Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Structured Embedding via Pairwise Relations and Long-Range Interactions in Knowledge Base

Fei Wu and **Jun Song**
College of Computer Science
Zhejiang University, China

Yi Yang
Centre for Quantum Computation and Intelligent Systems
University of Technology, Sydney

Xi Li
College of Computer Science
Zhejiang University, China

Zhongfei Zhang
Department of Information Science and Electronic Engineering
Zhejiang University, China

Yueting Zhuang
College of Computer Science
Zhejiang University, China

Abstract

We consider the problem of embedding entities and relations of knowledge bases into low-dimensional continuous vector spaces (*distributed representations*). Unlike most existing approaches, which are primarily efficient for modelling pairwise relations between entities, we attempt to explicitly model both pairwise relations and long-range interactions between entities, by interpreting them as linear operators on the low-dimensional embeddings of the entities. Therefore, in this paper we introduces *path ranking* to capture the long-range interactions of knowledge graph and at the same time preserve the pairwise relations of knowledge graph; we call it *s*tructured *e*mbedding via *p*airwise relation and *l*ong-range *i*nteractions (referred to as SePLi). Comparing with the-state-of-the-art models, SePLi achieves better performances of embeddings.

Introduction

A *knowledge graph* such as Freebase (Bollacker et al. 2008), WordNet (Miller 1995), and Yago (Suchanek, Kasneci, and Weikum 2007) is a multi-relational graph consisting of entities as nodes and relations as different types of labelled edges. An instance of the labelled edge is a triplet of one fact (*head entity*, *relation*, *tail entity*) (abbreviated as (h, r, t)) which indicates that there exists a *pairwise* relation of name *relation* between the entities *head entity* and *tail entity* (e.g., (Paris, capital_of, France)). In the past decade, great advance is achieved to embed elements of a knowledge graph into a continuous space (*distributed representation*) while preserving the intrinsic structures of the original graph.

The distributed representation of the words in a continuous space has been used in *Natural Language Processing* (NLP) via the framework of language models where an embedding per word is learnt in an unsupervised learning fashion rather than merely one-hot representation of a word (Xu and Rudnicky 2000; Bengio Y, P, and C 2003; Bengio et al. 2006; Bengio 2008). A similar algorithm is utilized to represent concepts as vectors and relations as matrices in *Linear Relation Embedding* (LRE) (Paccanaro 2000; Paccanaro and Hinton 2001) which maps entities into a low-dimensional space by imposing the constraint that relations in this low-dimensional space are modeled by linear operations. In other words, entities are modeled by real-valued vectors and relations by matrices. Parameters of both entities and relations are learnt during training. It has been shown that encoding knowledge graph in distributed embeddings improves the performance.

For examples, Figure 1 illustrates a family tree via a knowledge graph, where entities (family members) and their pairwise relations (i.e., wife and daughter) are encoded. In this paper, structured embedding is employed to map all of entities in the knowledge graph to their continuous vectors, and the pairwise relations are represented as matrixes.

Given a triplet (h, r, t), the embeddings of the entities and the relations are denoted with the same letter in boldface characters in this paper (i.e., $\boldsymbol{h}$,$\boldsymbol{r}$,$\boldsymbol{t}$). *Structured Embeddings* (SE) (Bordes et al. 2011) embeds entities h and t into $\mathbb{R}^d$ and relation r into two matrices $\boldsymbol{L_1} \in \boldsymbol{R}^{d\times d}$ and $\boldsymbol{L_2} \in \boldsymbol{R}^{d\times d}$ such that the dissimilarly (measured by l_1 distance) between $\boldsymbol{L_1} \cdot \boldsymbol{h}$ and $\boldsymbol{L_2} \cdot \boldsymbol{t}$ is less than that of the corrupted triples. The basic idea of structured embeddings (Bordes et al. 2011) is that when two entities belong to one triplet, their individual embeddings should be close to each other in a subspace that depends on their assigned relation. A semantic matching energy function is devised via a neural network and utilized in (Bordes et al. 2012; 2014) to map all of entities and relations into a same relatively low-dimensional embedding vector space, where the plausible triplets of a multi-relational graph are assigned low energies (i.e. high probabilities). Since entities and relations all share the same kind of representation, the usual conceptual difference between entities and relation are diminished. Another more expressive embedding approach is *Neural Tensor Networks* (NTN) proposed in (Socher et al. 2013). NTN replaces the traditional linear neural network layer with a bilinear tensor layer that directly mediates the

