On Completing Sparse Knowledge Base with Transitive Relation Embedding

Zili Zhou,[1,2] **Shaowu Liu,**[1] **Guandong Xu,**[*,1] **Wu Zhang**[2]
[1]Advanced Analytics Institute, University of Technology Sydney
[2]School of Computer Engineering and Science, Shanghai University
Zili.Zhou@student.uts.edu.au, Shaowu.Liu@uts.edu.au, Guandong.Xu@uts.edu.au, wzhang@shu.edu.cn

Abstract

Multi-relation embedding is a popular approach to knowledge base completion that learns embedding representations of entities and relations to compute the plausibility of missing triplet. The effectiveness of embedding approach depends on the sparsity of KB and falls for infrequent entities that only appeared a few times. This paper addresses this issue by proposing a new model exploiting the entity-independent transitive relation patterns, namely Transitive Relation Embedding (TRE). The TRE model alleviates the sparsity problem for predicting on infrequent entities while enjoys the generalisation power of embedding. Experiments on three public datasets against seven baselines showed the merits of TRE in terms of knowledge base completion accuracy as well as computational complexity.

Introduction

The last few years have seen a growing trend in constructing knowledge bases (KB) such as Freebase (Bollacker et al. 2008), WordNet (Fellbaum 2005), DBpedia (Auer et al. 2007) and Google Knowledge Graph (Singhal 2012). KB typically stores knowledge in form of the entity-relation-entity triplet, e.g., *"Sydney"-"is in"-"Australia"*. Collectively, a large number of triplets connect entities into a massive graph structure, which has a wide range of applications, such as recommender systems (Zhang et al. 2016), question answering (Yih et al. 2015) and information extraction (Yao and Van Durme 2014).

Despite its important role played in real-world applications, KB is often incomplete by its nature (Min et al. 2013) which is one of the main barriers to broader adoption. To address this issue, a considerable amount of literature on *knowledge base completion* has been published with an emphasis on embedding approach (Nickel, Tresp, and Kriegel 2011; Bordes et al. 2013; Wang et al. 2014; Yang et al. 2014; Lin et al. 2015b; Ji et al. 2015; Welbl, Riedel, and Bouchard 2016; Liu, Wu, and Yang 2017). The main idea of this approach is to learn low-dimensional representations of KB entities and relations which can then be used to infer missing triplets. Nevertheless, embedding approach looks at the global structures from the entire KB, thus its effectiveness depends on the sparsity of KB and falls for infrequent entities, i.e., reliable embedding representations can't be learnt for entities that only appeared a few times.

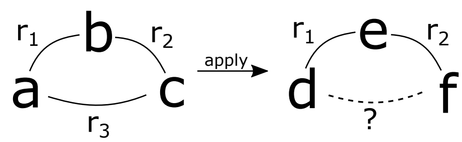

Figure 1: Entities a, b, c are connected through relations r_1, r_2, r_3. If these three relations, regardless of which entities are connected, appeared together frequently, then we may believe there is a pattern. The pattern is then applied to an incomplete triangle to predict the missing relation between entities d and f.

We argue that the local structure can be used to alleviate the sparsity problem by improving the completion of infrequent entities through frequent relation patterns. A typical local structure is the transitivity among relations as illustrated in Figure 1. The basic idea is that the missing relation between two entities could be inferred from a path connecting them. Although the idea is straightforward, it has several nice properties. Firstly, the relation patterns are independent of entities, thus makes it possible to predict missing relations for infrequent entities, which was a difficult task for embedding approach. Secondly, identifying relation patterns is less computational expensive (Tsourakakis 2008) comparing to the embedding approach because it does not require the learning of embedding representations for individual entities. Last but not least, relation patterns have great interpretability.

Nevertheless, the plain idea illustrated in Figure 1 has its flaws. Firstly, it favours frequent relation patterns thus unable to predict true relations that have never or infrequently appeared in relation patterns. Secondly, it learns strictly triangle relation pattern and does not generalise. To address these issues, we propose a new model called Transitive Relation Embedding (TRE). The idea behind TRE is to learn embedding representations for each relation from transitive relation patterns, which can be then used to predict missing relations. The main difference between TRE and traditional embedding models is that it does not require the learning of entity representations whilst be able to predict missing triplets involving infrequent entities. We summarise the main contributions of this work as follows:

- For the first time, the data sparsity problem in knowledge base completion is tackled by learning relation embeddings from transitive relation patterns.
- The new TRE model significantly improves completion performance on sparse knowledge bases comparing to state-of-the-art embedding models.
- We conducted extensive experiments on 3 datasets and 7 embedding models to evaluate the proposed TRE model in terms of accuracy and computational complexity.

