Translating Embeddings for Knowledge Graph Completion with Relation Attention Mechanism

Wei Qian[†], Cong Fu[†], Yu Zhu[†], Deng Cai[†♯*], Xiaofei He[⋆†]
[†]State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China
[♯]Alibaba-Zhejiang University Joint Institute of Frontier Technologies
[⋆]Fabu Inc., Hangzhou, China
{qwqjzju, fc731097343}@gmail.com, {zhuyu_cad, dcai}@zju.edu.cn, xiaofeihe@fabu.ai

Abstract

Knowledge graph embedding is an essential problem in knowledge extraction. Recently, translation based embedding models (e.g., TransE) have received increasingly attentions. These methods try to interpret the relations among entities as translations from head entity to tail entity and achieve promising performance on knowledge graph completion. Previous researchers attempt to transform the entity embedding concerning the given relation for distinguishability. Also, they naturally think the relation-related transforming should reflect attention mechanism, which means it should focus on only a part of the attributes. However, we found previous methods are failed with creating attention mechanism, and the reason is that they ignore the hierarchical routine of human cognition. When predicting whether a relation holds between two entities, people first check the category of entities, then they focus on fined-grained relation-related attributes to make the decision. In other words, the attention should take effect on entities filtered by the right category. In this paper, we propose a novel knowledge graph embedding method named TransAt to learn the translation based embedding, relation-related categories of entities and relation-related attention simultaneously. Extensive experiments show that our approach outperforms state-of-the-art methods significantly on public datasets, and our method can learn the true attention varying among relations.

1 Introduction

Knowledge graphs [Bakker, 1987] are directed graphs consisting of entities as nodes and relations as edges. Typically, a knowledge graph encodes structured information of millions of entities and billions of relational facts. But it is still far from completeness. Therefore, the purpose of knowledge graph completion is to predict new relations between entities according to the existing knowledge graph.

*corresponding author

Recently, embedding based methods [Zhu *et al.*, 2016] that encode each object in knowledge graphs into a continuous vector space present powerful effects. Thus, this kind of approach becomes increasingly popular. Among these methods, translation-based methods are the most popular for their simplicity and effectiveness. They obtain state-of-the-art link prediction performance. Inspired by the success of word embedding [Mikolov *et al.*, 2013; Zou *et al.*, 2013], TransE [Bordes *et al.*, 2013] learns vector embeddings for both entities and relationships. For a triple (h, r, t) which means head entity h has relation r with tail entity t, the basic idea of TransE model is that this triple induce a functional relation corresponds to a translation of the embeddings of entities in vector space $\mathcal{R}^k$, that is, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

There are a variety of problems for TransE, and a series of works are proposed to make up its shortcomings, such as TransH [Wang *et al.*, 2014], TransR [Lin *et al.*, 2015], TranSparse [Ji *et al.*, 2016], etc. Since TransE could not model one-to-many, many-to-one and many-to-many relations, TransH is proposed to address this issue by introducing relation-related projection vectors and projecting entities to relation-related hyperplanes. However, different relations may focus on just a few different attributes of entities. TransR is proposed to address this issue by introducing relation-related transformation matrix M_r and transform entity vectors to different relation-spaces. Still, relations are heterogeneous, and imbalance and thus TranSparse is proposed to address this issue by introducing complex relation-related transformation matrix.

We believe human cognition for a relation follows a hierarchical routine and there exist categories among entities. As shown in Figure 2, we will first analyze whether the category of candidate entities is reasonable by some attributes of entities and then determine whether the specific relationship is valid by other attributes. However, no matter how the previous methods transform the entity embeddings given corresponding relations, they finish cognition in one step. As a consequence, they must take some non-relation-related attributes (but maybe category-related) into consideration since they need to analyze category in the relationship discrimination step and can not focus on just a few different attributes of entities. We also prove this experimentally. Figure 1 shows

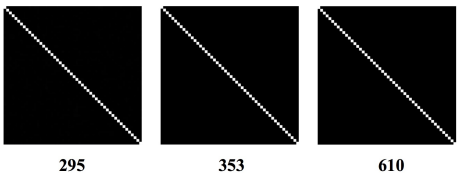

Figure 1: Transformation matrices learned by TransR. These are some randomly selected transformation matrices learned by TransR on FB15k. Each entry of matrix is multiplied by 255 and each matrix is shown as a grayscale image. The numbers below the figures are the relation IDs. Actually, all transformation matrices are similar and approximate to the unit matrix. This means TransR fails to learn transformations only focusing on part of dimensions.

some examples of transformation matrices learned by TransR. They are approximate to the unit matrix. It means their attention mechanisms do not work.

