Improving Knowledge Graph Embedding Using Affine Transformations of Entities Corresponding to Each Relation

Jinfa Yang, Yongjie Shi, Xin Tong, Robin Wang, Taiyan Chen, Xianghua Ying*
Key Laboratory of Machine Perception (MOE)
School of EECS, Peking University
{jinfayang, shiyongjie, xin_tong, robin_wang, chenty, xhying}@pku.edu.cn

Abstract

To find a suitable embedding for a knowledge graph remains a big challenge nowadays. By using previous knowledge graph embedding methods, every entity in a knowledge graph is usually represented as a k-dimensional vector. As we know, an affine transformation can be expressed in the form of a matrix multiplication followed by a translation vector. In this paper, we firstly utilize a set of affine transformations related to each relation to operate on entity vectors, and then these transformed vectors are used for performing embedding with previous methods. The main advantage of using affine transformations is their good geometry properties with interpretability. Our experimental results demonstrate that the proposed intuitive design with affine transformations provides a statistically significant increase in performance with adding a few extra processing steps or adding a limited number of additional variables. Taking TransE as an example, we employ the scale transformation (the special case of an affine transformation), and only introduce k additional variables for each relation. Surprisingly, it even outperforms RotatE to some extent on various data sets. We also introduce affine transformations into RotatE, Distmult and ComplEx, respectively, and each one outperforms its original method.

1 Introduction

Knowledge graphs are usually collections of factual triples—(head entity, relation, tail entity) also known as (subject, predicate, object), which represent human knowledge of the real world in a structured way. There are some outstanding knowledge graphs, such as WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015), YAGO (Suchanek et al., 2007). They have gained widespread attention from their successful usage in various applications, *e.g.*, question answering (Huang et al., 2019), natural language processing (Zhang et al., 2020a), recommendation systems (Zhou et al., 2020) ,*etc*.

Although millions of entities and billions of facts exist in the large-scale knowledge graphs, they still suffer from the incompleteness problem. Therefore, knowledge graph completion also known as link prediction which aims to predict missing links among entities based on the known triples has attracted much attention gradually. Recently, extensive studies have been done concerning knowledge graph embedding (Bordes et al., 2013; Yang et al., 2015; Dettmers et al., 2018). These methods represent entities and relations as low-dimensional vectors (or matrices, tensors, *etc*.), which not only preserve the semantic information of the knowledge graph, but also represent entities and relations in a fixed structure which is easier for machines' further processing. Therefore, apart from the link prediction task, knowledge graph embedding can also be used in various downstream tasks, such as triple classification (Nguyen et al., 2020), search personalization (Lu et al., 2020) and so on.

The success of existing knowledge graph embedding models heavily relies on their ability to model different types of the relations, such as symmetry/antisymmetry and composition. For example, TransE (Bordes et al., 2013), which represent relations as translations, can model the composition paterns. DistMult (Yang et al., 2015), which forces all relation embeddings to be diagonal matrices in bilinear model, can model the symmetry pattern. However, most models ignore the difference between single-relational and multi-relational triples.

Multi-relational triples are ubiquitous phenomena in knowledge graphs. For instance, WordNet (Miller, 1995) contains the entity {department_of_justice} with relations {_hypernym, _synset_domain_topic_of, _has_part}. Freebase

*Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 508–517
November 7–11, 2021.

(Bollacker et al., 2008) contains the entity {Bryan Singer} with relations {/film/director/film, /people/person/profession, /people/person/nationality, /people/person/place_of_birth and so on}. Different relations lead entities to different identities or concerns. Figure 1 briefly shows that multiple relations may have effects on the optimization of knowledge graph embedding models. We try to do some spatial transformations to make the entities contain the corresponding relation information, and help to distinguish the scenes of different relations. Although there exists similar works that project entities with each relation (Lin et al., 2015; Nguyen et al., 2016), they often require complex projection matrices, which lead to a large amount of calculation and are difficult to apply to other models. In addition, other than TransE series models, we also apply this transformation method to Bilinear Models similar to RESCAL (Nickel et al., 2011), and improve their performance obviously.