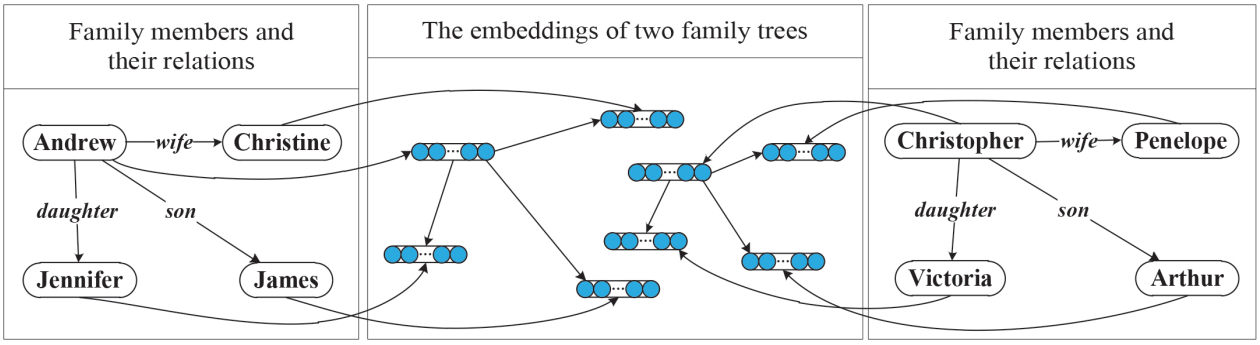


Figure 1: The embeddings of a knowledge graph. Two individual family trees are embedded into a common continuous vector spaces. Persons are represented as different vectors and family relations are represented as matrixes.

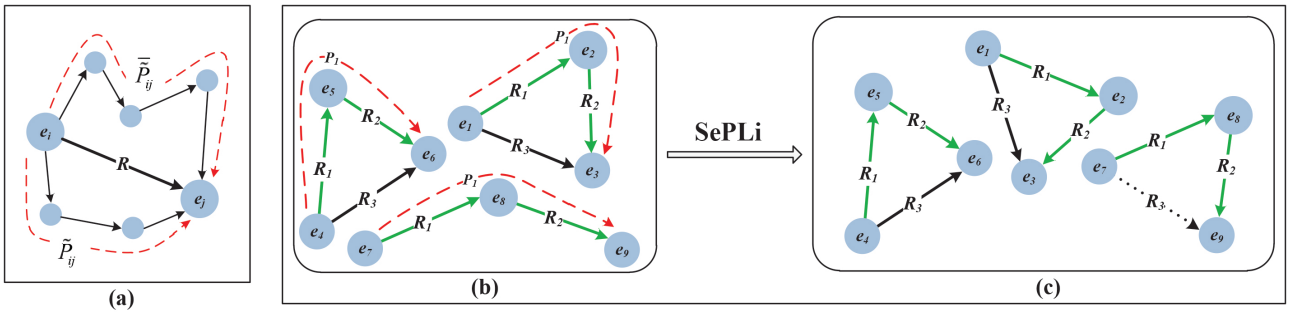


Figure 2: The intuitive illustration of the structured embedding of knowledge graph. In subfigure (a), the embeddings of the entities e_i and e_j will be decided by both the pairwise relation R and the long-range interactions $\tilde{P}_{ij}$ and $\tilde{\tilde{P}}_{ij}$. In subfigure (b), there are in total 9 entities $e_i (1 \leq i \leq 9)$ and 3 relations R_j $(1 \leq j \leq 3)$ in an incomplete knowledge graph. A *long-range interaction path* P_1 is defined as a sequence of pairwise relations $R_1 \rightarrow R_2$. Given one pair of source entity and target entity, there are one pairwise relation and long-range interaction path (such as $e_4 \xrightarrow{R_3} e_6$ and $e_4 \xrightarrow{P_1} e_6$, $e_1 \xrightarrow{R_3} e_3$ and $e_1 \xrightarrow{P_1} e_3$). The proposed SePLi can utilize the relevance between the pairwise relation R_3 and the long-range interaction P_1. As a result, , SePLi is able to predict the fact (e_7, R_3, e_9) by the minimizations of both pairwise loss and long-range loss. Here the *long-range interaction path* P_1 could be regarded as the composition, transitivity or inheritance of the pairwise relation R_3.

interactions between the two entity vectors across multiple dimensions (aspects).

In general, the embeddings of entities are learnt via a multi-tasking setting in the aforementioned approaches and the embedding of an entity contains factorized information coming from all the relations in which the entity is remarked. On the other hand, the embedding of a relation also contains factorized information from all the triplets in which the relation is involved. The interactions between entities and relations are the useful information to obtain appropriate embeddings of a knowledge graph. However, only *pairwise* relations are exploited during the learning of distributed representations in the aforementioned approaches. Although pairwise relations are essential to preserve the local structures in a multi-relational graph, the long-range interactions (e.g., *transitivity* or *composition* or *inheritance* relations encoded by a sequence of pairwise relations) are beneficial to uncover the underlying structures beyond pairwise relations.

We argue that both pairwise relations and long-range interactions between entities can boost the performance of embeddings of a knowledge graph. Therefore, this paper introduces *path ranking* (Lao and Cohen 2010b; Lao, Mitchell, and Cohen 2011) to capture the underlying long-range interactions of the knowledge graph and at the same time to preserve the pairwise relations of the knowledge graph, we call it *s*tructured *e*mbedding via *p*airwise relations and *l*ong-range *i*nteractions (referred to as SePLi). SePLi not only utilizes the pairwise relations, but also captures the intrinsic long-range dependent interactions between entities. Since the embeddings in SePLi is employed by the linear operators, SePLi provides a scalable way for large numbers of entities and relation types.