The rest of the paper is organised as follows: Section "Related Work" introduces the basic concepts of knowledge base and existing embedding approach. Section "Transitive Relation Embedding" describes the proposed TRE model in detail. Section "Experiments" compares our TRE model with several state-of-the-art embedding models followed by a conclusion in Section "Conclusions".

Related Work

This section briefly summarises necessary background of knowledge base completion and relevant embedding-based and path-basesd models in literature.

Knowledge Base Completion

A knowledge base consists a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$. Knowledge facts are stored as collection of triplets $\mathcal{D} = \{(h, r, t)\}$ where $h \in \mathcal{E}$ is the head entity, $t \in \mathcal{E}$ is the tail entity and $r \in \mathcal{R}$ is the relation connects the two entities. For example, "Sydney (h), is in (r), Australia (t)". The triplets set $\mathcal{D}$ is often incomplete, which calls for knowledge base completion to infer the missing triplets based on the given set of knowledge fact triplets.

Embedding-based Knowledge Base Completion

Multi-relational embedding is a type of knowledge base completion approach that embeds entities and relations in the same space $\mathbf{R}^d$. To be specific, each entity $h \in \mathcal{E}$ is represented by a vector $\mathbf{h}$ and each relation $r \in \mathcal{R}$ is represented by a vector $\mathbf{r}$. The core idea of multi-relational embedding (Bordes et al. 2013) is to learn these vectors such that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when the triplet (h, r, t) holds. For this particular model, the scoring function can be defined as:

$$f_r(h, t) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{1/2} \tag{1}$$

The latent vectors should be learnt to maximise $f_r(e_1, e_2)$ if the triplet (h, r, t) is true. There exist other variants of multi-relational embedding models (Wang et al. 2014; Lin et al. 2015b; Ji et al. 2015; Yang et al. 2014; Welbl, Riedel, and Bouchard 2016), nevertheless, latent vectors of entities and relations will be learnt.

Relation-based Knowledge Base Completion

Literature on Relation-based knowledge base completion is limited. A recent work done by Yoon et al. (Yoon et al. 2016) attempted to preserve the logical properties among relations by adding a role-specific mapping matrix for entities. Another work from Lin et al. (Lin et al. 2015a) considers path between entities, however, only in the form of line segments. Nevertheless, learning latent vectors for entities is required for all of these methods. Other works (Tang et al. 2015; Grover and Leskovec 2016; Dong, Chawla, and Swami 2017) exist to analyse the path or relation embeddings in graph structures, however, not targeting the knowledge base completion.

Limitations of Current Models

As shown in Eq. 1, existing multi-relational embedding models require the learning of both entity and relation representations. It can be difficult to learn entity representations when an entity has never appeared in the training set or only appeared a few times. This limitation makes it impractical to infer related missing triplets. Besides, learning entity representations can be costly due to the large number of entities in the KB that further limits large scale applications (Zhang et al. 2016).

Transitive Relation Embedding

To solve the sparse KB completion inaccuracy problem, we proposed an embedding based relation inference model. The proposed model focuses on following issues.

- Instead of training individual knowledge fact, proposed model extracts knowledge information by using co-occurrence statistics of relations. We use these statistics as the input of embedding model to learn inference rules. We highly improve the accuracy in prediction, especially on sparse KG.
- Our proposed model focuses on explicit relation inference. We use the transitivity of Knowledge Base relations (from the relations A-B and B-C we can infer the relation between A and C) to extract inference rules, this makes the result of our model highly interpretable.
- In the proposed model, we only learn the embedding for relations, which makes the number of the parameters extremely small. This makes the training process efficient.

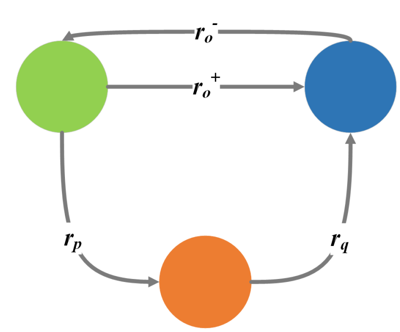

Figure 2: Triangle Pattern

Triangle Pattern

Inspired by previous work on triangle pattern (Tsourakakis 2008), we observed the triangle structure in our dataset and found that triangle pattern also existed in Knowledge Base. We extract 115,939 triangle patterns from FB15K, 1,068 triangle patterns from WN18 and 46,327 triangle patterns from DBP.

The triangle patterns in social network user relation graph and web page reference link graph can be used for social community detection and web page semantic structure discovery. This makes us assume that Knowledge Base triangle pattern can be helpful for Knowledge Base hidden relation discovery, and go a step further, Knowledge Base automatic completion. However, Knowledge Base is a multi-relational graph, different than social network user relation graph and web page reference link graph. With entities as nodes, the edges in Knowledge Base represent multiple types of relations between entities. Thus, we focus on relation inference based on the triangle pattern of Knowledge Base.