In this paper, we firstly utilize a set of affine transformations related to each relation to operate on entity vectors, and then these transformed vectors are used for performing embedding with previous methods like TransE (Lin et al., 2015), RotatE (Sun et al., 2019), DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016). All of these applications are correspondingly simplified based on different model structures. Our experimental results demonstrate that the proposed intuitive design with affine transformations provides a statistically significant increase in performance with adding a few extra processing steps or adding a limited number of additional variables. Taking TransE as an example, we employ the scale transformation (the special case of an affine transformation), and only introduce k additional variables for each relation. Surprisingly, it even outperforms RotatE to some extent on various data sets. The application in other models also shows better results than their original models. Especially for DistMult and ComplEx, experiments on three benchmark data sets show that the proposed affine-transformation-based algorithms outperform several other state-of-the-art algorithms.

Notations. Throughout this paper, we use lowercase letters e, h, r, and t to represent entities, head entities, relations, and tail entities, respectively. The triplet (h, r, t) denotes a fact in knowledge graphs. The corresponding boldface lower-case letters $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ denote the embeddings (vectors) of head entities, relations, and tail entities. d and k are the dimensionality of entity and relation embedding space, respectively (usually $d = k$).

2 Related Work

In this section, we briefly review the related work. Roughly speaking, the existing knowledge graph embedding models are mainly divided into three categories: translational models, bilinear models and deep learning models. Table 1 summarizes different score functions $f_r(\mathbf{h}, \mathbf{t})$ from previous state-of-the-art methods.

Translational models. TransE is the first link prediction model to propose translation distance constraints, which supposes that entities and relations satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$, and defines the score function as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$. TransH (Wang et al., 2014) is proposed to compensate for the shortcomings of transE. They find that TransE cannot handle 1-N, N-1, N-N relations well. TransH projects entities onto relation-specific hyperplanes with $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r$, and the score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2$. Moreover, RotatE (Sun et al., 2019) defines each relation as a rotation from the source entity to the target entity in a complex vector space, which is able to represent various relation patterns including symmetry/asymmetry, inversion and composition. Then QuatE (Zhang et al., 2019) represents entities and relations with Quaternion; HAKE Zhang et al. (2020b) considers the hierarchical of relations, and both of them achieved impressive results.

Bilinear models. RESCAL (Nickel et al., 2011) represents each relation as a full rank matrix and defines a bilinear function as score function $f_r(\mathbf{h}, \mathbf{t}) = \langle \mathbf{h}^\top \mathbf{M}_r \mathbf{t} \rangle$. Although the embedded relations have a large number of parameters, RESCAL can still get good results through some of the latest training methods (Ruffinelli et al., 2019). Subsequently, DistMult (Yang et al., 2015) forces all relation embeddings $\mathbf{M}_r$ to be diagonal matrices, which can reduce the space of parameters and result in an easier model to be trained. However, Distmult assumes that all relations are symmetric, and is not friendly to other types of relations, such as antisymmetry, composition. To solve this problem, ComplEx (Trouillon et al., 2016) extends DistMult to complex space: $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$, and uses conjugate-transpose $\bar{\mathbf{t}}$ to model asymmetric relations.