Our Approach

For a knowledge graph $\mathcal{K} =< \mathcal{E}, \mathcal{R}, \mathcal{F} >$, $\mathcal{E} = \{e^i\}_{i=1}^{N_e}$ is a set of all entities, $\mathcal{R} = \{R_j\}_{j=1}^{N_R}$ is a set of all pairwise relations and $\mathcal{F} = \{x^k\}_{k=1}^{N_F}$ contains all the facts (denoted as triplets) that can be observed in $\mathcal{K}$. x^i is the i-th fact, and $x^i = (e_{i,h}, R_i, e_{i,t}), e_{i,h}, e_{i,t} \in \mathcal{E}, R_i \in \mathcal{R}$. Let $\boldsymbol{e_{i,h}}, \boldsymbol{e_{i,t}} \in \mathbb{R}^d$ be the vector representations of the two entities $e_{i,h}$ and $e_{i,t}$, and $\boldsymbol{W_i} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b_i} \in \mathbb{R}^d$ are the embedding matrix

and the bias vector corresponding to the relation R_i in fact x^i, respectively.

Pairwise relations and long-range interactions

Given one fact x^i in the term of $(e_{i,h}, R_i, e_{i,t})$, we attempt to learn a linear function $g(\cdot)$ so that the embedding of $e_{i,h}$ is expected to be similar to that of $e_{i,t}$. The linear function $g(\cdot)$ is defined in equation (1) as follows:

$$g(R_i, \boldsymbol{e_{i,h}}) = \boldsymbol{W_i} \cdot \boldsymbol{e_{i,h}} + \boldsymbol{b_i} \tag{1}$$

In this paper, we argue that the appropriate utilization of both pairwise relations and long-range interactions are vital to improve the learning of the parameters of the linear function $g(\cdot)$ as well as the embedding vectors of the entities.

- *Pairwise* relation: In a multi-relational graph, the pairwise relation is a direct labelled edge between the two entities defined by a triplet of one fact (h, r, t). Here we call the entities h and t as a pairwise relation r. Since we expect that the result of a linear operation with the embedding vector of h is approximately the same as the embedding vector of t, the *pairwise loss* is therefore defined as the difference between $g(r, h)$ and t. The less the pairwise loss is, the better the linear function $g(\cdot)$ is.
- *Long-range* interaction: given one fact (h, r, t), we can probably find some long-range paths $\{P_i\}_{i=1}^k$ from the source vertex h to the target vertex t. k denotes the number of the path to be found. These paths $\{P_i\}_{i=1}^k$ indicate the long-range interactions between h and t. Since the long-range interactions $\{P_i\}_{i=1}^k$ and the pairwise relation r, respectively, describe the relation between the entities h and t from different aspects (direct versus implicit), the long-range interactions $\{P_i\}_{i=1}^k$ and the pairwise relation r should be relevant. Therefore, the *long-range loss* is defined as the difference of the pairwise relation r and long-range paths with the same source entity and target entity.

Since the proposed SePLi attempts to utilize both pairwise relations and long-range interactions to learn the linear function $g(\cdot)$ and the embeddings, the objective function of the proposed SePLi is defined as follows:

$$O(\Theta) = \underbrace{\mathcal{J}_1(\Theta)}_{\text{pairwise loss}} + \underbrace{\lambda_1 \mathcal{J}_2(\Theta)}_{\text{long-range loss}} + \frac{\lambda_2}{2}||\Theta||^2 \tag{2}$$

where Θ is the collection of all the model parameters. A better embedding tends to minimize the objective function in the training phase. Here, $\mathcal{J}_1(\Theta)$ and $\mathcal{J}_2(\Theta)$ are the terms to encode the pairwise loss and long-range loss, respectively. λ_1 is the weighting parameter of long-range loss. The last component with the hyper-parameter λ_2 is the standard L_2 regularization on all the model parameters. They will be described in more details in the next sections.

Minimizing the pairwise loss

As aforementioned, given $(e_{i,h}, R_i, e_{i,t})$, in order to minimize the pairwise loss, the linear operation of the embedding of $e_{i,h}$ with $g(\cdot)$ is expected to be the same as that of $e_{i,t}$, and we adopt the squared distance to measure the confidence of each fact x^i when $g(\cdot)$ is implied to x^i:

$$s(x^i) = \frac{1}{2}||g(R_i, \boldsymbol{e_{i,h}}) - \boldsymbol{e_{i,t}}||^2 \tag{3}$$

Like *Structured embeddings* (SE) (Bordes et al. 2011) and *Neural Tensor Networks* (NTN) (Socher et al. 2013), the optimal $g(\cdot)$ and the embeddings can be achieved in term of the contrastive max-margin learning. The main idea here is to ensure the positive facts which can be observed in the knowledge graph have a higher confidence score (less squared distance by equation (3)) than negative (false) facts which are constructed by replacing the *head entity* or the *tail entity* into a random entity from the set of all the entities. As a result, the pairwise relation of one given fact x^i is constrained by the following two constraints:

$$s(e_{i,h}, R_i, e_{i,t}) < s(e_{j,h}, R_i, e_{i,t}), \forall j : (e_{j,l}, R_i, e_{i,t}) \notin \mathcal{F} \tag{4}$$

$$s(e_{i,h}, R_i, e_{i,t}) < s(e_{i,h}, R_i, e_{j,t}), \forall j : (e_{i,h}, R_i, e_{j,t}) \notin \mathcal{F} \tag{5}$$

The value of $s(e_{i,h}, R_i, e_{i,t})$ is calculated according to equation (3).