In this paper, we propose a knowledge graph embedding method with attention mechanism named TransAt (Translation with Attention). The basic idea of TransAt is illustrated in Figure 2. We hold the view that when we consider whether a relationship described by a triple is true, it is done through a two-stage process. In the first stage, we will check whether the categories of head and tail entities with respect to a given relation make sense. Then, for those possible compositions, we need to consider whether the relationship holds based on the relation-related dimensions (attributes). Thus, we use the following function to evaluate a triple (h, r, t).

$$f_r(\mathbf{h}, \mathbf{t}) = \begin{cases} P_r(\mathbf{h}) + \mathbf{r} - P_r(\mathbf{t}) & h \in H_r, t \in T_r \\ \infty & otherwise \end{cases}$$

where $\mathbf{h}, \mathbf{t}$ are two entity vectors, $\mathbf{r}$ is a relation vector, $H_r(T_r)$ is capable head (tail) candidate set for relation r, P_r is a projection which only keeps dimensions related to r.

We evaluate our models with the tasks of link prediction and triple classification on public benchmark datasets: WN18, FB15k, WN11, and FB13. Experimental results show that our approach outperforms state-of-the-art models significantly.

It's worthwhile to highlight our main contributions of this paper as follows:

- We point out that previous approaches can not learn an indeed attention mechanism and analyze the reasons;
- We propose a method which genuinely pays attention to a part of the dimensions;
- We propose an asymmetric mechanism which can further improve the performance;
- Our experimental results outperform state-of-the-art models significantly.

The rest of this paper is organized as follows. In section 2, we provide a brief review of related works about knowledge graph embedding. Our detailed motivation and the two versions of our method are introduced in section 3. We show our experimental results and provide a comprehensive analysis of the experimental results in section 4. Finally, we provide some concluding remarks in section 5.

2 Related Work

Notation. Here we briefly introduce mathematical notations used in this paper. We denote a triplet by (h, r, t), their embeddings by bold lower case letters **h, r, t** and matrices by bold upper case letters, such as **M**. Score function is represented by $f_r(\mathbf{h}, \mathbf{t})$.

2.1 Translation-based Models

TransE [Bordes *et al.*, 2013] holds the view that the functional relation induced by a triple (h, r, t) corresponds to a translation of the embeddings of entities. Specifically, the entities h and t are learned to be low-dimensional embeddings, and their relationship r is represented as a translation in the embedding space, i.e. $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Hence, the score function is $f_r(\mathbf{h}, \mathbf{t}) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||^2_{l1/2}$. TransE is effective and applies well to irreflexive and one-to-one relations. However, since the representation of an entity is the same when involved in any relations in TransE, it has problems when dealing with reflexive or many-to-one/one-to-many/many-to-many relations [Wang *et al.*, 2014].

TransH [Wang *et al.*, 2014] tries to overcome the problems of TransE by enabling an entity to have distributed representations when the entity is involved in different relations. Specifically, given a triplet (h, r, t), the entity embeddings $\mathbf{h}$ and $\mathbf{t}$ are firstly projected to the relation-specific hyperplane to obtain their projections $\mathbf{h}_{\perp r}$ and $\mathbf{t}_{\perp r}$. The translation embedding $\mathbf{d}_r$ is simply posited in the hyperplane. Finally, its score function is defined as $f_r(h, t) = ||\mathbf{h}_{\perp r} + \mathbf{d}_r - \mathbf{t}_{\perp r}||^2_{l1/2}$.

TransR/CTransR [Lin *et al.*, 2015] proposes that an entity may have multiple attributes and various relations may focus on different attributes of entities. TransE and TransH use a common embedding space for both entities and relations, which is insufficient for modeling this. To address this issue, TransR/CTransR models entities and relations in different embedding spaces, i.e. the entity space and the relation space, and performs the translation in the relation space. It sets a transfer matrix $\mathbf{M}_r$ for each relation r to map entity embeddings to the relation space. The score function is $f_r(\mathbf{h}, \mathbf{t}) = ||\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}||^2_{l1/2}$.

TransD [Ji *et al.*, 2015] considers the diversity of not only relations but also entities. In addition, it has fewer parameters and replaces matrix-vector multiplication by vector-vector multiplication for an entity-relation pair, which is more scalable and can be applied to large scale graphs.