Deep learning models. MLP (Dong et al.,

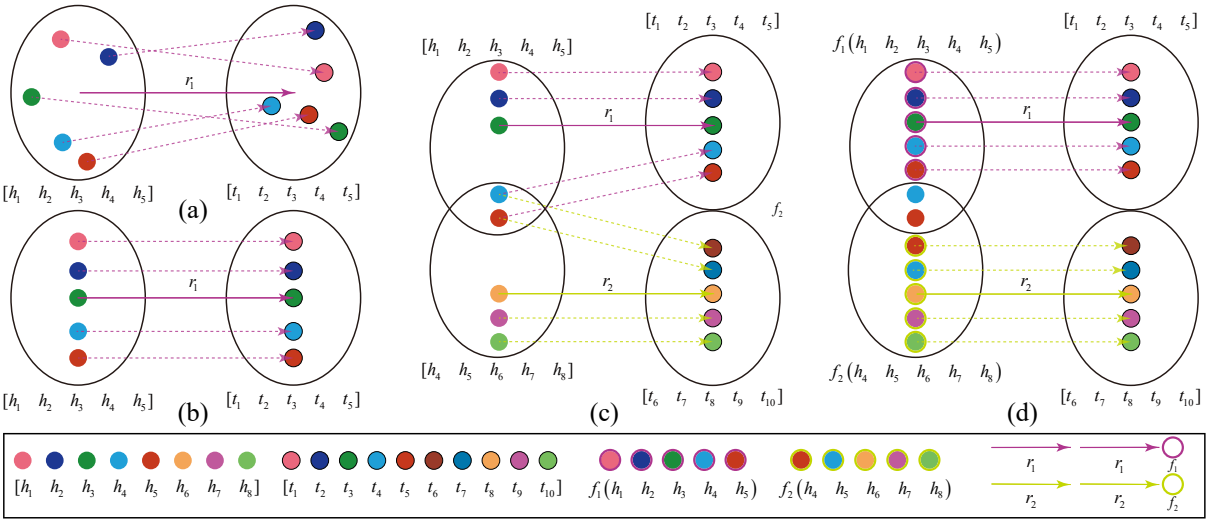

Figure 1: Illustration of the simplified optimization process of knowledge graph embedding. We expect the dashed arrow lines to be consistent with the solid arrow lines after optimization. (a) shows five sets of identical relation triples randomly initialized. (b) is the ideal optimization result. The same relation arrow lines should be consistent. (c) shows the optimization results of triples with two relations. Obviously, multiple relations affect the same relation arrow lines to be consistent. (d) shows the optimization result we expect after affine transformation.

2014) and NTN (Socher et al., 2013) use a fully connected neural network to calculate the scores of given triples. ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018) and ConvR (Jiang et al., 2019) employ convolutional neural networks to build score functions. There are also graph convolutional networks (Schlichtkrull et al., 2018) and recurrent neural networks RSN (Guo et al., 2019) which show promising performances.

3 Embedding with Affine Transformation

In this section, we briefly introduce affine transformation at first. Then we introduce our proposed method which utilizes affine transformation in TransE, RotatE, DistMult and ComplEx, respectively.

3.1 Affine Transformation

Consider a data set of k dimensional points $\{x_i\}$. We wish to learn a $k \times k$ linear transformation matrix $\mathbf{A}$ and a translation vector $\mathbf{b}$ which will help to find better embedding of the original data points. In general, an affine transformation is composed of linear transformations (dilation, reflection, rotation, scaling or shear) and a translation (or "shift"). In addition the affine transformation preserves collinearity and ratios of distances. In this regard, we perform affine transformation on the head entities and tail entities according to the corresponding relations:

$$\begin{cases} \mathbf{h}' = \mathbf{A}_r\mathbf{h} + \mathbf{b}_r \\ \mathbf{t}' = \mathbf{C}_r\mathbf{t} + \mathbf{d}_r, \end{cases} \tag{1}$$

where $\mathbf{A}_r, \mathbf{C}_r \in \mathbb{R}^{k\times k}$ and $\mathbf{b}_r, \mathbf{d}_r \in \mathbb{R}^k$ are the head entity and tail entity affine transformation parameters, respectively.

3.2 Improving TransE with AT

For TransE + AT (affine transformation), the expected distance relationship after affine transformation can be expressed as

$$\mathbf{t}' = \mathbf{h}' + \mathbf{r}. \tag{2}$$

Substituting Equation (1) into Equation (2), we can obtain

$$\mathbf{C}_r\mathbf{t} + \mathbf{d}_r = \mathbf{A}_r\mathbf{h} + \mathbf{b}_r + \mathbf{r}. \tag{3}$$

We further simplify Equation (3) as

$$\mathbf{t} = \mathbf{C}_r^{-1}\mathbf{A}_r\mathbf{h} + \mathbf{C}_r^{-1}(\mathbf{b}_r + \mathbf{r} - \mathbf{d}_r). \tag{4}$$

Since $\mathbf{C}_r^{-1}$ and $\mathbf{A}_r$ are also transformation matrices about $\mathbf{r}$, and the effect of $\mathbf{C}_r^{-1}$ on the product of $\mathbf{h}$ can be absorbed by $\mathbf{A}_r$, we denote $\mathbf{A}'_r$ as $\mathbf{C}_r^{-1}\mathbf{A}_r$. Similarly, denote $\mathbf{r}'$ as $\mathbf{C}_r^{-1}(\mathbf{b}_r + \mathbf{r} - \mathbf{d}_r)$. In fact, the symbolic representations of $\mathbf{A}'_r, \mathbf{r}'$ and $\mathbf{A}_r, \mathbf{r}$ are only used to distinguish the changes, and we still