To formulate the triangle patterns, as shown in Fig. 2, we define a restricted triangle structure with three nodes $a(green), b(orange), c(blue)$. We represent the relation from a to b as r_p, the relation from b to c as r_q. If the relation between a and c is from a to c, we represent it as r_o^+, otherwise, we use r_o^- for the relation from c to a. In each restricted triangle structure, either r_o^+ or r_o^- occurs between a and c. In each triangle structure, if one relation is missed, we can use the other two relations to predicted the missed one.

$$
\begin{aligned}
Confidence(r_o^+|r_p, r_q) &= \frac{Freqency(r_p, r_q, r_o^+)}{Freqency(r_p, r_q)} \\
Confidence(r_p|r_o^+, r_q) &= \frac{Freqency(r_p, r_q, r_o^+)}{Freqency(r_o^+, r_q)} \\
Confidence(r_q|r_o^+, r_p) &= \frac{Freqency(r_p, r_q, r_o^+)}{Freqency(r_o^+, r_p)} \\
Confidence(r_o^-|r_p, r_q) &= \frac{Freqency(r_p, r_q, r_o^-)}{Freqency(r_p, r_q)} \\
Confidence(r_p|r_o^-, r_q) &= \frac{Freqency(r_p, r_q, r_o^-)}{Freqency(r_o^-, r_p)} \\
Confidence(r_q|r_o^-, r_p) &= \frac{Freqency(r_p, r_q, r_o^-)}{Freqency(r_o^-, r_q)}
\end{aligned} \tag{2}
$$

If we found a complete triangle structure, "a-r1-b-r2-c, a-r3-c", we call it triangle pattern. Besides, if we found a triangle consisting of three entities with only two edges, such as "a-r1-b-r2-c", we call it potential triangle pattern. We use existed triangle pattern in KB for model training, and we predict new triangle pattern based on existed potential triangle pattern in KG.

Collecting all the triangle patterns in Knowledge Base, we count the co-occurrence frequency of potential triangle patterns as $Freqency(r_1, r_2)$, r_1, r_2 can be replaced by picking two from $r_p, r_q, r_o^+/r_o^-$. We also count the co-occurrence frequency of $r1$, $r2$ and $r3$ in triangle patterns as $Freqency(r_1, r_2, r_3)$, r_1, r_2, r_3 can be replaced by $r_p, r_q, r_o^+/r_o^-$, the order can be changed. We can conclude a candidate inference rule, "$n_1 - r_1 - n_2 - r_2 - n_3 => n_1 - r_3 - n_3$". In textual expression, "If r_1 and r_2 occur on a specific role in the triangle pattern consists of three nodes n_1, n_2, n_3, we can measure how likely r_3 occurs by a confidence". As shown in Eq. 2, based on occurrence dependency between relations, the confidence can be computed as $Confidence(r_3|r_1, r_2) = Frequency(r_1, r_2, r_3)/Frequency(r_1, r_2)$. The confidence of each candidate inference rule is used in embedding learning process to indicate how likely the rule is tenable.

Transitive Relation Embedding

To measure the probability of candidate inference rules based on the relation transitive inference, we propose the Transitive Relation Embedding model.

If we observe $r1$ and $r2$ in a triangle pattern, then we can predict the occurrence probability of $r3$ and represent it as $P(r3|r1, r2)$. We have the following probabilities needed to be predicted in our model, $P(r_o^+|r_p, r_q)$, $P(r_o^-|r_p, r_q)$, $P(r_p|r_o^+, r_q)$, $P(r_p|r_o^-, r_q)$, $P(r_q|r_o^+, r_p)$ and $P(r_q|r_o^-, r_p)$.