KG2E [He *et al.*, 2015b] switches to density-based embedding. This is different from previous translation models, which focus on point-based embedding. The point-based embedding methods ignore that different entities and relations may contain different (un)certainties. To address this, KG2E uses Gaussian embedding to model the data uncertainty. It works well on one-to-many and many-to-one relations.

TranSparse [Ji *et al.*, 2016] focuses on solving the heterogeneity and imbalance issues in knowledge graphs, which are ignored by previous translation models. The *heterogeneity* refers to that different relations link different number of entity pairs, and the *imbalance* means that the number of head entities and that of tail entities in a relation could be different. To address these two issues, TranSparse replaces trans-

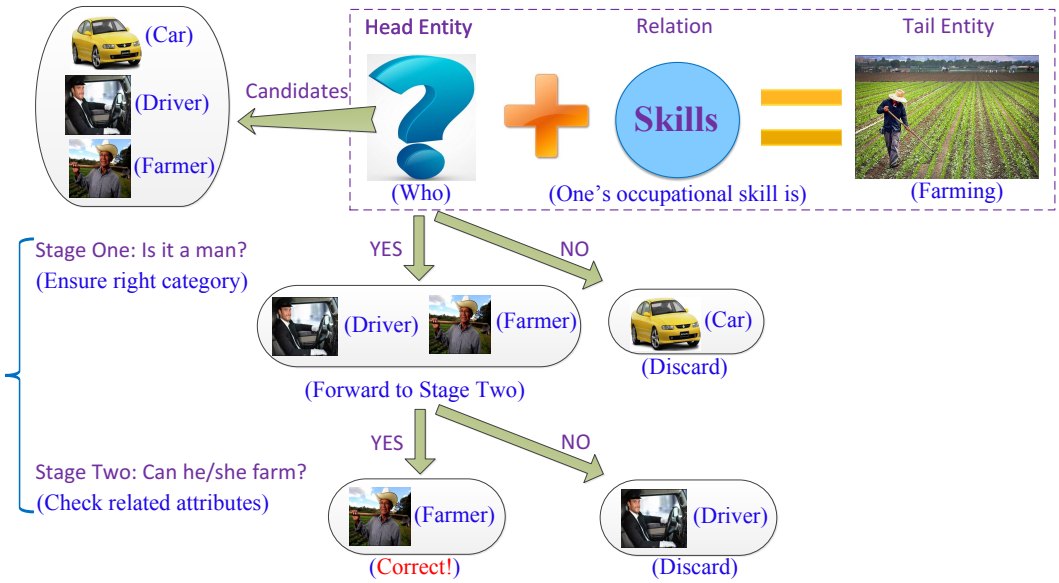

Figure 2: A simple illustration of our motivation. Above is a simple prediction task. From the human perspective, we will reject entities from wrong categories first, so 'car' is not in the option. Then we will focus on relation-related attributes 'occupational skill' and choose 'farmer'.

fer matrices with adaptive sparse matrices, whose sparse degrees are determined by the number of entity pairs or entities linked by relations. Its score function is $f_r(\mathbf{h},\mathbf{t}) = ||\mathbf{M}_r^h(\theta_r^h)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^t(\theta_r^t)\mathbf{t}||_{l1/2}^2$, where $\mathbf{M}_r^h(\theta_r^h)$ and $\mathbf{M}_r^t(\theta_r^t)$ are adaptive sparse matrices for head entities and tail entities, respectively. θ_r^h and θ_r^t are their sparse degrees.

2.2 Other Methods

Structured Embedding (SE) [Bordes *et al.*, 2011]. For each relation r, SE sets two different matrices $\mathbf{M}_r^h$ and $\mathbf{M}_r^t$ to project head entities and tail entities, respectively. Its score function is $f_r(\mathbf{h},\mathbf{t}) = ||\mathbf{M}_r^h\mathbf{h} - \mathbf{M}_r^t\mathbf{t}||_1$.

Semantic Matching Energy (SME) [Bordes *et al.*, 2012; 2014]. SME exploits a neural network to encode each entity and relation into vectors and makes them interact with each other via linear algebra operates.

Latent Factor Model (LFM) [Jenatton *et al.*, 2012]. LFM considers the second-order correlations between entity embeddings with a quadratic form. Specifically, it encodes each entity (including head and tail entities) as a vector and sets a matrix $\mathbf{M}_r$ for each relation r. It defines the score function as $f_r(\mathbf{h},\mathbf{t}) = \mathbf{h}\mathbf{M}_r\mathbf{t}$.