Model	Score Function $f_r(\mathbf{h}, \mathbf{t})$	Parameters
TransE (Bordes et al., 2013)	$-\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransH (Wang et al., 2014)	$-\|(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r)\|_2^2$	$\mathbf{h}, \mathbf{t}, \mathbf{w}_r, \mathbf{d}_r \in \mathbb{R}^k$
TransR (Lin et al., 2015)	$-\|\mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t}\|_2$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d, \mathbf{r} \in \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k \times d}$
RotatE (Sun et al., 2019)	$-\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_1$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k, \|r_i\| = 1$
RESCAL (Nickel et al., 2011)	$\langle \mathbf{h}^\top \mathbf{M}_r \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k \times k}$
DistMult (Yang et al., 2015)	$\langle \mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}\left(\langle \mathbf{h}^\top \mathrm{diag}(\mathbf{r}) \bar{\mathbf{t}} \rangle\right)$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$
ConvE (Dettmers et al., 2018)	$\langle \sigma(\mathrm{vec}(\sigma([\bar{\mathbf{h}}, \bar{\mathbf{t}}] * \omega))\mathbf{W})\mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransE + AT	$\|\mathrm{diag}(\mathbf{a}_r)\mathbf{h} + \mathbf{r} - \mathbf{t}\|_1$	$\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{a}_r \in \mathbb{R}^k$
DistMult + AT	$\langle (\mathbf{h} + \mathbf{b}_r)^\top \mathrm{diag}(\mathbf{r})(\mathbf{t} + \mathbf{d}_r) \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{b}_r, \mathbf{d}_r \in \mathbb{R}^k$
ComplEx + AT	$\mathrm{Re}\left(\langle (\mathbf{h} + \mathbf{b}_r)^\top \mathrm{diag}(\mathbf{r})(\bar{\mathbf{t}} + \mathbf{d}_r) \rangle\right)$	$\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{b}_r, \mathbf{d}_r \in \mathbb{C}^k$
RotatE + AT	$\|\mathrm{diag}(\mathbf{a}_r)\mathbf{h} \circ \mathbf{r} + \mathbf{b}_r - \mathbf{t}\|_1$	$\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{b}_r \in \mathbb{C}^k, \mathbf{a}_r \in \mathbb{R}^k$

Table 1: Details of several knowledge graph embedding models, where $\langle \cdot \rangle$ denotes the generalized dot product, $\circ$ denotes the Hadamard product, σ denotes activation function, $*$ denotes 2D convolution, $\bar{\cdot}$ denotes conjugate for complex vectors and 2D reshape for real vectors in ConvE model.

use $\mathbf{A}_r, \mathbf{r}$ to represent in the following equation. Then we can get

$$\mathbf{t} = \mathbf{A}_r \mathbf{h} + \mathbf{r}. \tag{5}$$

In experiments, using full matrices $\mathbf{A}_r \mathbf{h}$ may cause parameter redundancy and overfitting. Therefore, we refer to DistMult (Yang et al., 2015) to take the diagonal parameters of the full matrix and mark it as $\mathrm{diag}(\mathbf{a}_r)$. And a simplified equation is obtained

$$\mathbf{t} = \mathrm{diag}(\mathbf{a}_r)\mathbf{h} + \mathbf{r}. \tag{6}$$

Then the corresponding score function of TransE + AT can be expressed as

$$f_r(\mathbf{h}, \mathbf{t}) = \|\mathrm{diag}(\mathbf{a}_r)\mathbf{h} + \mathbf{r} - \mathbf{t}\|_1, \tag{7}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{a}_r \in \mathbb{R}^k$.

The simplified model of TransE + AT accidentally obtains a score function similar to MuRE (Balazevic et al., 2019), The scoring function of MuRE is

$$f(\mathbf{h}, \mathbf{t}) = -\mathrm{d}(\mathbf{R}\mathbf{h}, \mathbf{t} + \mathbf{r})^2 + \mathrm{b_s} + \mathrm{b_o}, \tag{8}$$

where d is a distance function, $\mathbf{R}$ is a diagonal relation matrix, $\mathrm{b_s}$ and $\mathrm{b_o}$ are constants. Internally, MuRE ($\mathbf{R}\mathbf{h} - \mathbf{r} - \mathbf{t}$) (Balazevic et al., 2019), TransE+AT ($\mathrm{diag}(\mathbf{a}_r)\mathbf{h} + \mathbf{r} - \mathbf{t}$) are very similar, but MuRE calculates the square of the distance and there are two deviation terms, so the two are not totally the same.

