



International Conference on Computational Science, ICCS 2017, 12-14 June 2017,
Zurich, Switzerland
Learning Entity and Relation Embeddings for Knowledge
Resolution

Hailun Lin, Yong Liu*, Weiping Wang, Yinliang Yue, and Zheng Lin

Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
linhailun@iie.ac.cn

Abstract

Knowledge resolution is the task of clustering knowledge mentions, e.g., entity and relation mentions into several disjoint groups with each group representing a unique entity or relation. Such resolution is a central step in constructing high-quality knowledge graph from unstructured text. Previous research has tackled this problem by making use of various textual and structural features from a semantic dictionary or a knowledge graph. This may lead to poor performance on knowledge mentions with poor or not well-known contexts. In addition, it is also limited by the coverage of the semantic dictionary or knowledge graph. In this work, we propose ETransR, a method which automatically learns entity and relation feature representations in continuous vector spaces, in order to measure the semantic relatedness of knowledge mentions for knowledge resolution. Experimental results on two benchmark datasets show that our proposed method delivers significant improvements compared with the state-of-the-art baselines on the task of knowledge resolution.

© 2017 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: knowledge graph, knowledge resolution, knowledge representation, entity embedding, relation embedding

1 Introduction

Access to an organized knowledge graph is critical for many real-world tasks, such as query suggestion and question answering. Most real-world information is unstructured, interconnected, noisy, and often expressed in the form of text. This inspires constructing an organized knowledge graph from the large volume of noisy text data. A large number of knowledge graphs have been constructed, such as Freebase [1], Knowledge Vault [8], YAGO [9], Probase [18]. An important component in constructing knowledge graphs is knowledge resolution. Given the knowledge mentions (e.g., entity mentions and relation mentions) in unstructured text data, the goal of knowledge resolution is to cluster knowledge mentions into disjoint groups with each group representing a unique knowledge.

*Corresponding author: Yong Liu (liyong@iie.ac.cn)

The knowledge resolution task is challenging due to the fact that many knowledge mentions are ambiguous: the same mention can refer to various different real world entities or relations when they appear in different contexts, and many knowledge has various mention forms. Knowledge resolution plays a critical role in high-quality knowledge graph construction. When knowledge extracted from text data is ready to be inserted into the knowledge graph, it is necessary to know which real world knowledge this piece of information should be associated with. If a wrong decision is made here, the knowledge graph will not only lose some information, but also introduce errors.

Traditional approach to knowledge resolution usually makes use of various textual and structural features from a semantic dictionary or a knowledge graph (e.g., [5], [16]), which may lead to poor performance on knowledge mentions with poor not well-known contexts. Moreover, its performance is also limited by the coverage of the semantic dictionary or knowledge graph. Recently, a promising approach for the task is embedding knowledge into a continuous vector space to learn feature representations, in order to measure the semantic relatedness of knowledge mentions for knowledge resolution. Following this approach, many methods have been explored, which will be introduced in detail in Section 2. Among these methods, some notable works, including TransE [3], TransH [17], TransR [11] are effective and efficient. TransE represents entities as points and relations as translations from head entities to tail entities in a vector space. TransH models a relation as a hyperplane together with a translation operation on it. TransR models entity and relation embeddings in separate vector spaces, which are bridged by relation-specific matrices. These methods mainly focus on modeling single knowledge in continuous space, which ignore modeling the semantic relatedness between knowledges.

In order to learn better knowledge representations to model the complicated semantic correlations between knowledge triples, we propose a new model, named ETransR. ETransR is the extension form of TransR, which models entities and relations in distinct vector spaces, i.e., entity space and multiple relation spaces, and performs translation in the corresponding relation space. The basic idea of ETransR is illustrated in Figure 1. In ETransR, for each knowledge tripe (h, r, t) , it first embeds entities in entity space, relations in relation space. And then it projects entity embeddings into relation space by relation-specific matrix M_r , which can obtain $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$. The model can make the head/tail entities that actually hold the relation (denoted as circles) close with each other, and get far away from those that do not hold the relation (denoted as triangles).