Towards guaranteeing the constraints of all the observable facts, the pairwise loss of knowledge graph is defined as follows in equation (6):

$$\mathcal{J}_1(\Theta) = \sum_{i=1}^{N^{tr}} \sum_{c=1}^{C_i} \max(0, s(x^i) + 1 - s(x_c^i)) \tag{6}$$

While N^{tr} denotes the number of the positive facts. C_i is the number of the negative facts that are generated from positive fact x^i, and x_c^i is the c-th negative fact. We set the margin between positive facts and negative facts to 1.

Minimizing the long-range loss

In a knowledge graph, the underlying long-range interactions are usually important. For examples, the pairwise relation 'grandfather' is relevant to the consequent combination of two pairwise relation 'father' in term of inheritance; or the pairwise relation 'place_of_birth' is a compositional concept of the concatenation of pairwise relation 'parents' and 'nationality'.

Taking the fact $x^i = (e_{i,h}, R_i, e_{i,t})$ as an example, although the entities $e_{i,h}$ and $e_{i,t}$ are labelled as the pairwise relation R_i, we also find a long-range path relation P from $e_{i,h}$ to $e_{i,t}$. Here P can be denoted as a sequence of pairwise relations between entities identified for the path P. The long-range interaction P is relevant to the pairwise relation R_i because they share the same source and target entities.

For each pairwise relation R_i, assume that we find the top K paths P_j^i with weights p_j^i ($1 \leq j \leq K$). In order to find the top K paths P_j^i, a walker samples uniformly from the neighbours of the last visited vertex until the maximum length is reached. While we set the maximum length of our random walks in the experiments to be fixed, there is no restriction for the random walks to be of the same length. For the j-th path P_j^i which is a sequence of M_j pairwise relations $R_1^{P_j^i}, \cdots, R_{M_j}^{P_j^i}$, when starting from entity $e_{i,h}$, the

embedding vector of next entity $\boldsymbol{v}^{\boldsymbol{1,i,j}}_{\boldsymbol{e_{i,h}}}$ is approximately obtained by equation (7). $R^{P^i_j}_m$ is the m-th pairwise relation in the pairwise relation sequence P^i_j. $\boldsymbol{W}^{\boldsymbol{P^i_j}}_{\boldsymbol{m}}$ and $\boldsymbol{b}^{\boldsymbol{P^i_j}}_{\boldsymbol{m}}$ denote the embedding matrix and bias vector of the pairwise relation $R^{P^i_j}_m$, respectively.

$$\boldsymbol{v}^{\boldsymbol{1,i,j}}_{\boldsymbol{e_{i,h}}} = g(R^{P^i_j}_1, \boldsymbol{e_{i,h}}) = \boldsymbol{W}^{\boldsymbol{P^i_j}}_{\boldsymbol{1}} \cdot \boldsymbol{e_{i,h}} + \boldsymbol{b}^{\boldsymbol{P^i_j}}_{\boldsymbol{1}} \tag{7}$$

Then at the second step, we obtain the embedding vector of $\boldsymbol{v}^{\boldsymbol{2,i,j}}_{\boldsymbol{e_{i,h}}}$ by equation (8).

$$\boldsymbol{v}^{\boldsymbol{2,i,j}}_{\boldsymbol{e_{i,h}}} = g(R^{P^i_j}_2, \boldsymbol{v}^{\boldsymbol{1,i,j}}_{\boldsymbol{e_{i,h}}}) \tag{8}$$

$$= \boldsymbol{W}^{\boldsymbol{P^i_j}}_{\boldsymbol{2}} \cdot (\boldsymbol{W}^{\boldsymbol{P^i_j}}_{\boldsymbol{1}} \cdot \boldsymbol{e_{i,h}} + \boldsymbol{b}^{\boldsymbol{P^i_j}}_{\boldsymbol{1}}) + \boldsymbol{b}^{\boldsymbol{P^i_j}}_{\boldsymbol{2}} \tag{9}$$

After M_j steps, the embedding vector of the target entity $e_{i,t}$, $\boldsymbol{v}^{\boldsymbol{M_j,i,j}}_{\boldsymbol{e_{i,h}}}$ is recursively obtained. Combining all the weighted K paths we have found before, we have equation (10).

$$f(R_i, e_{i,h}) = \frac{1}{\sum_{j=1}^{K} p_j} \sum_{j=1}^{K} p_j \cdot \boldsymbol{v}^{\boldsymbol{M_j,i,j}}_{\boldsymbol{e_{i,h}}} \tag{10}$$

Therefore, for $(e_{i,h}, R_i, e_{i,t})$, the embedding vector of the target entity $e_{i,t}$ either can be approximated by one linear operation $g(R_i, \boldsymbol{e_{i,h}})$ via the pairwise relation, or by a sequence of linear operations $f(R_i, e_{i,h})$ via long-range interactions. As a result, we expect that the difference of $g(R_i, \boldsymbol{e_{i,h}})$ and $f(R_i, e_{i,h})$ is the minimum. We propose to minimize the following objective function:

$$\gamma(R_i) = ||\boldsymbol{W_i} - \boldsymbol{W^{long}_i}||^2 + ||\boldsymbol{b_i} - \boldsymbol{b^{long}_i}||^2 \tag{11}$$