Single Layer Model (SLM) [Socher *et al.*, 2013]. SLM uses relation-specific weights $\mathbf{u}_r$ and a non-linear function $tanh$ on triplet representation $\mathbf{W}_{rh}\mathbf{h} + \mathbf{W}_{rt}\mathbf{t} + \mathbf{b}_r$ to represent the score function, which is as $f_r(\mathbf{h},\mathbf{t}) = \mathbf{u}_r^T tanh(\mathbf{W}_{rh}\mathbf{h} + \mathbf{W}_{rt}\mathbf{t} + \mathbf{b}_r)$.

Neural Tensor Network (NTN) [Socher *et al.*, 2013]. NTN extends SLM model by considering the second-order correlations between the head and tail entity embeddings. They are combined with a matrix parameterized by the relation.

GAKE [Feng *et al.*, 2016]. GAKE formulates the triple set in a knowledge graph as a unified directed graph and introduces three types of graph context for embedding: neighbor context, path context and edge context, where each reflects properties of knowledge from different perspectives. Then it learns embeddings for vertices and edges by leveraging these context information.

2.3 The ORC structure

[Zhang, 2017] proposes that the embedding methods should consider graph structures. It exploits substructures called "one-relation-circle" (ORC) to improve the performance of existing translation-based models, such as TransH, TransR and TransD.

The ORC structure refers to that some entity nodes form a circle by edges representing the same relation r in a knowledge graph. Then the corresponding relation vector $\mathbf{r}$ would approximate 0 (i.e. $\mathbf{r} \approx 0$) when using traditional translation-based methods to model related triplets. The paper uses an asymmetric operation on head entities and tail entities to solve this problem.

3 Our Method

3.1 Motivation

We believe human cognition for a relation follows a hierarchical routine and there exist categories among entities. All attributes form it discriminative from entities of other categories, and some make it discriminative from entities within the same category. As shown in Figure 2, let us consider a triple (x, one's occupational skill is, farming), where x needs to be determined. Intuitively, people will exclude the entity 'car', because it doesn't belong to the category possessing the attribute 'occupational skill'. 'Farmer' is obviously more suitable to fit in this position than 'driver' when taking the 'farming' skill into account. They all have the attribute 'occupational skill', but they differ from each other on it. The driver and the farmer have many other common attributes,

such as wage, working hours, etc, but for this relation, we only care about the 'occupational skill' attribute. In a few words, we believe the link prediction process has two stages. In the first stage, we will collect candidate entities from reasonable categories. In the second stage, for those possible triplet compositions, we need to focus on fine-grained attributes of entities to distinguish them concerning the particular relation. This process brings up two more tasks besides learning embedding, which are learning relation-related candidates and relation-related attention.

In the ideal scenario, given a relation, the relevant entities should have significant differences within a particular subset of dimensions, or say subspace. While at a macro perspective, the proper candidate entities for the relation differ from others on an overall consideration of all their attributes. There are previous papers (e.g. TransH, TransR, TranSparse) proposing that an entity has multiple attributes and various relations focus on different attributes of entities. However, they fail to achieve this. For example, we can see from Figure 1 that the transformation matrix learned by TransR [1] is approximate to the unit matrix. The main reason is that although they try to determine the transformation matrix for different relations, they hope to fulfill the category filtering and fine-grained category filtering simultaneously at a scale of all entities. As we stated above, distinguishing entities among different categories and fine-grained difference involve all attributes at all level. This demands the transformation matrix considering all dimensions of entity vector and reaches a contradiction to their purpose of focusing on the part of them. As a consequence, their models failed to learn the fine-grained attention.

3.2 TransAt

Inspired by above, we propose a novel algorithm to learn the embedding, relation-related candidates and relation-related attention simultaneously. We propose a piecewise evaluation function,

$$f_r(\mathbf{h},\mathbf{t}) = \begin{cases} P_r(\mathbf{h}) + \mathbf{r} - P_r(\mathbf{t}) & h \in H_r, t \in T_r \\ \infty & otherwise \end{cases} \quad (1)$$

where $H_r(T_r)$ is capable head (tail) candidate set for the relation r, P_r is a projection which only keeps dimensions related to r.

Equation (1) means that if the head entity or the tail entity is not suitable for relation r, their distance for relation r is infinite. If both the head entity and the tail entity are suitable for relation r, their distance to relation r is similar to TransE but only focuses on interested dimensions.

Now the problem becomes how to obtain the capable candidate sets and how to set the projection function. For the candidate set, we use Kmeans [Pedregosa *et al.*, 2011] to aggregate all entities to some clusters and merge clusters which have head (tail) entities in triples related to relation r to constitute head (tail) candidate set for relation r. Though some datasets have ground truth categories, we use Kmeans to generate categories for generality.