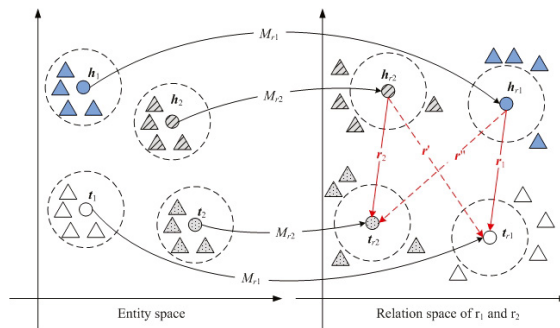


Figure 1: The illustration of ETransR

Given two knowledge triples (h_1, r_1, t_1) and (h_2, r_2, t_2) , if they are two mention forms of a unique knowledge, then they will satisfy $\mathbf{t}_{r_1} - \mathbf{h}_{r_1} \approx \mathbf{t}_{r_2} - \mathbf{h}_{r_2}$. For head entities, if h_1 is the same as h_2 , it will have $\mathbf{h}_{r_2} + \mathbf{r}_1 \approx \mathbf{t}_{r_1}$, $\mathbf{h}_{r_1} + \mathbf{r}_2 \approx \mathbf{t}_{r_2}$. And for tail entities, if t_1 is the same as t_2 , it

will have $\mathbf{h}_{r_1} + \mathbf{r}_1 \approx \mathbf{t}_{r_2}$, $\mathbf{h}_{r_2} + \mathbf{r}_2 \approx \mathbf{t}_{r_1}$. In addition, considering that the representations of a group of knowledge mentions of a unique knowledge usually exhibit same semantic patterns, we extend ETransR by clustering same knowledge mentions into groups and learning knowledge embeddings for each group, named as CETransR (cluster-based ETransR), which follows the idea of piecewise linear regression [14].

Our main contribution is the development of an innovative knowledge representation model for measuring knowledge semantic correlations to help boost the knowledge resolution performance, which automatically learns entity and relation feature representations in continuous vector spaces. Experimental results on two benchmark datasets validate the effectiveness and efficiency of our method compared with the state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, we give a review of related work. Section 3 introduces our entity and relation embedding models for learning knowledge representations. Section 4 is devoted to the experimental results. Finally, the paper is concluded in Section 5.

2 Related Work

Knowledge embedding has received a lot of attentions in recent years, existing knowledge embedding methods aim to represent entities and relations of knowledge graphs as vectors in a continuous vector space, where they usually defined a loss function to evaluate the representations. A variety of approaches have been explored for knowledge embedding. They can be divided into three major categories.

The first category is the translation based approach. Some notable works, including TransE [3], TransH [17], TransR [11] and TransA [10] are effective and efficient. TransE assumes $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) is a golden triple, which indicates that \mathbf{t} should be the nearest neighbor of $\mathbf{h} + \mathbf{r}$. To deal with relations with different mapping properties, TransH is established to project entities into a relation-specific hyperplane and the relation becomes translating operation on hyperplane. For a relation r , TransH models the relation as a vector \mathbf{r} on a hyperplane with \mathbf{w}_r as the normal vector. For a triple (h, r, t) , the entity embeddings \mathbf{h} and \mathbf{t} are first projected to the hyperplane of \mathbf{w}_r , denoted as \mathbf{h}_\perp and \mathbf{t}_\perp . By projecting entity embeddings into relation hyperplanes, it allows entities playing different roles in different relations. Both TransE and TransH assume embeddings of entities and relations being in the same space \mathbb{R}^k . However, an entity may have multiple aspects, and various relations focus on different aspects of entities. Hence, it is intuitive that some entities are similar and thus close to each other in the entity space, but are comparably different in some specific aspects and thus far away from each other in corresponding relation spaces. To address this issue, TransR is proposed to model entities and relations in distinct spaces, i.e., entity space and multiple relation spaces, which are bridged by relation-specific matrices. In TransR, for each triple (h, r, t) , entities embeddings are set as $\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$ and relation embedding is set as $\mathbf{r} \in \mathbb{R}^d$. For each relation r , TransR sets a projection matrix $\mathbf{M}_r \in \mathbb{R}^{k \times d}$, which may projects entities from entity space to relation space. These methods define a global margin-based loss function to learn knowledge embedding representations, and share the same set of candidate settings for different entities and relations, which ignore the individual locality of knowledge representations and seems not convincing in theory. To address this issue, TransA is proposed to adaptively finds the optimal loss function by adaptively determining its margin over different knowledge triples.