$$
\begin{aligned}
\overrightarrow{V_{r_p,r_q}} &= M_1\overrightarrow{r_p} + M_2\overrightarrow{r_q}, \\
\overrightarrow{U_{r_o}^+} &= M_3^+\overrightarrow{r_o}, \\
\overrightarrow{U_{r_o}^-} &= M_3^-\overrightarrow{r_o}
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
P(r_o^+|r_p, r_q) &= \frac{exp(\overrightarrow{U_{r_o}^+}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R [exp(\overrightarrow{U_{r_k}^+}^T\overrightarrow{V_{r_p,r_q}}) + exp(\overrightarrow{U_{r_k}^-}^T\overrightarrow{V_{r_p,r_q}})]} \\
P(r_o^-|r_p, r_q) &= \frac{exp(\overrightarrow{U_{r_o}^-}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R [exp(\overrightarrow{U_{r_k}^+}^T\overrightarrow{V_{r_p,r_q}}) + exp(\overrightarrow{U_{r_k}^-}^T\overrightarrow{V_{r_p,r_q}})]} \\
P(r_p|r_o^+, r_q) &= \frac{exp(\overrightarrow{U_{r_o}^+}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U_{r_o}^+}^T\overrightarrow{V_{r_k,r_q}})} \\
P(r_p|r_o^-, r_q) &= \frac{exp(\overrightarrow{U_{r_o}^-}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U_{r_o}^-}^T\overrightarrow{V_{r_k,r_q}})} \\
P(r_q|r_o^+, r_p) &= \frac{exp(\overrightarrow{U_{r_o}^+}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U_{r_o}^+}^T\overrightarrow{V_{r_p,r_k}})} \\
P(r_q|r_o^-, r_p) &= \frac{exp(\overrightarrow{U_{r_o}^-}^T\overrightarrow{V_{r_p,r_q}})}{\sum_{r_k}^R exp(\overrightarrow{U_{r_o}^-}^T\overrightarrow{V_{r_p,r_k}})}
\end{aligned} \tag{4}
$$

We learn a k-dimensional vector embedding for each relation in Knowledge Base, each relation is represented as a point in a k-dimensional space, all relations share the same space. But in each triangle pattern, each relation has different role, r_p, r_q, r_o^+ or r_o^-. As shown in Eq. 3, for two relations occurs on positions of r_p and r_q, we map their embeddings into one joint role-specific space point as $\overrightarrow{V_{r_p,r_q}}$ by using role-specific matrices M_1 and M_2. We also map the embedding of the relation in position r_o^+ / r_o^- into role-specific space as $\overrightarrow{U_{r_o}^+}$ / $\overrightarrow{U_{r_o}^-}$. We use M_3^+ as role-specific matrix for r_o^+, and use M_3^- for r_o^-.

As shown in Eq. 4, we can compute the probabilities of relation occurrence based on mapped vector space points $\overrightarrow{V_{r_p,r_q}}$ and $\overrightarrow{U_{r_o}^{+}}$ / $\overrightarrow{U_{r_o}^{-}}$. We give the assumption that if the relation inference rule is likely to be true, the probability we predict in our model should be high, which means interaction of $\overrightarrow{V_{r_p,r_q}}$ and $\overrightarrow{U_{r_o}^{+}}$ / $\overrightarrow{U_{r_o}^{-}}$ should be high. Our probability equation meets the restrictions in Eq. 5.

$$
\begin{aligned}
&\sum_{r_k}^{R}[P(r_k^{+}|r_p,r_q)+P(r_k^{-}|r_p,r_q)]=1,\\
&\sum_{r_k}^{R}P(r_k|r_o^{+},r_q)=1,\sum_{r_k}^{R}P(r_k|r_o^{-},r_q)=1,\\
&\sum_{r_k}^{R}P(r_k|r_o^{+},r_p)=1,\sum_{r_k}^{R}P(r_k|r_o^{-},r_p)=1.
\end{aligned}
\tag{5}
$$

An important reason to use embedding model is that the result of embedding model is one vector for each relation, it can not only compute the probabilities for relation triples occurred in training data, it can also generalize to the relation triples never occurred in training data but need to be predicted in test data. For example, (r_1, r_3, r_5^{+}), (r_1, r_4, r_5^{+}) and (r_2, r_3, r_6^{+}) occurred in training data, r_2, r_4 and r_6 never occurred in the same triangle pattern, we can still compute probabilities for inference rule consists of r_2, r_4 and r_6, because we learn the transitive inference information of r_2, r_4 and r_6 and represent it by embedding vector.

Training

With the above definitions, we can get a training likelihood equation for our detail embedding model. We define two triple sets, $\mathcal{S}^{+}$ and $\mathcal{S}^{-}$, $\mathcal{S}^{+}$ consists of relation triples with forward direction third relation r_o^{+} (from a to c). Conversely, $\mathcal{S}^{-}$ consists of relation triples with backward direction third relation r_o^{-} (from c to a). As shown in Eq. 6, in the likelihood equation, we use KL-divergence for distributions of confidence and predicted probability. By maximizing the likelihood equation, the relation inference rule with higher confidence have more chance to result in high probability.