[1] We use the code publicly available at https://github.com/thunlp/KB2E

Following PCA [Dunteman, 1989], we hold the view that if a dimension is related to relation r, then the variance on this dimension among the candidate entities for relation r should be large and vice versa. So we use head (tail) entities capable of relation r in training set to obtain variances for each dimension and use a threshold to determine whether a dimension should be retained. In order not to introduce additional parameters, we set the threshold to be half of the maximum variance. Specifically, $P_r(\mathbf{h}) = \mathbf{a}_r * \mathbf{h}$, where $\mathbf{a}_r \in \{0,1\}^k, \mathbf{h} \in \mathcal{R}^k$.

Considering that some relations have ORC structures, we propose an asymmetric version of score function. ORC structure proposed in [Zhang, 2017] means that some entity nodes form a circle by edges with the same relation in a knowledge graph. ORC structures will make some relation vector $||\mathbf{r}|| \approx 0$. It can be solved by using an asymmetric operation on head entity and tail entity, which means the same entity will have different representations of head position and tail position. Since we hope that there exists only single embedding representation for entities, we use two vectors to scale each dimension of origin embeddings on head position and tail position respectively. If both the head entity and the tail entity are suitable for relation r, our asymmetric score function becomes,

$$f_r(\mathbf{h},\mathbf{t}) = P_r(\sigma(\mathbf{r}_h)\mathbf{h}) + \mathbf{r} - P_r(\sigma(\mathbf{r}_t)\mathbf{t}) \quad (2)$$

where P_r is a projection which only keeps dimensions related to r, σ is sigmoid function and r_h, r_t are two vectors related to relation r.

3.3 Learning

To implement our method, we define the following margin-based ranking loss for effective discrimination between positive triplets (h,r,t) and negative triplets(h',r,t'):

$$\begin{aligned} \mathcal{L} &= \sum_{h' \in H_r, t' \in T_r} [f_r(\mathbf{h},\mathbf{t}) + \gamma - f_r(\mathbf{h}',\mathbf{t}')]_+ \quad (3) \\ &+ \alpha\Big(\sum_{h' \notin H_r, t' \in T_r} [||\mathbf{t} - \mathbf{t}'||_2 + \gamma - ||\mathbf{h} - \mathbf{h}'||_2]_+ \\ &+ \sum_{h' \in H_r, t' \notin T_r} [||\mathbf{h} - \mathbf{h}'||_2 + \gamma - ||\mathbf{t} - \mathbf{t}'||_2]_+ \Big) \end{aligned}$$

where $[x]_+ := max(0,x)$ aims to obtain the maximum between 0 and x. γ is the margin separating positive and negative triplets. Following prior methods, we enforce constraints on the norms of the embeddings **h**, **r**, **t**, such that $\forall h,r,t, ||\mathbf{h}||_2 \le 1, ||\mathbf{r}||_2 \le 1, ||\mathbf{t}||_2 \le 1$.

This means if both head and tail entities of a corrupted triple are in the suitable candidate set, we hope it will have a massive score on our score function. Otherwise, we hope entities not in the same are candidate set far from each other.

We use Xavier initialization [He *et al.*, 2015a] to initialize the embeddings. Since sampling matters in embedding learning [Wu *et al.*, 2017; Kanojia *et al.*, 2017], we use a common sampling method for fair comparison. Then we use "bern" method [Wang *et al.*, 2014] to sample batches of data and use SGD [Bottou, 2010] to optimize our loss function batch by batch. In particular, we reset candidate sets and attention vectors when it reaches a given number of iterations.

Dataset	#Rel	#Ent	#Train	#Valid	#Test
WN11	11	38,696	112,581	2,609	10,544
WN18	18	40,493	141,442	5,000	5,000
FB13	13	75,043	316,232	5,908	23,733
FB15k	1,345	14,951	483,142	50,000	59,071

Table 1: Statistics of datasets

4 Experiments

4.1 Datasets

Our experiments are carried out on public knowledge graphs dataset retrieved from WordNet [Miller, 1995] and Freebase [Bollacker *et al.*, 2008] corpus. WordNet is a corpus designed to produce an intuitively usable dictionary and supports automatic lexical analysis. Freebase is an online collection of structured text data harvested from many sources, including individual, user-submitted wiki contributions. Specifically, WN11 [Socher *et al.*, 2013] and WN18 [Bordes *et al.*, 2013] are knowledge graphs retrieved from WordNet, while FB15k [Bordes *et al.*, 2013] and FB13 [Socher *et al.*, 2013] are extracted from FreeBase. All these datasets consist of training set, validation set and testing set which are well organized. Table 1 lists the data statistics of the four datasets.