The second category is the tensor decomposition based approach, which is a well-developed mathematical tool for data analysis. Some notable works, including TUCKER [6], RESCAL [13] and TRESICAL [7] belong to this category. TUCKER decomposition, also known as high-

order singular value decomposition (SVD), factorizes a tensor into a core tensor multiplied by a matrix along each dimension. RESCAL is a tensor factorization based method. Compared with other tensor factorizations, the main advantage of RESCAL is that it can exploit a collective learning effect when applied to relational data. RESCAL has shown very good results in various canonical relational learning tasks such as link predication. TRESICAL is an extension of RESCAL, which tries to encode rules into RESCAL. However, it focuses solely on a single rule, i.e., the arguments of a relation should be entities of certain types.

The third category is the energy based approach, which assigns low energies to plausible triples of a knowledge graph and employs neural network for learning. For example, Unstructured model [2] is proposed as a naive version of TransE by assigning all $\mathbf{r} = 0$, leading to score function $f_r(h, t) = \|\mathbf{h} - \mathbf{t}\|_2^2$. This model cannot consider differences of relations. Structured embedding (SE) model [4] designs two relation-specific matrices for head and tail entities, i.e., $M_{r,1}$ and $M_{r,2}$, and defines the score functions as an L_1 distance between two projected vectors, i.e., $f_r(h, t) = \|\mathbf{M}_{r,1}\mathbf{h} - \mathbf{M}_{r,2}\mathbf{t}\|_1$. Since the model has two separate matrices for optimization, it cannot capture precise relations between entities and relations. Neural tensor network (NTN) model [15] represents entities in a d -dimensional vectors created separately by averaging pre-trained word vectors, and then learns a $d \times d \times m$ tensor describing the interactions between these latent components in each of m relations. Meanwhile, the corresponding high complexity of NTN may prevent it from efficiently applying on large-scale knowledge graphs. However, these methods mainly focus on modeling single knowledge in continuous vector spaces, which ignore the semantic relatedness modeling between knowledges. We intend to propose an comprehensive translation method to automatically learning entity and relation feature representations in continuous vector spaces, in order to measure the semantic relatedness of knowledge mentions for knowledge resolution.

3 Embedding by Translating on Two Vector Spaces

In order to model the semantic correlations between knowledges, we propose ETransR model, which models entities and relations in distinct vector spaces, and performs translation in the corresponding relation space, in order to measure the semantic relatedness of knowledge mentions for knowledge resolution.

3.1 Embedding Model

In ETransR model, for each knowledge triple (h, r, t) , $\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$ represents entity embeddings of h and t , respectively; $\mathbf{r} \in \mathbb{R}^m$ represents relation embeddings of r . k denotes the entity embeddings dimension, m denotes the relation embeddings dimension. It is noted that k and m are not necessarily identical, i.e., $k \neq m$. For each relation r , we set a projection matrix $\mathbf{M}_r \in \mathbb{R}^{k \times m}$, to project entity embeddings in entity space to relation space. Specifically, for a triple (h, r, t) , the embeddings \mathbf{h} and \mathbf{t} are first projected to the relation space with projection matrix \mathbf{M}_r . The entities embedding projections are denoted as \mathbf{h}_r and \mathbf{t}_r , where $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$, $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$. We expect \mathbf{h}_r and \mathbf{t}_r can be connected by the relation embedding \mathbf{r} on the relation space with low error if (h, r, t) is a golden triple, and high error if (h, r, t) is a incorrect triple. Thus we define a score function $\|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_d$ to measure the plausibility that the knowledge triple is incorrect, where d denotes L_1 or L_2 -norm. Given two knowledge triples (h_1, r_1, t_1) and (h_2, r_2, t_2) , we define a score function to measure the dissimilarity between the two triples. The score function is correspondingly defined as

$$f_{r_1, r_2}(h_1, t_1; h_2, t_2) = \|(\mathbf{t}_{r_1} - \mathbf{h}_{r_1}) - (\mathbf{t}_{r_2} - \mathbf{h}_{r_2}) + (\mathbf{r}_1 - \mathbf{r}_2)\|_d \quad (1)$$

The score is expected to be lower for two similar triples and higher for two dissimilar triples. In this paper, we restrict $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$, $\|\mathbf{r}\|_2 \leq 1$, $\|\mathbf{h}\mathbf{M}_r\|_2 \leq 1$, $\|\mathbf{t}\mathbf{M}_r\|_2 \leq 1$. The model parameters consist of all elements of the entities’s embeddings vectors, relations’ embeddings vectors and relations’ projection matrices.