Here $\boldsymbol{W^{long}_i}$ and $\boldsymbol{b^{long}_i}$ are defined as follows:

$$\boldsymbol{W^{long}_i} = \frac{1}{\sum_{j=1}^{K} p_j} \sum_{j=1}^{K} p_j \prod_{m=1}^{M_j} \boldsymbol{W}^{\boldsymbol{P^i_j}}_{\boldsymbol{m}} \tag{12}$$

$$\boldsymbol{b^{long}_i} = \frac{1}{\sum_{j=1}^{K} p_j} \sum_{j=1}^{K} p_j B(P^i_j) \tag{13}$$

where $B(P^i_j)$ is defined by equation (14).

$$B(P^i_j) = \sum_{m=1}^{M_j} (\prod_{q=m+1}^{M_j} \boldsymbol{W}^{\boldsymbol{P^i_j}}_{\boldsymbol{q}}) \cdot \boldsymbol{b}^{\boldsymbol{P^i_j}}_{\boldsymbol{m}} \tag{14}$$

Then the long-range loss is defined as follows in (15).

$$\mathcal{J}_2(\Theta) = \frac{1}{2} \sum_{i=1}^{N_R} \gamma(R^i) \tag{15}$$

Therefore, the optimization of those linear functions and embeddings of entities are obtained by minimizing the objective function as follows:

$$O(\Theta) = \underbrace{\sum_{i=1}^{N^{tr}} \sum_{c=1}^{C_i} \max(0, s(x^i) + 1 - s(x^i_c))}_{\text{pairwise loss}} + \underbrace{\frac{\lambda_1}{2} \sum_{i=1}^{N_R} \gamma(R^i)}_{\text{long-range loss}} + \tfrac{\lambda_2}{2} ||\Theta||^2 \tag{16}$$

Algorithm 1 summarizes the procedure of our proposed SePLi .

Algorithm 1: structured embedding in SePLi

Input: The set of training facts $\mathcal{F}^{tr}$
initialization;
finding long-range paths for each pairwise relation;
while *maximum number of iterations not exceeded* **do**
  foreach x^i *in* $\mathcal{F}^{tr}$ **do**
    generate negative facts set $\{x^i_c\}^{C_i}_{c=1}$;
    minimizing pairwise loss relevant to x^i;
  end
  foreach R^i *in* $\mathcal{R}$ **do**
    minimizing long-range loss relevant to R^i
  end
end
Output: structured embeddings of entities and relations

Finding long-range interaction

Given a set of pairs of entities that are directly pairwise relation R, *Path Ranking Algorithm* (PRA) (Lao, Mitchell, and Cohen 2011; Lao and Cohen 2010b) then performs a random walk on the knowledge graph to identify the long-range paths starting at all the source entities (nodes). The paths that reach the target entities (nodes) are considered successful.

While we set the largest path length of our random walks in the experiments to be fixed as L , there is no restriction for the random walks to be of the same length. Taking one long-range path $P = R_1, R_2, \cdots, R_l, (l \leq L)$ for example, the distribution $h_{e_s,P}$ is recursively defines by (17).

$$h_{e_s,P}(e) = \sum_{e' \in range(P')} h_{e_s,P'}(e') \cdot P(e|e'; R_l) \tag{17}$$

The distribution $h_{e_s,P}(e)$ indicates the probability such that e is in the path connecting the source entity e_s. $P(e|e'; R_l) = \frac{R_l(e',e)}{|R_l(e',\cdot)|}$ is the probability that the entity e is connected to the entity e' with one step distance as defined by the relation R_l. $R_l(e', e)$ is the indicator that equals to 1 when entity e' can be connected to the entity e via relation R_l, otherwise 0. $|R_l(e', \cdot)|$ is the number of entities that entity e' can connect to via relation R_l. The set of entities that can be reached from the source entity though path P' is denoted as $range(P')$.

Given a set of paths $P_1, \cdots, P_n$, each $\{h_{e_s,P_i}(e)\}_{i=1}^n$ could be treated as the path features for certain fact (e_s, R, e). Thus, we conduct *Logistic Regression* to train the appropriate weights of paths $P_1, \cdots, P_n$. We find negative training samples in the same way as (Lao and Cohen 2010a) does.

Finally, the long-range paths which are relevant to the pairwise relation R as well as their weights are used in equation (15) to minimize the long-range loss.

Experiments and Results

We compare our SePLi with several state-of-the-art knowledge embedding methods as follows:

- **NTN**(Socher et al. 2013): The *Neural Tensor Network* (NTN) replaces a standard linear neural network layer with a bilinear tensor layer that directly relates the two entity vectors across multiple dimensions.
- **SE**(Bordes et al. 2011): *Structured Embeddings* (SE) (Bordes et al. 2011) obtains the embedding vectors of head entity and tail entity by corresponding left and right hand relation matrices respectively for the given pairwise relation.
- **SME(linear) or SME(bilinear)** (Bordes et al. 2012; 2014): In *Semantic Matching Energy* (SME), all the elements of a knowledge graph are represented into the relatively low-dimensional embedding vector space. If the relation-dependent embedding function is simply a linear layer, it is called SME (linear). If the relation-dependent embedding function is using 3-modes tensors as the core weights, it is called SME (bilinear).
- **SePLi(local)**: SePLi(local) only utilizes the pairwise relations and disregards the long-range interactions.