4.2 Link Prediction

Following [Bordes *et al.*, 2013], we also use link prediction to estimate our method. Link prediction for knowledge graph is to predict the missing h or t for a correct triplet (h, r, t). Instead of obtaining the best one entity, this task emphasizes the rank of the origin right entity. The evaluation of this task mainly contains two metrics: the average rank of correct entities (Mean Rank) and the proportion of correct entities ranked in top 10 (Hits@10).

The Mean Rank evaluation contains two parts. One is to evaluate the head entity prediction. It requires us to replace the real head entity for each triple in the test set with all the entities in the dictionary in turn which produces sets of fake triples containing the ground truth. Then, we use our score function to compute scores of fake triplets and rank each fake set to the original triple in ascending order. Next, we adjust those entities in the candidate set for relation r ahead of non-candidate set and keep the order relatively unchanged both in candidate set and non-candidate set. Finally, we record the rank of the origin correct head entity. Another is to evaluate the tail entity prediction in the same way. The Mean Rank is calculated by average the head rank and the tail rank among all test triples we get above.

The Hits@10 evaluation also contains two parts, Hits@10 for head entities and Hits@10 of tail entities. As long as the correct head entity ranks in top 10, there will be a hit count. The Hits@10 for head entities will be the overall hit rate among all test triples. Similarly, the Hits@10 for tail entities is calculated in the same way, and the whole Hits@10 will be the average of Hits@10 for head entities and tail.

Given that there exist one-to-many and many-to-one structures, it's alright that other correct fake triples rank ahead of the ground truth in the test set. We may call these correct fake triples corrupted triples, and the corrupted triples should be count as positive. To eliminate this issue, we subtract the number of corrupted triplets included in the train, valid and test sets ranking ahead of the correct entity. We denote this evaluation setting as "Filter" and the original setting as "Raw".

Datasets	#WN11	#FB13
SE	53.0	75.2
SME(bilinear)	70.0	63.7
SLM	69.9	85.3
LFM	73.8	84.3
NTN	70.4	87.1
TransE(unif/bern)	75.9/75.9	70.9/81.5
TransH(unif/bern)	77.7/78.8	76.5/83.3
TransR(unif/bern)	85.5/85.9	74.7/82.5
CTransR(bern)	85.7	-
TransD(unif/bern)	85.6/86.4	85.9/89.1
TranSparse(share, S, unif/bern)	86.2/86.3	85.5/87.8
TranSparse(share, US, unif/bern)	86.3/86.3	85.3/87.7
TranSparse(separate, S, unif/bern)	86.2/86.4	86.7/88.2
TranSparse(separate, US, unif/bern)	86.8/86.8	86.5/87.5
TranAt(bern)	**88.2**	**89.1**

Table 3: Experimental results of triplet classification.

We conduct our experiments using two datasets: WN18 and FB15k, and compare with all the methods mentioned in section "Related Work". We select the margin γ among $\{1, 2, 4, 6\}$, the learning rate lr_rate for SGD among $\{0.01, 0.001\}$, the dimension of vectors k among $\{50, 100, 200\}$, the loss weight α among $\{0.1,1,10\}$. According to results of [Wang *et al.*, 2014; Lin *et al.*, 2015; Zhang, 2017], we just adjust mini-batch size to make each epoch has 100 mini-batch and run training process with 1000 epochs. We reset attention every 100 epochs. The cluster number of Kmeans c is set according to the number of relations. We choose the optimal configuration according to Mean Rank on valid set and they are as follows: $\gamma = 6, lr\_rate = 0.01, k = 100, \alpha = 1, c = 10$ on WN18 and $\gamma = 2, lr\_rate = 0.001, k = 200, \alpha = 1, c = 200$ on FB15k.

Our experimental results on WN18 and FB15k are shown in Table 2. Our methods have the best Mean Rank performance on both two datasets. And on WN18, our approaches significantly improve all the metrics. Owing to our attention mechanism, we find that our method has bigger γ and k in the optimal configuration. Without an efficient attention mechanism, k should be small since the accumulation of subtle changes from unrelated dimensions can not be ignored in score function. After adopting an efficient attention mechanism, the model can use large γ to fine adjust related dimensions. Finally, the asymmetric version of method outperforms the symmetrical one.