The scoring function of the unsimplified version of TransE + AT can be expressed as

$$f_r(\mathbf{h}, \mathbf{t}) = \|(\mathbf{A}_r \mathbf{h} + \mathbf{b}_r) + \mathbf{r} - (\mathbf{C}_r \mathbf{t} + \mathbf{d}_r)\|_1. \tag{9}$$

Compared with other models based on relational transformation to improve TransE, such as TransH (Wang et al., 2014) and TransR (Lin et al., 2015) (refer to Table 1 for the scoring functions). TransH projects the entity onto the hyperplane where the relation $\mathbf{r} \in \mathbb{R}^k$ is located, and TransR transform the entity based on the relation-specified matrix $\mathbf{M}_r \in \mathbb{R}^{k \times k}$. The entity of TransR has a larger transformation range than that of TransH, so it can be understood that TransH is a special case of TransR. When $\mathbf{A}_r = \mathbf{C}_r$ and $\mathbf{b}_r = \mathbf{d}_r = 0$, the original TransE + AT is equivalent to TransR. That is, TransR is a special case of TransE + AT, and we simplify TransE + AT on this basis.

3.3 Improving RotatE with AT

For RotatE + AT, the expected rotation relationship after affine transformation can be expressed as

$$\mathbf{t}' = \mathbf{h}' \circ \mathbf{r}. \tag{10}$$

Substituting Equation (1) into Equation (10), we can obtain

$$\mathbf{C}_r \mathbf{t} + \mathbf{d}_r = (\mathbf{A}_r \mathbf{h} + \mathbf{b}_r) \circ \mathbf{r}. \tag{11}$$

We further simplify Equation (11) as

$$\mathbf{t} = \mathbf{C}_r^{-1}((\mathbf{A}_r \mathbf{h}) \circ \mathbf{r}) + \mathbf{C}_r^{-1}(\mathbf{b}_r \circ \mathbf{r} - \mathbf{d}_r). \tag{12}$$

We simplify $\mathbf{C}_r^{-1}((\mathbf{A}_r\mathbf{h}) \circ \mathbf{r})$ as $\text{diag}(\mathbf{a}'_r)\mathbf{h} \circ \mathbf{r}$ to represent scale transformation, and denote $\mathbf{b}'_r$ as $\mathbf{C}_r^{-1}(\mathbf{b}_r \circ \mathbf{r} - \mathbf{d}_r)$. Again, we use $\mathbf{a}_r, \mathbf{b}_r$ to represent $\mathbf{a}'_r, \mathbf{b}'_r$ in the following equation. And we can obtain

$$\mathbf{t} = \text{diag}(\mathbf{a}_r)\mathbf{h} \circ \mathbf{r} + \mathbf{b}_r. \tag{13}$$

Then the corresponding score function of RotatE + AT can be expressed as

$$f_r(\mathbf{h}, \mathbf{t}) = \|\text{diag}(\mathbf{a}_r)\mathbf{h} \circ \mathbf{r} + \mathbf{b}_r - \mathbf{t}\|_1, \tag{14}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{b}_r \in \mathbb{C}^k, \mathbf{a}_r \in \mathbb{R}^k$.

3.4 Improving DistMult and ComplEx with AT

Since the loss functions of RESCAL, DistMult and ComplEx have similar structures, we use RESCAL + AT to show the application process. For RESCAL + AT, the expected score function after affine transformation can be expressed as

$$f_r(\mathbf{h}, \mathbf{t}) = \left\langle \mathbf{h}'^{\top} \mathbf{M}_r \mathbf{t}' \right\rangle. \tag{15}$$

Substituting Equation (1) into Equation (15), we can obtain

$$f_r(\mathbf{h}, \mathbf{t}) = \left\langle (\mathbf{A}_r\mathbf{h} + \mathbf{b}_r)^{\top} \mathbf{M}_r (\mathbf{C}_r\mathbf{t} + \mathbf{d}_r) \right\rangle. \tag{16}$$