Datasets

Four benchmark datasets are used for the performance evaluation. We divide each dataset into three distinct parts: $observed$, $unobserved\_entities$, and $unobserved\_relations$. For each dataset, some of the entities and all the facts that contain these entities are randomly collected and removed from each database to be as $unobserved\_entities$. Several pairwise relations and all the facts that denote these relations are also randomly collected and removed from each database to be as $unobserved\_relations$. The remaining facts which do not belong to $unobserved\_entities$ and $unobserved\_relations$ are collected as $observed$. Statistics of the datasets are shown in Table 1. In the optimization of entities (or relations) embeddings from $unobserved\_entities$ (or $unobserved\_relations$), it is noted that we have employed the entities and relations from $observed$, which is very useful for the incremental knowledge learning because we do not need to re-train all of the entities/relations when new (unobserved) entities or relations are added into a growing knowledge graph like NELL (Betteridge et al. 2009; Carlson et al. 2010) or KV (Dong et al. 2014). The detailed information of each dataset used in this paper is summarized below.

Kinship Alyawarra Kinship data (Denham 1973) records the family relations of Alyawarra, which includes more than 10 thousand relationships of 26 relation types among 104 tribe members.

UMLS Unified Medical Language System (UMLS) is an upper-level ontology dataset created by McCray (McCray 2003).

WordNet WordNet (Miller 1995) is an online lexical dataset. We obtain a subset of this data base, which has been used in (Socher et al. 2013).

Freebase Following (Socher et al. 2013), we obtained a subset of Freebase from the $People$ domain.

Parameter settings

In the experiments, hyper-parameters are set using grid search: (i) entity vectors, relation matrixes, and biases are all initialized by a uniform distribution on $[-0.001, 0.001]$; (ii) for datasets Kinship (Denham 1973) and UMLS (McCray 2003), the dimensionality of the embedding space is set to $d = 10$; for datasets WordNet (Miller 1995) and FreeBase (Bollacker et al. 2008), the dimensionality of the embedding space is set to $d = 100$; (iii) the weighting parameter of long-range loss is $\lambda_1 = 0.01$, and the regularization parameter is $\lambda_2 = 0.0001$; (iv) the length of each path is no longer than 4 ($L = 4$) and the maximum number of long-range interactions for each pairwise relation is $K = 10$; (v) mini-batched Limited-memory BFGS (L-BFGS) (Malouf 2002; Andrew and Gao 2007) is used for optimizing non-convex object function and the number of training iterations is set to 500.

Pairwise relations and long-range interactions

Given a pairwise relation, SePLi can discover several appropriate long-range interactions relevant to this pairwise relation. Table 2 shows some top weighted long-range interaction paths from different datasets, which are compatible with our rational knowledge.

Entity Retrieval

Our first goal is to retrieve the unknown facts. It is like to answer the question "Who is Alice's father?" by retrieving the *tail entity*, taking "Alice" as the *head entity* and "father" as the *pairwise relation*. For each test fact, the tail entity is removed and replaced by each of the entities in the dataset in turn. The results are ranked according to the confidence of each triplet.

We use first-rank, mean-rank, and MAP as the evaluation criteria. The first-rank indicates the average of the ranking position of the first correct answer of each test fact. The mean-rank is the average of the mean ranking position of all the correct answers of each test fact. Mean Average Precision (MAP) is defined as follows:

$$AP = \frac{1}{N_a} \sum_{r=1}^{N_e} prec(r)\delta(r) \tag{18}$$

where N_a is the number of the correct answers for the current test fact, N_e is the number of all the entities, $prec(r)$

Table 1: Statistics of datasets used in our experiments

Dataset	Kinship	UMLS	WordNet	Freebase
No. of entites/unobserved entities	104/9	135/12	38698/3868	75043/7503
No. of relations/unobserved relations	26/3	49/4	11/3	13/3
No. of $observed$ facts(train/test)	4006/4006	2407/2407	42121/42121	106832/106832
No. of $unobserved\_entities$ facts(train/test)	155/1451	134/1339	1848/17706	2899/36719
No. of $unobserved\_relations$ facts(train/test)	197/790	77/311	3575/14303	15104/60416

Table 2: The examples of top weighted long-range interaction paths relevant to a given pairwise relation over different datasets. The weights of each long-range path are shown in italic.

pairwise relation	top weighted long-range interaction paths
mother(Kinship)	father → son → mother (*0.258*); nephew → mother → mother (*0.0977*)
cause(UMLS)	isa → affects → affects (*0.100*); isa → affects → co-occurs_with (*0.100*)
_similar_to(WordNet)	_similar_to → has_instance → _type_of (*0.143*); _similar_to → _type_of → has_instance (*0.143*)
place_of_birth(Freebase)	parents → nationality (*0.100*); parents → place_of_death (*0.100*)

represents the precision of the r-th retrieved entity. $\delta(r) = 1$ if the r-th retrieved entity is a correct answer of the current query, and $\delta(r)$ = 0 otherwise. MAP is defined as the average AP of all the queries.

Table 3 and Table 4 shows the results of each compared model over Kinship and UMLS. From Table 3 and Table 4, we see that our proposed method achieves the best average performance.