4.3 Triple Classification

Triplet classification aims to make a judgment on the correctness of a triplet (**h**, **r**, **t**), which is a binary classification task. In this paper, we use two datasets WN11 and FB13 to evaluate our models. The test sets of WN11 and FB13 provided by [Socher *et al.*, 2013] contain positive and negative triplets. For triplet classification task, we need to learn a threshold θ_r for each relation **r**. When judging a triplet is correct or not, we just calculate the translation distance between the head entity and tail entity with respect to given relation as the score. The score function is Equation 1. Following the previous works, we learn the θ_r by maximizing the classification accuracy on

Datasets	#WN18				#FB15k			
Metric	Mean Rank		Hits@10		Mean Rank		Hits@10	
	Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
SE	1,011	985	68.5	80.5	273	162	28.8	39.8
SME (linear/bilinear)	545/526	533/509	65.1/54.7	74.1/61.3	274/284	154/158	30.7/31.3	40.8/41.3
LFM	469	456	71.4	81.6	283	164	26.0	33.1
GAKE	-	-	-	-	228	119	44.5	64.8
TransE	263	251	75.4	89.2	243	125	34.9	47.1
TransH (unif/bern)	318/401	303/388	75.4/73.0	86.7/82.3	211/212	84/87	42.5/45.7	58.5/64.4
TransR (unif/bern)	232/238	219/225	78.3/79.8	91.7/92.0	226/198	78/77	43.8/48.2	65.5/68.7
CTransR (unif/bern)	243/231	230/218	78.9/79.4	92.3/92.3	233/199	82/75	44.0/48.4	66.3/70.2
TransD (unif/bern)	242/224	229/212	79.2/79.6	92.5/92.2	211/194	67/91	49.4/53.4	74.2/77.3
TranSparse (share, S, unif/bern)	248/237	236/224	79.7/80.4	93.5/93.6	226/194	95/88	48.8/53.4	73.4/77.7
TranSparse (share, US, unif/bern)	242/233	229/221	79.8/80.5	93.7/93.9	231/191	101/86	48.9/53.5	73.5/78.3
TranSparse (separate, S, unif/bern)	235/224	223/221	79.0/79.8	92.3/92.8	211/187	**63**/82	50.1/53.3	77.9/79.5
TranSparse (separate, US, unif/bern)	233/223	221/211	79.6/80.1	93.4/93.2	216/190	66/82	50.3/**53.7**	78.4/**79.9**
TransAt (bern)	214	202	**81.4**	**95.1**	187	83	52.6	78.1
TransAt (asy,bern)	**169**	**157**	**81.4**	95.0	**185**	82	52.9	78.2

Table 2: Experimental results of link prediction.

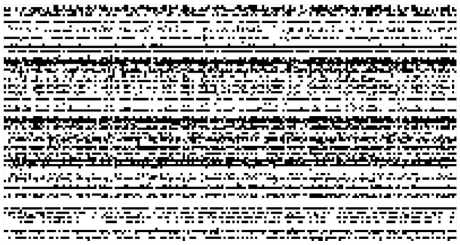

Figure 3: Attention vectors learned by TransAt. These are some randomly selected attention vectors learned by TransAt on FB15k. Multiplying each entry of vectors with 255 and concatenating these vectors together (each row is an attention vector), we show them as a gray scaleimage (white has the grayscale value 255). This figure demonstrates that our method can truly pay attention to a part of the dimensions of some relations.

the valid set. For a given triplet (**h**, **r**, **t**), if its score is lower than θ_r and both the head entity and tail entity are in respective candidate sets, it will be classified as positive, otherwise, negative.

In this experiments, we select the margin γ among $\{1, 2, 4, 6\}$, the learning rate lr_rate for SGD among $\{0.01, 0.001\}$, the dimension of vectors k among $\{50, 100, 200\}$, the loss weight α among $\{0.1,1,10\}$. We adjust mini-batch size to make each epoch has 100 mini-batch and run training process with 1000 epochs. We reset attention every 100 epochs. The cluster number of Kmeans c is set according to the number of relations. We still choose the optimal configuration based on performance on valid set. The best configuration: $\gamma = 6, lr\_rate = 0.01, k = 50, \alpha = 1, c = 10$ on WN11 and $\gamma = 2, lr\_rate = 0.001, k = 200, \alpha = 1, c = 200$ on FB13.

Our experimental results on WN11 and FB13 are shown in Table 3. Our methods have the best classification performance on both two datasets. This means our approach could find the right candidate sets with high probability. Otherwise, we will obtain decidedly worse results since we merely think the triples with entities not in candidate sets are negative.