ETransR learns a unique vector for each relation, which may be under-representative to fit all knowledge triple mentions representing a unique knowledge triple, because these knowledge triple mentions exhibit same semantic patterns. In order to better model these knowledge triples, we extend ETransR by clustering same knowledge triple mentions into groups and learning knowledge embeddings for each group, named as CTransR (cluster-based ETransR), which follows the idea of piecewise linear regression [14].

In CTransR, it first segments input knowledge triple mentions into several groups, where each group represents a unique knowledge triple. Formally, for a specific knowledge triple (h, r, t) , all knowledge triple mentions in the dataset are clustered into multiple groups, and knowledge triple mentions in each group are exhibit same semantic patterns. All entity pairs (h, t) are represented with their vector offset $(\mathbf{t} - \mathbf{h})$ for clustering. Afterwards, we learn a separate relation vector \mathbf{r}_c for each cluster and projection matrix \mathbf{M}_r for each relation, respectively. The entities embeddings projection vectors are defined as $\mathbf{h}_{r,c} = \mathbf{h}\mathbf{M}_r$, $\mathbf{t}_{r,c} = \mathbf{t}\mathbf{M}_r$. Given two similar knowledge triples (h_1, r_1, t_1) and (h_2, r_2, t_2) , the score function is correspondingly defined as

$$f_{r_1,r_2}(h_1, t_1; h_2, t_2) = \|(\mathbf{t}_{r_1,c} - \mathbf{h}_{r_1,c}) - (\mathbf{t}_{r_2,c} - \mathbf{h}_{r_2,c})\|_d + \alpha(\|\mathbf{r}_c - \mathbf{r}_1\|_d - \|\mathbf{r}_c - \mathbf{r}_2\|_d) \quad (2)$$

where $(\|\mathbf{r}_c - \mathbf{r}_1\|_d - \|\mathbf{r}_c - \mathbf{r}_2\|_d)$ aims to ensure cluster-specific relation vector \mathbf{r}_c not far away from the original relation vector \mathbf{r}_1 and \mathbf{r}_2 ; α controls the effect of this constraint, $0 \leq \alpha \leq 1$. In this paper, we restrict $\|\mathbf{h}\|_2 \leq 1$, $\|\mathbf{t}\|_2 \leq 1$, $\|\mathbf{r}\|_2 \leq 1$, $\|\mathbf{r}_c\|_2 \leq 1$, $\|\mathbf{h}_{r,c}\|_2 \leq 1$, $\|\mathbf{t}_{r,c}\|_2 \leq 1$. The model parameters consist of all elements of the entities’s embeddings vectors, relations’ embeddings vectors and relations’ projection matrices.

3.2 Training

To learning knowledge embeddings with our proposed models, we define a margin-based ranking criterion as objective for training:

$$\mathcal{L} = \sum_{((h_1,r_1,t_1);(h_2,r_2,t_2)) \in S} \sum_{((h'_1,r_1,t'_1);(h'_2,r_2,t'_2)) \in S'_{(h_1,r_1,t_1;h_2,r_2,t_2)}} [\gamma + f_{r_1,r_2}(h_1, t_1; h_2, t_2) - f_{r_1,r_2}(h'_1, t'_1; h'_2, t'_2)]_+ \quad (3)$$

where $[x]_+ \triangleq \max(0, x)$; γ is a margin hyperparameter, $\gamma > 0$; S is the correct triples set; S' is the incorrect triples set, constructed according to Equation 4, is composed of correct triples with either the head or tail entity replaced by a random entity, but not both at the same time.

$$S'_{(h_1,r_1,t_1;h_2,r_2,t_2)} = \{(h'_1, r_1, t_1); (h'_2, r_2, t_2) | h_1, h_2 \in E\} \cup \{(h_1, r_1, t'_1); (h_2, r_2, t'_2) | t_1, t_2 \in E\} \quad (4)$$

More specifically, during constructing incorrect triples, we follow the method described in [17], which sets different probabilities for replacing the head or tail entity for corrupting the golden triple. We denote this sampling method as “bern”. In addition, we also construct incorrect triples by randomly corrupting the golden triple, which sets equal probability for replacing the head or tail entity. We denote this sampling method as “unif”. In experiments, we will validate

the impact that how the performance of our approach changes with the “unif” and “bern” incorrect triple sampling methods.