Knowledge Prediction

In this section, we compare different models by predicting correct facts in the test data of WordNet and Freebase. The negative facts are generated by replacing the *head* or *tail* entity of each test fact with randomly selected entities. Following (Socher et al. 2013), we use accuracy as the criterion according to how many triplets are correctly predicted. For the large scale dataset such as WordNet dataset and the Freebase dataset, this criterion is more appropriate than the others to evaluate the performance of embedding.

The results are shown in Table 5. Our model achieves the best or a comparable performance. For $observed$ of these two datasets, models like NTN which have much more parameters always have a better performance because the entities and relations in the $observed$ are all well studied with more parameters. For $unobserved\_entities$ and $unobserved\_relations$ in which only a few facts are available, our model always has a better performance than the others due to the ability of SePLi to utilize the rich structures in knowledge graph.

Knowledge Embedding

In this section, we illustrate the interpretable embeddings by SePLi . "religion" is a relation with which the corresponding facts have the form '(entity, religion, religion_name)' in Freebase. Taking different religion_name's as different categories, and entities as instances, we select five categories with the most instances from Freebase such as "jew", "african american", "germans", "scottish people" and "poles". For each category, we randomly select no more than 100 instances from the Freebase. This results in a small dataset containing more than 400 entities. After learning the embedding of each entity using SePLi , we use t-SNE (Van der Maaten and Hinton 2008) to project the embeddings into a 2-dimensional space. The results are shown in Figure 3 (a). The data points with the same colors and shapes indicate that they come from the same category.

The family tree dataset is used in (Paccanaro and Hinton 2001) which contains 24 family members and 12 relations like "father","mother" and "husband". The embeddings of each entity are learnt using SePLi in a 2-dimensional space. The embeddings of entities are plotted in red points in Figure 3 (b). We show only two pairwise relations (i.e.,"*husband*" and "*brother*") between entities for the ease of visualization: the entities that have the pairwise relation "husband" are connected by blue chain lines and the entities having relation "brother" are connected by green lines. We find that the same pairwise relations are arranged nearly in parallel.

Complexity

The convergence of the objective function in our experiment indeed is the most time-consuming process. Figure 4 shows the time needed for training of the proposed SePLi in terms of the different embedding dimensions over Kinship dataset when the training iterations are set to 500. This experiment is conducted on a Intel Core i7-4790 CPU @ 3.6GHz. The time needed for find the long-rang interactions on Kinship is about 147 seconds and we find in total 3545 long-range interactions from 26 different observed pairwise relations.

Conclusions

We propose SePLi for knowledge graph embedding. In SePLi , both pairwise relations and long-range interactions between entities are encoded to boost the structured embeddings of the entities and relations. SePLi obtains almost the best performance in terms of entity retrieval and knowledge prediction.

Acknowledgement

This work is supported in part by National Basic Research Program of China (2012CB316400), NSFC (61472353),

Table 3: The comparisons of entity retrieval over Kinship(Denham 1973) in terms of first-rank, mean-rank, and MAP. The results shown in boldface are the best results.

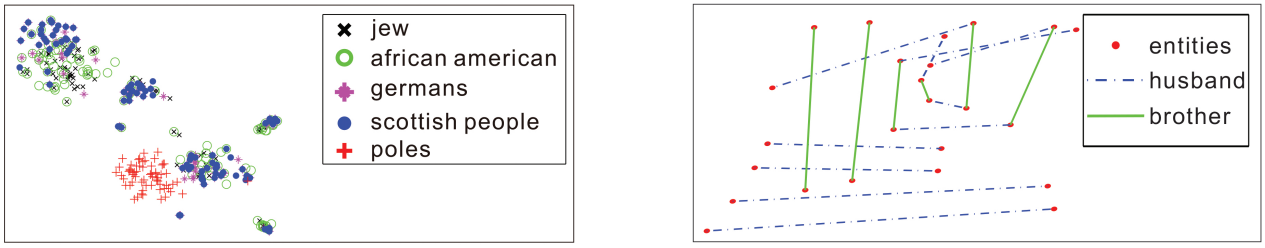
Method	*observed*			*out_entities*			*out_relations*		
	first-rank	mean-rank	MAP	first-rank	mean-rank	MAP	first-rank	mean-rank	MAP
SePLi	1.475	**8.032**	**0.797**	**4.045**	**9.358**	**0.556**	**1.856**	**6.044**	**0.672**
SePLi(local)	**1.376**	10.735	0.771	36.673	43.981	0.413	6.437	22.689	0.289
NTN	1.795	10.014	0.761	5.936	13.236	0.514	5.327	16.848	0.354
SE	3.392	11.505	0.467	14.770	22.986	0.225	2.737	9.342	0.482
SME-lin	13.244	37.596	0.169	35.330	54.505	0.130	12.459	32.716	0.163
SME-bil	24.149	37.310	0.206	25.911	36.860	0.179	5.899	15.033	0.288

Table 4: The comparisons of entity retrieval over UMLS (McCray 2003) in terms of first-rank, mean-rank, and MAP. The results shown in boldface are the best results.