$$
\begin{aligned}
\mathcal{L}=&\sum_{r_p,r_q,r_o^{+}}^{\mathcal{S}^{+}}\{Confidence(r_o^{+}|r_p,r_q)log[P(r_o^{+}|r_p,r_q)]\\
&\qquad+Confidence(r_p|r_o^{+},r_q)log[P(r_p|r_o^{+},r_q)]\\
&\qquad+Confidence(r_q|r_o^{+},r_p)log[P(r_q|r_o^{+},r_p)\}\\
&+\sum_{r_p,r_q,r_o^{-}}^{\mathcal{S}^{-}}\{Confidence(r_o^{-}|r_p,r_q)log[P(r_o^{-}|r_p,r_q)]\\
&\qquad+Confidence(r_p|r_o^{-},r_q)log[P(r_p|r_o^{-},r_q)]\\
&\qquad+Confidence(r_q|r_o^{-},r_p)log[P(r_q|r_o^{-},r_p)]\}
\end{aligned}
\tag{6}
$$

Joint Prediction Strategy

We also observed a limitation of the proposed model. It can't predict links between a pair of entities if there is no existing potential triangle pattern in training data between them. We use a strategy to combine the prediction results of the proposed model and baseline models including TransE, TransH, TransR, RESCAL, TransD, DistMult, ComplEx. The final result of the combined model is improved.

In entity link prediction and relation link prediction tasks, we target to predict how likely the given entity-relation-entity triple is true or false. For each triple (a, r_o, c), a and c are entities, r is one relation from a to c, we detect all the potential triangle pattern between entity pair a and c. Specifically, we find all the combination of r_p, b, r_q which can link a and c, $a-r_p-b-r_q-c$, r_p and r_q are relations, b is an entity. If there is any potential triangle pattern between entity pair a and c, we can compute the probability based on the triangle pattern inference embedding vectors of r_p, r_q and r_o, we predict the triple authenticity with computed probability, we can achieve more accurate prediction than baselines because of the advantages of triangle pattern inference embedding. Otherwise, if there is no potential triangle pattern between a and c, we can't compute the probability. In this condition, we predict a new triple based on score function value of baselines. In brief, our proposed method uses the same prediction result for entity pairs with no potential triangle pattern, it uses embedding based triangle pattern inference to achieve better prediction result for entity pairs with the potential triangle pattern.

Advantages of Proposed Model

Outperform on sparse KG. In sparse KG, some entities occur infrequently, traditional embedding models can't do accurate prediction for these entities, while our proposed method focuses on relation inference, as long as there is valid triangle pattern between two entities, we can give accurate prediction, the entity occurrence frequency is irrelevant.

High interpretability. Our proposed model can achieve high interpretability because the probability of each candidate relation inference rule can be explicitly computed. Given two entities a, c, and a potential triangle pattern $a - r_1 - b - r_2 - c$ between them, we can explicitly compute the probability of relation r_3 between a and c, $P(r_3|r_1, r_2)$. We can conclude an interpretable rule, "If r_1 between a and b, r_2 between b and c, then the probability of there is r_3 between a, c is $P(r_3|r_1, r_2)$". In Tab. 1, we list some interpretable relation inference rule example computed by our proposed model on dataset FB15K.

Efficient parameter learning. Our proposed model only needs to learn embedding for relations in order to simplify the parameters. With less number of parameters, we can achieve the training efficiency with the proposed model. We list the space complexity, time complexity and running time for convergence of different methods in Tab. 2. n is entity number, m is relation number, d is the entity embedding dimension, k is relation embedding dimension. We can see that TRE method has an advantage on space complexity. And on time complexity, TRE can converge in fewer iter-

Table 1: Relation Inference Example

Potential triangle pattern	Inferred relation	Probability
r_p: film_release_region, r_q: languages_spoken	r_o^+: language	0.9916
r_p: actor, r_q: languages	r_o^+: language	0.9578
r_p: film_release_region, r_q: currency	r_o^+: currency	0.9433
r_p: spouse, r_q: place_lived_location	r_o^+: location_of_ceremony	0.9969
r_p: sibling, r_o^+: ethnicity	r_q: ethnicity	0.9664
r_p: computer_videogame/sequel, r_q: computer_videogame/developer	r_o^-: games_developed	0.9987

Table 2: Space and Time Complexity

	Space complexity	Time complexity	Approx. Exec. Time (s)
TransE	$\mathcal{O}(nd+md)$	$\mathcal{O}(d)$	400
TransH	$\mathcal{O}(nd+md)$	$\mathcal{O}(d)$	400
TransR	$\mathcal{O}(nd+mdk)$	$\mathcal{O}(dk)$	9,000
RESCAL	$\mathcal{O}(nd+md^2)$	$\mathcal{O}(d^2)$	6,200
TransD	$\mathcal{O}(nd+mk)$	$\mathcal{O}(max(d,k))$	400
DistMult	$\mathcal{O}(nd+md)$	$\mathcal{O}(d)$	1,400
ComplEx	$\mathcal{O}(nd+md)$	$\mathcal{O}(d)$	1,800
TRE	$\mathcal{O}(mk)$	$\mathcal{O}(k^2)$	360

ations than other methods, we find that the execution time of TRE is shorter than others.