4.4 Attention Mechanism

In this section, we show our most crucial experimental result, and it can emphatically prove that our attention mechanism works. We stress again that we believe there exists hierarchical structure among the attributes of an entity, and human cognition of an entity follows a hierarchical routine. When predicting whether a relation holds between two entities, people first check the category of entities, then they focus on fined-grained relation-related attributes to make the decision. This means, in our methods, some attention vectors $\mathbf{a}_r$ should be sparse. Figure 3 shows some attention vectors $\mathbf{a}_r$ learned by TransAt. They are randomly selected attention vectors learned by TransAt on FB15k. Multiplying each entry of vectors with 255 and concatenating these vectors together (each row is an attention vector), we show them as a grayscale image (white has the grayscale value 255). Figure 3 demonstrates that our method can truly pay attention to a part of the dimensions of some relations.

5 Conclusion and Future Work

In this paper, we focus on the translation-based embedding methods for the task of knowledge graph completion, which can embed entities and relations in knowledge graphs to continuous vector space. We find that previous methods fail to learn attention. Thus we introduce a model named TransAt to solve it. Specifically, we notice that there exists a hierarchical structure among the attributes in an entity and human cognition of an entity follows a hierarchical routine. Thus we propose a two stage discriminative method to achieve attention mechanism. Our experiments show that TranAt indeed pays attention to a part of the dimensions and outperforms all baseline models on triple classification and link prediction tasks. In the future, we will explore the better attention mechanism and try to use this method to solve some important problems in knowledge extraction[Guan *et al.*, 2015] and recommeder systems [Zhu *et al.*, 2017].

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China (Grant Nos: 61751307), in part by the grant ZJU Research 083650 of the ZJUI Research Program from Zhejiang University and in part by the National Youth Top-notch Talent Support Program. The experiments are supported by Chengwei Yao in the Experiment Center of

the College of Computer Science and Technology, Zhejiang University.

References

[Bakker, 1987] René Ronald Bakker. *Knowledge Graphs: representation and structuring of scientific knowledge*. 1987.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250. AcM, 2008.

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. Learning structured embeddings of knowledge bases. In *AAAI*, volume 6, page 6, 2011.

[Bordes *et al.*, 2012] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*, pages 127–135, 2012.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

[Bordes *et al.*, 2014] 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.

[Bottou, 2010] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. Springer, 2010.

[Dunteman, 1989] George H Dunteman. *Principal components analysis*. Number 69. Sage, 1989.

[Feng *et al.*, 2016] Jun Feng, Minlie Huang, Yang Yang, et al. Gake: Graph aware knowledge embedding. In *COLING*, pages 641–651, 2016.

[Guan *et al.*, 2015] Ziyu Guan, Shengqi Yang, Huan Sun, Mudhakar Srivatsa, and Xifeng Yan. Fine-grained knowledge sharing in collaborative environments. *TKDE*, 27(8):2163–2174, 2015.

[He *et al.*, 2015a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[He *et al.*, 2015b] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *CIKM*, pages 623–632. ACM, 2015.

[Jenatton *et al.*, 2012] Rodolphe Jenatton, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. A latent factor model for highly multi-relational data. In *NIPS*, pages 3167–3175, 2012.

[Ji *et al.*, 2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, pages 687–696, 2015.

[Ji *et al.*, 2016] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI*, pages 985–991, 2016.

[Kanojia *et al.*, 2017] Vibhor Kanojia, Hideyuki Maeda, Riku Togashi, and Sumio Fujita. Enhancing knowledge graph embedding with probabilistic negative sampling. In *WWW*, pages 801–802. International World Wide Web Conferences Steering Committee, 2017.

[Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

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

[Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934, 2013.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119, 2014.

[Wu *et al.*, 2017] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *arXiv*, 2017.

[Zhang, 2017] Wen Zhang. Knowledge graph embedding with diversity of structures. In *WWW*, pages 747–753. International World Wide Web Conferences Steering Committee, 2017.

[Zhu *et al.*, 2016] Yu Zhu, Ziyu Guan, Shulong Tan, Haifeng Liu, Deng Cai, and Xiaofei He. Heterogeneous hypergraph embedding for document recommendation. *Neurocomputing*, 216:150–162, 2016.

[Zhu *et al.*, 2017] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: modeling user behaviors by time-lstm. In *IJCAI*, pages 3602–3608, 2017.

[Zou *et al.*, 2013] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398, 2013.