The loss function (Equation 3) favors lower values for similar triples than for dissimilar triples, and is thus a natural implementation of the intended criterion. It is worthwhile to note that for a given entity, its embedding vector is the same when the entity appears as the head or as the tail of a triple. The optimization is carried out by stochastic gradient descent (SGD) in minibatch mode to minimize the above loss function. The set of golden triples are randomly traversed multiple times. After a minibatch, the gradient is computed and the model parameters are updated. The detailed optimization procedure is described in Algorithm 1. It firstly exploits the random procedure proposed in [3] to initialize all entities embeddings and relations embeddings, and uses identity matrix to initialize all relation-specific projection matrices. All the entities embedding vectors, relation embedding vectors and projection matrices are first normalized during each main iteration. Then, it samples a small set of triples from the training dataset, and adds these triples into the training triples of the minibatch. It is noted that, for each such triple, a single incorrect triple is sampled. In the algorithm, the parameters are then updated by taking a gradient step with constant learning rate. The algorithm is stopped based on its performance on a validation dataset.

Algorithm 1 Model learning algorithm

Input: Correct triples set S , incorrect triples set S' , entities set E , relations set R , entity embedding dimension k , relation embedding dimension m , margin γ , data batch size B ;

Output: The well trained embedding model;

```

1: for all  $e \in E, r \in R$  do
2:    $\mathbf{e} \leftarrow \text{uniform}(-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}})$ ;
3:    $\mathbf{r} \leftarrow \text{uniform}(-\frac{6}{\sqrt{m}}, \frac{6}{\sqrt{m}})$ ;
4:    $\mathbf{M}_r \leftarrow \mathbf{I}_{k \times m}$ ;
5: end for
6: loop
7:    $\text{norm}(\mathbf{e}), \text{norm}(\mathbf{r}), \text{norm}(\mathbf{eM}_r)$ ;
8:    $S_{batch} \leftarrow \text{Sample}(S, B)$ ;
9:    $T_{batch} \leftarrow \emptyset$ ;
10:  for  $(h_1, r_1, t_1), (h_2, r_2, t_2) \in S_{batch}$  do
11:     $(h'_1, r_1, t'_1) \leftarrow \text{Sample}(S'_{(h_1, r_1, t_1)})$ ; // sample a incorrect triple according to  $(h_1, r_1, t_1)$ 
12:     $(h'_2, r_2, t'_2) \leftarrow \text{Sample}(S'_{(h_2, r_2, t_2)})$ ; // sample a incorrect triple according to  $(h_2, r_2, t_2)$ 
13:     $T_{batch} \leftarrow T_{batch} \cup \{(h_1, r_1, t_1), (h_2, r_2, t_2), (h'_1, r_1, t'_1), (h'_2, r_2, t'_2)\}$ ;
14:  end for
15:  Update knowledge embeddings w.r.t.  $\sum_{((h_1, r_1, t_1); (h_2, r_2, t_2)); ((h'_1, r_1, t'_1); (h'_2, r_2, t'_2))) \in T_{batch}} \nabla[\gamma + f_{r_1, r_2}(h_1, t_1; h_2, t_2) - f_{r_1, r_2}(h'_1, t'_1; h'_2, t'_2)]_+$ ;
16: end loop
17: return The trained model;

```

4 Experiments

In this section, we will evaluate the effectiveness of our proposed method for the knowledge resolution task. We will: (1) compare the knowledge resolution accuracy of our method, with

four state-of-the-art baseline methods; (2) show how the performance of our method changes with respect to the increase of the number of the entity and relation embedding dimensions.

4.1 Experimental Settings

All experiments were conducted on a server running 64-bit Linux OS, with 16 core 2GHz AMD Opteron(tm) 6128 Processors and 32GB RAM. In the experiments, we used two widely used datasets: one from WordNet [12], and the other from Freebase [1]. For the dataset from WordNet, we employ WN11 used in [15]. For the dataset from Freebase, we employ FB15K used in [2]. WN11 contains 11 relation types and 38,696 related entities. FB15K contains 1,345 relation types and 14,951 related entities. They were randomly split into three groups, i.e., train data, valid data and test data. The details of these datasets are shown in Table 1.