Why we focus on triangle pattern rule inference? By observing KB structure, we find that transitivity existed in most of existed KG, and it is a key factor for KB completion because it can reliably infer new relations between a pair of entities. To model the KB transitivity, we need a simple but reasonable representation, triangle pattern. We use triangle pattern to represent the transitive relation inference, and conclude interpretable relation inference rule based on triangle pattern.

Why we use relation transitivity statistics? Compare with baselines, especially TransX KB completion methods, we use a totally different training framework to solve KB sparsity problem. We focus on KB relations only and abandon entity embedding learning, we also use relation transitivity statistics, The occurrence of triangle pattern, as input of our learning process. As long as the relation triangle pattern related to an entity is proved by large number of samples in entire KG, the infrequent occurrence of this entity doesn't influence the KB completion accuracy.

Why we use embedding model? We use embedding model in this paper for learning result generalization. If a triangle pattern in the testing set has never occurred in training data, we can't determine whether it is true with traditional methods, such as rule-based model, however, the embedding model generalizes the learning result. By embedding model, we can learn embedding for the relations consisting the triangle pattern, which enables us to determine the probability of that the triangle pattern is tenable. Through the generalization of embedding model, as long as the relations occurred in training data, we can do prediction for triangle patterns which has ever occurred in training data.

Experiments

We use several entity embedding required knowledge base embedding models, which are popular KB Embedding models used in previous works, as our baselines including TransE(Bordes et al. 2013), TransH(Wang et al. 2014), TransR(Lin et al. 2015b), RESCAL(Nickel, Tresp, and Kriegel 2011), TransD(Ji et al. 2015), DistMult(Yang et al. 2014), and ComplEx(Welbl, Riedel, and Bouchard 2016). We test the performance of these methods on several widely used KB datasets, including FB15K and WN18. We also construct an extremely sparse dataset by extracting subset from entire DBpedia project, we call this dataset "DBP" in experiment. The sizes of datasets are listed in Tab. 3, we also list average and median time of entities occur in training dataset in last two column of table. We can see that DBP is far more sparse than other two datasets.

Entity Link Prediction

To compare the performance of proposed model with baselines direct, we do an experiments for entity link prediction under the framework of previous works, predicting tail entity $(h, r, ?)$ and predicting head entity $(?, r, t)$. Given a pair (h, r) or (r, t), h is head entity of triple, r is relation, t is tail entity, our task is to predict the missing part of triple, to predict t for (h, r) and to predict h for (r, t). For tail entity prediction, $(h, r, ?)$, we fill tail entity with any entity e in KG, and rank the entities with the probability that the triple (h, r, e) is true, the triple with the higher rank is more likely to be true. Similarly, for head entity prediction, $(?, r, t)$, we rank the triples (e, r, t) and determine which triples are more likely to be true.

For triple ranking, we need to compute the score of each triple. In baselines, we compute the score of triple with the score function of baselines, we rank the triples in ascending order, low score indicates high rank. In the proposed model, we first detect all potential triangle pattern between pair $(h, e)/(e, t)$, then we sum up all the probability that r occurs between $(h, e)/(e, t)$ as score of triple, we rank the triples in descending order, high score indicates high rank. With ranking result, we use three evaluation methods, Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hit@10. The target of prediction task is to achieve low MR, high MRR and high Hit@10.

In Tab. 4, we show the entity link prediction results on FB15K dataset. In Tab. 5, we show the entity link prediction results on WN18 dataset. The left part of tables contain re-

Table 3: Dataset Size

	Entity count	Relation count	Triple count	Triangle pattern count	Avg. entity occurrence	Med. entity occurrence
FB15K	$14,951$	$1,346$	$483,142$	$115,939$	64.63	41
WN18	$40,943$	18	$141,442$	$1,068$	6.91	4
DBP	$376,941$	566	$432,760$	$46,327$	2.27	1

Table 4: Result of FB15K Entity Prediction

	$(h, r, ?)$						$(?, r, t)$					
	Baseline			Baseline+TRE			Baseline			Baseline+TRE		
	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10
TransE	.2371	222.22	.4355	**.5444**	**60.17**	**.7974**	.1786	346.65	.3536	**.4789**	**83.51**	**.7184**
TransH	.2317	234.33	.4222	**.5442**	**60.54**	**.7965**	.1733	364.00	.3428	**.4784**	**85.57**	**.7176**
TransR	.2428	209.27	.4422	**.5451**	**55.74**	**.7986**	.1822	347.35	.3627	**.4794**	**82.67**	**.7201**
RESCAL	.1519	523.31	.2615	**.5392**	**114.90**	**.7825**	.1015	806.08	.1879	**.4741**	**163.55**	**.7054**
TransD	.2307	244.57	.4193	**.5441**	**62.56**	**.7967**	.1735	375.40	.3404	**.4785**	**86.25**	**.7173**
DistMult	.1904	231.57	.3603	**.5425**	**58.32**	**.7909**	.1370	334.54	.2852	**.4768**	**84.14**	**.7130**
ComplEx	.2430	207.30	.4671	**.5467***	**52.05***	**.8043***	.1871	296.30	.3899	**.4811***	**71.82***	**.7257***

