(e_i^X = e_k^Y) * \\ & \{[1-\sigma(\vec{e_j}'^{Y^T}\cdot \vec{e_i}^X)]\vec{e_j}'^Y - \sigma(\vec{e_n}'^{Y^T}\cdot \vec{e_i}^X)\vec{e_n}'^Y\}\end{aligned} \tag{14}$$

Similarly, we can obtain the other partial derivatives w.r.t. the other context vectors of the concerned entities.

Finally, the overall learning algorithm for EARR is shown in Algorithm 1.

D. Time Complexity

For the edge selection, it takes $O(|E^X| * E|Y|)$ to get the anchor entity pairs and other entity pairs, meanwhile, we can get the scores of all the relations. Ranking the Top N relations will take $O(|E^X| * E|Y| * logN)$, the overall time will be approximate to $O(|E^X| * E|Y| * logN)$.

For the across knowledge graph embedding, it takes $O(1)$ to sample an edge. Optimization using K negative samples takes $O(d(K+1))$ time, where d is the dimension. Therefore, the overall complexity for each step is $O(dK)$. In practice,

Algorithm 1 The EARR Learning Algorithm

Require: Two knowledge graphs KG^X and KG^Y, a set of pre-aligned entities E_a, learning rate η, $\#$ of negative samples K, a representative relations set RR_n.
Ensure: The set of estimated parameters $\Theta = \{\vec{e_i}, \vec{e_j}, \vec{e_i}'', \vec{e_j}', \vec{e_n}', \vec{e_n}''\}$

procedure LEARNING(KG^X,KG^Y, E_a, η, K)
  Evaluate the score of every relation r_i based on Eqs.(1)
  Get the representative relation set RR_n by ranking the scores of relations.
  Initialize $\Theta = \{\vec{e_x}^X, \vec{e_x}'^X, \vec{e_x}''^X, \vec{e_y}^Y, \vec{e_y}'^Y, \vec{e_y}''^Y\}$
  repeat
    for N in (X, Y) **do**
      Sample one edge (e_i,e_j) from G^N
      Update $\vec{e_i}$, based on Eqs.(14) et al.
      for $i = 0; i < K; i = i + 1$ **do**
        Sample a negative node e_n
        Update $\vec{e_i}$ based on Eqs.(14) et al.
      end for
    end for
  until convergence
  return Θ
end procedure

the number of steps need for the optimization is usually proportional to the number of edges $|E|$ [15]. Therefore, the overall time complexity of our model is $O(dK|E|)$ which is linear to the number of edges $|E|$ and does not depend on the number of nodes $|V|$.

E. Aligning Entities across Knowledge Graphs

To map entities across networks, we compute cosine similarity between vector representations of one entities in network X and another in Y to determine their correspondence.

$$rel(e_i^X, e_j^Y) = \frac{\sum_{p=1}^d e_{ip}^X \times e_{jp}^Y}{\sqrt{\sum_{p=1}^d {e_{ip}^X}^2} \times \sqrt{\sum_{p=1}^d {e_{jp}^Y}^2}} \tag{15}$$

So, for each entity e_i^X in graph X, we can find the most relevant entity e_j^Y in network Y to be an anchor candidate.

IV. EXPERIMENTS

For evaluating the performance of the model proposed in this paper, we employ a trilingual knowledge graph dataset WK31-15K [9] including English(En), German(De) and French(Fr) knowledge graphs which are extracted from DBpedia with known aligned entities as groundtruth. The statistics of dataset are listed in Table I.

Several state-of-the-art methods are selected as the baselines for the comparison, including:

- **ITransE**: A pTransE-based method [4] which performs entity alignment by inferring a linear transformation in the embedding spaces for different knowledge graphs.
- **NTAM**: A non-translational alignment model proposed in [9], which utilizes a probabilistic model to project each

TABLE I
STATISTICS OF THE DATASETS USED FOR EVALUATION

Graphs	#Entity	#Relations	#Triplet	#Anchors
WK31-15K-En-De	15,617(En)	1,935(En)	217,860(En)	2070
	15,167(En)	1,173(En)	156,452(En)	
WK31-15K-En-Fr	15,517(En)	2,488(En)	215,301(En)	3116
	16,676(Fr)	2,735(Fr)	189,416(Fr)	

network onto the same vector space. NTAM can learn the entity embedding and relation embedding simultaneously.