Dataset	WN11	FB15K
Entities	38,696	14,951
Relations	11	1,345
Train	112,581	483,142
Valid	2,609	50,000
Test	10,544	59,071

Table 1: Statistics of the datasets

As knowledge resolution aims to cluster knowledge triples into several disjoint groups, this means it is to judge whether a given triple is correctly classified or not. This can be seen as a classification task. Evaluation of classification needs negative labels. The dataset WN11 already contains negative triples, which are obtained by corrupting by correct ones. For FB15K, we construct the incorrect triples following the same way as [15]. The decision rule for classification is as follows. For a triple (h, r, t) , if the dissimilarity score obtained by f_r is less than a relation-specific threshold δ_r , then the triple is classified to be positive, and negative otherwise. δ_r is determined by maximizing classification accuracy on the validation set.

4.2 Experimental Results and Analysis

Triple classification accuracy. For experiments of ETransR and CTransR, the learning rate λ for stochastic gradient descent (SGD) is select among $\{0.001, 0.005, 0.01, 0.05, 0.1\}$, the margin γ among $\{1, 2, 4, 8, 10\}$, the dimensions of entity embedding k , relation embedding m among $\{10, 20, 40, 50, 60, 100, 200\}$, the batch size B among $\{20, 30, 70, 120, 420, 480, 960, 1440\}$, the α for CTransR among $\{0.001, 0.005, 0.01, 0.1\}$. The dissimilarity measure d is set either to the L_1 -norm or L_2 -norm. All parameters are determined according to classification accuracy on the validation set. Regarding the strategy of constructing negative labels, we use “unif” to denote the traditional way of replacing head entity or tail entity with equal probability, and use “bern” to denote reducing false negative labels by replacing head entity or tail entity with different probabilities. Under the “unif” setting, the optimal configurations are: $\lambda = 0.01$, $\gamma = 1$, $k = 40$, $m = 40$, $B = 70$, $\alpha = 0.1$ and taking L_1 -norm as dissimilarity measure on WN18; $\lambda = 0.005$, $\gamma = 2$, $k = 60$, $m = 60$, $B = 180$, $\alpha = 0.005$ and taking L_1 -norm as dissimilarity measure on FB15K. Under the “bern” setting, the optimal configurations are: $\lambda = 0.01$, $\gamma = 2$, $k = 50$, $m = 50$, $B = 420$, $\alpha = 0.01$ and taking L_1 as dissimilarity measure on WN18; $\lambda = 0.001$, $\gamma = 2$, $k = 100$, $m = 100$, $B = 480$, $\alpha = 0.01$ and taking L_1 as dissimilarity measure on FB15K. For both datasets, we traverse all the training triples for 1000 rounds.

Experimental results on WN11 and FB15K are shown in Table 2. From Table 2 we can see that: (1) ETransR and CTransR outperform baseline methods consistently. It indicates that ETransR and CTransR find a better model to express the semantic features of entities and relation types. (2) All the methods using “bern” sampling strategy perform better than using “unif” strategy, especially on FB15K which have much more relation types. (3) CTransR performs better than ETransR. It indicates that a fine-grained model may handle complicated internal semantic correlations under each relation type better than coarse-grained model.

Approach	Accuracy	
	WN11	FB15K
TransE(unif)	0.759	0.796
TransE(bern)	0.759	0.802
TransH(unif)	0.777	0.790
TransH(bern)	0.788	0.802
TransR(unif)	0.855	0.817
TransR(bern)	0.859	0.839
CTransR(unif)	0.856	0.841
CTransR(bern)	0.857	0.845
ETransR(unif)	0.871	0.864
ETransR(bern)	0.873	0.869
CTransR(unif)	0.882	0.873
CTransR(bern)	0.885	0.876

Table 2: Evaluation results of triple classification

Some relation types classification results in FB15K triples with CTransR are shown in Table 3. From Table 3 we can find obvious patterns that: Category#1 is about team-play position relation type, Category#2 is about award-winner relation type, Category#3 is about file-writer relation type. It is obvious that by clustering we can learn more and fine-grained knowledge embeddings, which can further help improve the performance of triple classification.

No.	Relation types
1	/soccer/football_player/position_s
	/soccer/football_player/current_team./sports/sports_team_roster/position
	/sports/pro_athlete/teams./sports/sports_team_roster/position
2	/award/award_winning_work/awards_won./award/award_honor/award_winner
	/award/award_ceremony/awards_presented./award/award_honor/award_winner
	/award/award_category/winners./award/award_honor/award_winner
3	/film/film/written_by
	/film/film/writer

Table 3: Evaluation results of triple classification

Impact of entity and relation embedding dimensions. Next we conducted experiments over the FB15K dataset to illustrate how the performance of our CTransR method changes, with respect to the increase of the number of entities and relation embedding dimensions, using “bern” sampling strategy. Here the parameter settings are the same as those described in the above experiments. The results are graphically depicted in Figure 2.