Table 5: Result of WN18 Entity Prediction

	$(h, r, ?)$						$(?, r, t)$					
	Baseline			Baseline+TRE			Baseline			Baseline+TRE		
	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10
TransE	.2424	625.15	.4564	**.3649**	**582.83**	**.5422**	.2184	614.51	.4202	**.3400**	**555.27**	**.5072**
TransH	.0378	2974.43	.073	**.1922**	**2572.43**	**.229**	.0400	2948.18	.0716	**.1933**	**2500.31**	**.2262**
TransR	.2776	469.16	.5268	**.3924**	**443.12***	**.5954**	.2537	482.39	.5114	**.3698**	**441.37***	**.581**
RESCAL	.0408	7172.78	.0722	**.1962**	**6167.40**	**.2278**	.0584	6677.36	.0984	**.2101**	**5777.30**	**.252**
TransD	.2208	769.21	.3986	**.3471**	**706.03**	**.4946**	.2042	850.86	.3752	**.3323**	**747.70**	**.4782**
DistMult	.3226	761.08	.5866	**.4333**	**703.62**	**.655**	.2966	767.61	.5602	**.4091**	**709.83**	**.634**
ComplEx	.5627	819.72	.8012	**.6351***	**719.08**	**.8328***	.5399	839.39	.781	**.6177***	**730.66**	**.818***

Table 6: Result of FB15K, WN18 and DBP Relation Prediction $(h, ?, t)$

	FB15K						WN18						DBP					
	Baseline			Baseline+TRE			Baseline			Baseline+TRE			Baseline			Baseline+TRE		
	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10	MRR	MR	Hit@10
TransE	.5049	63.99	.7256	**.6730**	**19.54**	**.8882**	.5833	3.61	.9398	**.6100**	**3.34**	**.9466**	.0121	258.56	.0150	**.1827***	**192.50***	**.2265**
TransH	.5281	55.47	.7494	**.6780***	**20.37**	**.8891**	.1568	10.68	.3764	**.2372**	**9.52**	**.4666**	.0124	273.02	.0169	**.1824**	**201.92**	**.2263**
TransR	.4169	194.95	.5998	**.6550**	**37.72**	**.8638**	.4582	4.33	**.9628**	**.4976**	**4.04**	.9618	.0119	267.27	.0159	**.1821**	**198.93**	**.2260**
RESCAL	.4314	11.47	.7969	**.6629**	**8.47***	**.9035***	.6417	3.52	.9046	**.6486**	**3.43**	**.9136**	.0112	284.79	.0144	**.1820**	**210.73**	**.2254**
TransD	.5115	67.04	.7304	**.6743**	**22.19**	**.8856**	.6891	3.52	.8404	**.7060**	**3.23**	**.8716**	.0120	258.92	.0165	**.1824**	**192.57**	**.2270**
DistMult	.0855	44.77	.2120	**.5959**	**15.68**	**.7972**	**.7669**	**1.84**	**.9764***	.7489	2.04	.9728	.0124	275.04	.0194	**.1823**	**204.69**	**.2278***
ComplEx	.4442	19.89	.8467	**.6591**	**11.04**	**.9010**	**.9493***	**1.48***	**.9752**	.9016	1.75	.9716	.0117	270.99	.0162	**.1817**	**202.58**	**.2256**

sult of tail entity prediction, $(h, r, ?)$, the right part of tables contain result of head entity prediction, $(?, r, t)$. By comparing prediction result of baselines and joint prediction result of baselines and TRE, we find that joint prediction result outperform the baselines on both two datasets. For each baseline, we compare the results for baseline and TRE+baseline, the bold font results means the better one between baseline and TRE+baseline, the results with star represent the best result among all 14 methods including 7 baseline methods and 7 TRE+baseline methods. As we can see, in entity link prediction, joint prediction (TRE+baseline) result outperforms the baselines on both two datasets.

Relation Link Prediction

We also test relation link prediction, predicting (h, ?, t), to show the proposed model is capable to predict new relations between entities. Given an entity pair (h, t), the task is to predict the relation r between h and t. For each pair (h, t), we fill triple (h, ?, t) with any relation r, the score of triple (h, r, t) is computed as same as in entity link prediction. We rank the triples to determine which triples are more likely to be true, we also use MR, MRR and Hit@10 for result evaluation.