We further simplify Equation (16) as

$$f_r(\mathbf{h}, \mathbf{t}) = \left\langle (\mathbf{h} + \mathbf{A}_r^{-1}\mathbf{b}_r)^{\top} \mathbf{A}_r^{\top} \mathbf{M}_r \mathbf{C}_r (\mathbf{t} + \mathbf{C}_r^{-1}\mathbf{d}_r) \right\rangle. \tag{17}$$

Here, we denote $\mathbf{b}'_r$ as $\mathbf{A}_r^{-1}\mathbf{b}_r$, $\mathbf{d}'_r$ as $\mathbf{C}_r^{-1}\mathbf{d}_r$ and $\mathbf{M}'_r$ as $\mathbf{A}_r^{\top}\mathbf{M}_r\mathbf{C}_r$. Also, we use $\mathbf{b}_r, \mathbf{d}_r$ and $\mathbf{M}_r$ to represent $\mathbf{b}'_r, \mathbf{d}'_r$ and $\mathbf{M}'_r$ in the following equation. Correspondingly, the score function of RESCAL + AT can be expressed as

$$f_r(\mathbf{h}, \mathbf{t}) = \left\langle (\mathbf{h} + \mathbf{b}_r)^{\top} \mathbf{M}_r (\mathbf{t} + \mathbf{d}_r) \right\rangle, \tag{18}$$

where $\mathbf{h}, \mathbf{t}, \mathbf{b}_r, \mathbf{d}_r \in \mathbb{R}^k, \mathbf{M}_r \in \mathbb{R}^{k \times k}$.

Similarly, for DistMult, the corresponding score function of DistMult + AT is

$$f_r(\mathbf{h}, \mathbf{t}) = \left\langle (\mathbf{h} + \mathbf{b}_r)^{\top} \text{diag}(\mathbf{r}) (\mathbf{t} + \mathbf{d}_r) \right\rangle, \tag{19}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{b}_r, \mathbf{d}_r \in \mathbb{R}^k$.

For ComplEx, the corresponding score function of ComplEx + AT is

$$f_r(\mathbf{h}, \mathbf{t}) = \text{Re}\left(\left\langle (\mathbf{h} + \mathbf{b}_r)^{\top} \text{diag}(\mathbf{r}) (\bar{\mathbf{t}} + \mathbf{d}_r) \right\rangle \right), \tag{20}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{b}_r, \mathbf{d}_r \in \mathbb{C}^k$.

4 Experiments

This section is organized as follows: Firstly, we introduce the experimental settings in detail. Secondly, we show the effectiveness of our proposed model on three benchmark datasets. Finally, we analyze the embeddings generated by TransE + AT, RotatE + AT, Dismult + AT and ComplEx + AT, and show the results of ablation studies and visualize some parameters of models.

4.1 Experimental Settings

We evaluate our proposed models on three commonly used knowledge graphs, which are statistically summarized in Table 2.

Dataset	#En	#Re	#train	#valid	#test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Table 2: Number of entities, relations, and observed triples in each split for three benchmarks.

- FB15k-237 (Toutanova and Chen, 2015) is a subset of FB15k (Bordes et al., 2013), where inverse relations are deleted. A large fraction of content in this knowledge graph describes facts about movies, actors, awards, sports, and sport teams.

- WN18RR (Dettmers et al., 2018)is a subset of WN18 (Bordes et al., 2013). The inverse relations are deleted. Most of the triples consist of hyponym and hypernym relations which make WN18RR tend to follow a strictly hierarchical structure.

- YAGO3-10 (Dettmers et al., 2018) is a subset of YAGO3 (Mahdisoltani et al., 2013) which has a minimum of 10 relations for each entity. Most of the triples deal with descriptive attributes of people, such as citizenship, gender, and profession.

As pointed out by Toutanova and Chen (2015) and Dettmers et al. (2018), FB15k, WN18 and YAGO3 suffer from the test leakage. This issue is primarily due to the presence of relations that are nearly identical or the inverse of one another. One can achieve the state-of-the-art results even using a simple rule-based model. Therefore, we use WN18RR, FB15k-237 and YAGO3-10 as the benchmark datasets.