From Figure 2 we can see that: (1) The triple classification accuracy increases as the number

of entity and relation embedding dimensions increases. It indicates that fine-grained entity and relation embeddings are better for modeling knowledge complicated semantic relationships. (2) When we fixed the relation embedding dimension m , the triple classification accuracy increases as the number of entity embedding dimension k increases. But the increasing speed slows down, vice versa. In addition, the accuracy decreases when the value of $|k - m|$ increases. It indicates that it should ensure the granularity of entity embeddings and relation embeddings, otherwise it will cause distortion of entity and relation semantic representations.

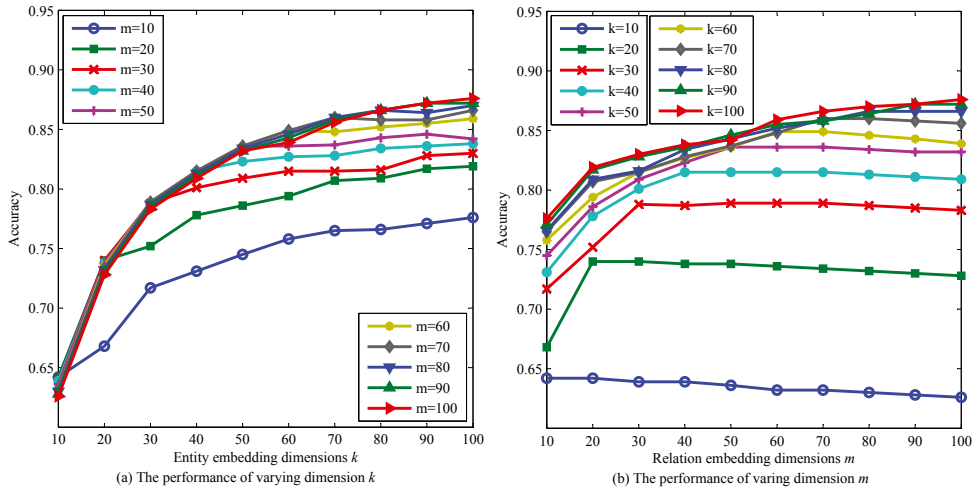


Figure 2: The impact of the entity and relation embedding dimensions on the triple classification of CTransR over the FB15K dataset

5 Conclusion

In this paper, we studied the problem of knowledge resolution. We proposed a knowledge representation method to automatically learning the feature representations of entities and relations in embedding spaces. This paper proposed ETransR and CTransR models, which embed entities and relations in continuous vector spaces. These models aim to model complicated semantic correlations between knowledge triple mentions, in order to improve the performance of synonym and multi-sense knowledge mentions resolution. Experimental results show that this method outperforms the state-of-the-art methods on the task of knowledge resolution.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2016YFB1000902), the National Natural Science Foundation of China (No. 61602467, 61502478, 61402464, 61402473).

References

- [1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the*

- ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
- [2] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.
 - [3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.
 - [4] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Conference on Artificial Intelligence*, number EPFL-CONF-192344, 2011.
 - [5] David Guy Brizan and Abdullah Uz Tansel. A survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, 6(3):5, 2015.
 - [6] Kai-Wei Chang, Wen-tau Yih, and Christopher Meek. Multi-relational latent semantic analysis. In *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1602–1612, 2013.
 - [7] Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1579, 2014.
 - [8] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM, 2014.
 - [9] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 3161–3165. AAAI Press, 2013.
 - [10] Yantao Jia, Yuanzhuo Wang, Hailun Lin, Xiaolong Jin, and Xueqi Cheng. Locally adaptive translation for knowledge graph embedding. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 992–998, 2016.
 - [11] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187, 2015.
 - [12] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
 - [13] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM, 2012.
 - [14] HP Ritzema. *Drainage principles and applications*. Number 16. ILRI, 2006.
 - [15] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.
 - [16] Norases Vesdapunt, Kedar Bellare, and Nilesh Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.
 - [17] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.
 - [18] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 481–492, 2012.