- **NTAM-2**: A variant of proposed NTAM which requires only second-order connectivities to be preserved in knowledge embedding. Its joint embedding component is the same as NTAM.
- **NTAM-1**: Differing from NTAM-2, NTAM-1 requires only the first-order connectivity to be preserved in knowledge embedding.
- **IONE**: A social network alignment approach [10], which consider the relation between users as single-type relation. In order to fit the entity alignment task, we consider the relations as the same type and sum all relation number as the weight between entities.

For IONE, NTAMs and the EARR, the iteration number is set as $3 * 10^9$. The learning rate of IONE and the EARR is set as 0.025, and 0.2 for NTAM the same as they used in the work [9]. The dimension is set as 100 for all the models. The size of representative relations set is set as 50.

A. Evaluation Metrics

We use $Hits@N$ as the evaluation metrics. Since there are different results when different knowledge graph is set as the source graph, we try to evaluate all the results by different source graph settings. The metrics given as:

$$Hits@N^{X \to Y / Y \to X} = \frac{|CorrEntity@N|^{X/Y}}{|UnAlignedAnchors|} \quad (16)$$

where $|CorrEntity@N|^{X/Y}$ is the number of unaligned anchor entities with their corresponding entities found among the top N list in the other graph and $|UnAlignedAnchors|$ is the number of unaligned anchor entities.

B. Experimental Results

We firstly use 80% of pre-aligned entities as the training data and 20% for testing. The results of all the models are tabulated in Table II. From which we can see that, the EARR outperforms all the baselines in most cases. For all baselines, NTAMs and IONE perform more robust than ITransE, they outperform ITransE in all the datasets except $Hits@1$ in Fr-En. One possible reason is that the non-translational embedding approaches always fail in dealing with some complex relation structure such as triangulars and other complex link structures with more than one path from head to tail [9]. Besides, there is a triplets re-forming process in ITransE model, which generates new triples by replacing head or tail in one knowledge graph based on its counterpart in another knowledge graphs. This process may introduce the facts does not exists in the knowledge graph. IONE performs better than NTAMs models. We argue that NTAMs model only consider one direction of relation between head and tails, which may lose some useful information as sharing similar set of heads and tails can identify the same entities more precisely. The EARR performs better than IONE except the $Hits@1$ metric in Fr-En, we believe that the representative relations benefit the entity alignment especially in $Hits@1$ in En and De knowledge graph. For the exception $Hits@1$ metric in Fr-En, we found that one reason is the $Scores$ of relations are small in Fr-En graph. We roughly choose Top 50 as representative relation set may produce some "noisy" relations. A heuristic method should be introduced for effectively selecting RR_n set. We leave it as our feature work.

In order to evaluate the performance of the EARR with different train-to-test settings, we split the anchor entities with different training ratios varying [10%-90%], the rest are used as testing sets. Fig 2 illustrates the results, from which we can see that when the training ratio is low, the EARR does not perform well. But with increasing the training ratio, it shows sharp increasing tendency. When the training ratio higher than 50%, the EARR outperforms all the baselines. The reason is that, when there is not enough anchor pairs, it is hard to select the RR_n set as the relations between training anchor are not "diverse" enough for covering all the representative relations and some useless relations may be introduced by a large number of non_anchor pairs.

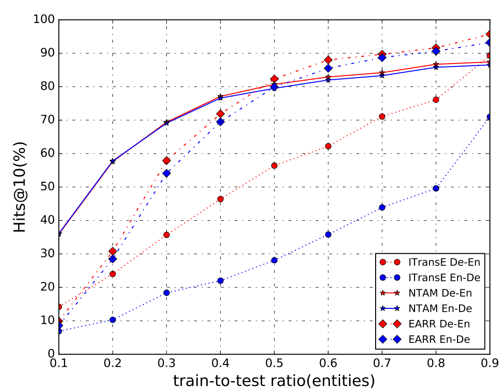

Fig. 2. Performance of Different Train-to-Test Settings

As there is an iteration process in the models, whether the model can be converged is an important issue. For the iteration evaluation, En-De dataset is adopted, 80% of anchors used as the training set, and the rest as testing . $Hits@1$ is used for metric. Note that in order to compare the models in the same scale, we exclude IPTransE for the comparison as the model uses a different way for modeling the objective. Fig 3 illustrates the results. We can see that all the models can be converged when the iteration reaches $3 * 10^9$. Both IONE and EARR converge faster than NTAM, which means we need less time for the training and the models are more effective.