Method	*observed*			*unobserved_entities*			*unobserved_relations*		
	first-rank	mean-rank	MAP	first-rank	mean-rank	MAP	first-rank	mean-rank	MAP
SePLi	**1.378**	**11.427**	**0.922**	**4.364**	**15.971**	**0.750**	**1.736**	**7.989**	**0.748**
SePLi(local)	1.403	11.982	0.910	5.419	19.633	0.613	4.180	19.736	0.491
NTN	5.741	24.175	0.511	20.253	37.572	0.306	2.637	18.576	0.490
SE	3.802	15.212	0.631	20.527	37.532	0.308	7.579	17.116	0.451
SME(linear)	3.900	17.021	0.545	11.388	28.213	0.358	3.370	25.557	0.426
SME(bilinear)	3.362	14.121	0.673	9.416	22.367	0.493	3.698	13.733	0.653

Table 5: Knowledge prediction task on WordNet (Miller 1995) and Freebase (Bollacker et al. 2008). Comparisons among six different models. The results shown in boldface are the best results. (*unob* is short for *unobserved*)

Method	WordNet			Freebase		
	observed	*unob_entities*	*unob_relations*	*observed*	*unob_entities*	*unob_relations*
SePLi	0.671	**0.683**	**0.703**	0.704	**0.722**	**0.675**
SePLi(local)	0.676	0.671	0.693	0.670	0.695	0.665
NTN	0.666	0.664	0.667	**0.717**	0.686	0.667
SE	0.667	0.666	0.672	0.667	0.667	0.667
SME(linear)	**0.682**	0.667	0.671	0.677	0.669	0.667
SME(bilinear)	0.667	0.667	0.667	0.676	0.667	0.673



(a) (b)

Figure 3: The illustration of embeddings of entities and relations learnt by SePLi in 2-dimensional space. (a) The data points with the same colors and shapes indicate that they come from the same category. We observe that the embeddings of entities learnt by SePLi have implicit margins according to their belonging categories (b) The embedding plot of each entity in toy family tree dataset in 2-dimensional space. Data points in red represent 24 entities. Here we observe that two pairwise relations (i.e., "husband" and "brother") between entities are arranged nearly in parallel respectively.

863 program (2012AA012505), China Knowledge Centre for Engineering Sciences and Technology and the Fundamental Research Funds for the Central Universities. ZZ is supported in part by US NSF (CCF-1017828) and Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis.

References

Andrew, G., and Gao, J. 2007. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, 33–40. ACM.

Bengio, Y.; Schwenk, H.; Senécal, J.-S.; Morin, F.; and Gau-

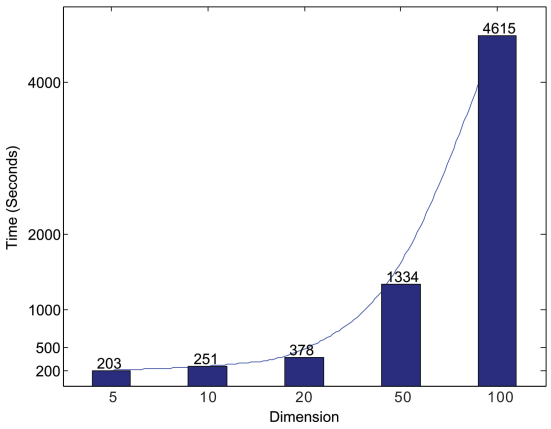


Figure 4: The complexity of the proposed SePLi in terms of the different embedding dimensions over Kinship dataset. The number of training iterations is set to 500

vain, J.-L. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*. Springer. 137–186.

Bengio Y, D. R.; P, V.; and C, J. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3(1137-1155).

Bengio, Y. 2008. Neural net language models. *Scholarpedia* 3(1):3881.

Betteridge, J.; Carlson, A.; Hong, S. A.; Hruschka Jr, E. R.; Law, E. L.; Mitchell, T. M.; and Wang, S. H. 2009. Toward never ending language learning. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, 1–2.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. ACM.

Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y.; et al. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *International Conference on Artificial Intelligence and Statistics*, 127–135.

Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 94(2):233–259.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka Jr, E. R.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, 3.

Denham, W. W. 1973. The detection of patterns in alyawarra nonverbal behavior.

Dong, X. L.; Murphy, K.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion.

Lao, N., and Cohen, W. W. 2010a. Fast query execution for retrieval models based on path-constrained random walks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 881–888. ACM.

Lao, N., and Cohen, W. W. 2010b. Relational retrieval using a combination of path-constrained random walks. *Machine learning* 81(1):53–67.

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 529–539. Association for Computational Linguistics.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning-Volume 20*, 1–7. Association for Computational Linguistics.

McCray, A. T. 2003. An upper-level ontology for the biomedical domain. *Comparative and Functional Genomics* 4(1):80–84.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Paccanaro, A., and Hinton, G. E. 2001. Learning distributed representations of concepts using linear relational embedding. *Knowledge and Data Engineering, IEEE Transactions on* 13(2):232–244.

Paccanaro, A. 2000. Learning distributed representations of concepts from relational data. *Knowledge and Data Engineering, IEEE Transactions on* 13:200.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, 926–934.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, 697–706. ACM.

Van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(2579-2605):85.

Xu, W., and Rudnicky, A. I. 2000. Can artificial neural networks learn language models? Computer Science Department.