We can see on FB15K and DBP, the TRE+baseline methods outperform the baseline methods, however, on WN18, we observe that some result baseline methods are slightly better than TRE+baseline methods, this is because there are only 18 relations in WN18 dataset, we can extract a limited number of triangle pattern for training. As shown in Tab. 3, the entity size and triple size of WN18 is close to other two datasets, but the relation size is extremely small, which causes the triangle pattern extracted is extremely less than other two datasets. The meaning of bold font and star is as same as in entity link prediction task.

Table 7: Result of Sparse FB15K Relation Prediction

	MRR	MR	Hit@10
TransE	.3632	204.63	.5295
TransH	.3867	174.89	.5810
TransR	.3322	287.74	.4668
RESCAL	.3713	27.88	.6438
TransD	.3663	199.49	.2659
DistMult	.0414	78.20	.1128
ComplEx	.4488	42.01	.8088
TRE	**.6429**	**14.84**	**.8723**

Accurate Prediction on Extremely Sparse KG

We do the relation link prediction on this sparse dataset DBP. The result shows that the proposed model has a clear advantage than other baselines when they deal with sparse dataset. From Tab. 3, we can see that the average occurrence of each entity in DBP is far less than other two datasets, which makes the dataset DBP extremely sparse. We test the baseline methods and TRE+baseline methods, the result shows that our proposed model has large advantage, the baseline methods perform poorly, however, the predictions of proposed model are still accurate.

For further testify our assumption, we also extract a sparse subset of FB15K to show the capability of the proposed method on sparse KB data through leveraging transitive relation inference. We extract a sparse subset of FB15K by extracting the triples meeting following conditions, at least one entity occurred less than 5 times in training dataset, at least one triangle pattern existed between two entities. Comparing result in Tab. 7 with the FB15K result in Tab. 6, we can find that the results of baseline methods fall largely on at least one evaluations, however, the TRE results is barely influenced by sparsity.

Conclusions

In This paper, we proposed TRE, a new embedding model using the transitivity of Knowledge Base relations to efficiently solve KB sparsity problem. To take advantages of Knowledge Base relation transitivity, we extract relation triangle pattern from large-scale Knowledge Bases. We measure the reliability of Knowledge Base relation inference rule with confidences of relation triangle patterns, which are used as input to train TRE model. We evaluate our proposed model with two tasks, entity prediction and relation prediction, the proposed model outperforms baselines. We specially test our model on sparse dataset, the advantage of proposed model is greater. Because of using relation triangle pattern statistics as training data, the entity occurrence frequency is irrelevant, proposed model can achieve good result on sparse data. The possibility of each given triangle pattern relation inference rule can be explicitly computed, which makes the prediction result of TRE interpretable. By learning embedding of relations only instead of both entities and relations, TRE can achieve efficient training. The above advantages of TRE are evaluated in our experiments.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer. 722–735.

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.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *International Conference on Neural Information Processing Systems*, 2787–2795.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 135–144. ACM.

Fellbaum, C. 2005. Wordnet and wordnets.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Acm Sigkdd International*

Conference on Knowledge Discovery & Data Mining, 855–864.

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, 687–696.

Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; and Liu, S. 2015a. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 705–714.

Lin, Y.; Liu, Z.; Zhu, X.; Zhu, X.; and Zhu, X. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2181–2187.

Liu, H.; Wu, Y.; and Yang, Y. 2017. Analogical inference for multi-relational embeddings. In *International Conference on Machine Learning*, 2168–2178.

Min, B.; Grishman, R.; Wan, L.; Wang, C.; and Gondek, D. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 777–782.

Nickel, M.; Tresp, V.; and Kriegel, H. P. 2011. A three-way model for collective learning on multi-relational data. In *International Conference on International Conference on Machine Learning*, 809–816.

Singhal, A. 2012. Introducing the knowledge graph: things, not strings. *Official google blog* 5.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line:large-scale information network embedding. 2(2):1067–1077.

Tsourakakis, C. E. 2008. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *Eighth IEEE International Conference on Data Mining*, 608–617.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.

Welbl, J.; Riedel, S.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2071–2080.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases.

Yao, X., and Van Durme, B. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 956–966.

Yih, S. W.-t.; Chang, M.-W.; He, X.; and Gao, J. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base.

Yoon, H.-G.; Song, H.-J.; Park, S.-B.; and Park, S.-Y. 2016. A translation-based knowledge graph embedding preserving logical property of relations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 907–916.

Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W.-Y. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 353–362. ACM.