TABLE II
PERFORMANCE OF ENTITY ALIGNMENT

Aligned Language	En-De		De-En		En-Fr		Fr-En	
Metric	$Hits@1$	$Hits@10$	$Hits@1$	$Hits@10$	$Hits@1$	$Hits@10$	$Hits@1$	$Hits@10$
ITransE	29.4	49.6	50.4	76.1	55.0	80.0	54.6	80.8
NTAM-1	66.5	83.4	54.9	82.7	62.0	86.9	38.8	85.3
NTAM-2	58.3	76.7	42.4	73.3	60.6	86.2	39.3	84.0
NTAM	66.7	85.8	73.3	86.7	73.6	92.9	62.7	81.1
IONE	72.3	89.4	78.3	90.8	79.9	93.9	**82.6**	93.4
EARR	**75.4**	**90.6**	**80.0**	**91.6**	**81.3**	**94.4**	81.4	**93.6**

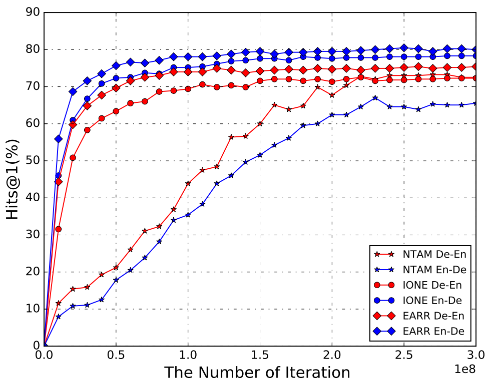

Fig. 3. Performance of Different Train-to-Test Settings

V. CONCLUSION

In this paper, we propose a knowledge graph embedding model for the entity alignment. With multi-type relations between entities are took into consideration, an alignment-task driven representative relation selection algorithm are proposed. Then a probabilistic model is adopted for modeling bi-directional relations with the help of context information transfer between anchors entities. For effective model inference, negative sampling and stochastic gradient descent are used for parameter learning. Experiment on real-world dataset show that the proposed model (EARR) have a better performance comparing with several state-of-the-art methods. In future work, we will try to find a heuristic way for selecting representative relation set and embedding the entities and relations simultaneously.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 6180603, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grants KJ1717371, KJQN201801907, KJQN201800638, KJ1717370, and in part by Chongqing Research Program of Basic Research and Frontier Technology under Grants cstc2017jcyjAX0325, cstc2017jcyjAX0406 and cstc2018jcyjAX0151 and we thanks the authors of NTAM[9] for sharing baseline codes to us.

REFERENCES

[1] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Soren Auer, et al. Dbpediaa large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web, 6(2):167195, 2015.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of KDD, pages 12471250, 2008.

[3] 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, IJCAI, Melbourne, Australia, August 19-25, 2017, pages 1511-1517, 2017.

[4] 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, Melbourne, Australia, August 19-25 , pages 4258-4264, 2017.

[5] Hao Y, Zhang Y, He S, et al. A joint embedding method for entity alignment of knowledge bases. In Proceedings of China Conference on Knowledge Graph and Semantic Computing. Springer, Singapore, pages 3-14, 2016.

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

[7] Antoine Bordes, Nicolas Usunier,Alberto Garca-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems, pages 2787-2795, 2013.

[8] Lin Liu, Xin Li, William K. Cheung, and Chengcheng Xu. A structural representation learning for multi-relational networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25 , pages 4047-4053, 2017.

[9] Shengnan Li, Xin Li, Rui Ye, Mingzhong Wang, Haiping Su, Yingzi Ou. Non-translational Alignment for Multi-relational Networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, Stockholm, Sweden, July 13-19, pages 4180-4186, 2018.

[10] Li Liu, William K. Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI, New York, NY, USA, 9-15 July , pages 1774-1780, 2016.

[11] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, July 27 -31, Quebec City, Quebec, Canada., pages 1112-1119, 2014.

[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, January 25-30,Austin, Texas, USA. , pages 2181-2187, 2015.

[13] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, October 25-29, Doha, Qatar, pages 1591-1601, 2014.

[14] Zequn Sun,Wei Hu, Qingheng Zhang and Yuzhong Qu. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, July 13-19, pages 4396-4402, 2018.

[15] Tang J, Qu M, Wang M, et al. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pages: 1067-1077, 2015.