Training Protocol. We use Adam (Kingma and Ba, 2014) as the optimizer and fine-tune the hyper-parameters on the validation dataset based on grid search. Both the entity and relation embeddings are uniformly initialized. For TransE + AT and RotatE + AT, we use self-adversarial negative sampling Sun et al. (2019) with margin λ. For DistMult + AT and ComplEx + AT, we use the "reciprocal" setting Lacroix et al. (2018) with N3 regularizers.

	WN18RR	FB15k-237	YAGO3-10
TransE	8.1908M	2.9556M	24.6438M
RotatE	16.3794M	5.8638M	49.2802M
DistMult	8.1908M	2.9556M	24.6438M
ComplEx	16.3816M	5.9112M	49.2876M
TransE + AT	8.1930M	3.0030M	24.6512M
RotatE + AT	16.3860M	6.0060M	49.3024M
DistMult + AT	8.1952M	3.0504M	24.6586M
ComplEx + AT	16.3904M	6.1008M	49.3172M
RESCAL	8.4086M	7.6482M	25.3764M
TransH	8.1930M	3.0030M	24.6512M
TransR	8.4086M	7.6482M	25.3764M
HAKE	16.3838M	5.9586M	49.295M
QuatE	32.7632M	11.8224M	98.5752M

Table 3: The number of parameters that different models need to learn on WN18RR, FB15k-237 and YAGO3-10 data sets. Here we assume that the dimension of the entities and relations embedding vector is 200.

Evaluation Protocol. For each triple (h, r, t) in the test dataset, we replace either the head entity h or the tail entity t with the total list of the embedding entities. Then we base the score function to rank the candidate entities in descending order. The filtered setting is used to remove some correct results that appear in the training set or validation set but not in test set. We choose Mean Reciprocal Rank (MRR) and Hits at N (H@N) as the evaluation metrics. Higher MRR or H@N indicates better performance.

Number of parameters. Table 3 shows the number of parameters that different models need to learn on WN18RR, FB15k-237 and YAGO3-10 data sets. Compared with the original models: TransE, RotatE, DistMult and ComplEx, our proposed TransE + AT, RotatE + AT, DistMult + AT and ComplEx + AT models only adds a small number of parameters. Especially for the WN18RR and YAGO3-10 data sets, the number of added parameters is almost negligible, but the final experimental results are significantly improved. TransH has the same number of parameters as TransE + AT, but needs more computing resources for Hyperplanes translating in both head entities and tail entities, and TransR needs more number of parameters and calculations for the matrix multiplication with $\mathbf{M}_r$. Compared with the recent state of art methods, *i.e.*, QuatE and HAKE, the number of parameters of TransE + AT and DistMult + AT are smaller than both, while ComplEx + AT and RotatE + AT are close to HAKE but smaller than QuatE, and our method also exceeds their results in some quality indexes.

4.2 Main Results

In this section, we compare the performance of affine transformation against several state-of-the-art Knowledge graph completion models on WN18RR, FB15k-237 and YAGO3-10 datasets, including TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019), MuRE (Balazevic et al., 2019), QuatE (Zhang et al., 2019), InteractE (Vashishth et al., 2020) and HAKE (Zhang et al., 2020b). In order to avoid the influence of negative sample sampling and other training strategies, We reimplement TransE using self-adversarial negative sampling Sun et al. (2019), DistMult and ComplEx using the "reciprocal" setting Kazemi and Poole (2018); Lacroix et al. (2018). Table 4 shows the effectiveness of affine transformation applied in TransE, RoataE, DistMult and ComplEx models.

For TransE + AT, compared with the retrained TransE, our results on the three data sets have an average MRR increase of 10.4%. Especially for the WN18RR data set, TransE + AT can handle symmetric relations, while WN18RR contains a large number of symmetric relations, so the result is significantly improved. For RotatE + AT, compared with the original RotatE, our results have an

	WN18RR				FB15k-237				YAGO3-10			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
MuRE	.475	.436	.487	.554	.336	.245	.370	.521	-	-	-	-
QuatE	.488	.438	.508	.582	.366	.271	.401	.556	-	-	-	-
InteractE	.463	.430	-	.528	.354	.263	-	.535	.541	.462	-	.687
HAKE	.497	.452	**.516**	.582	346	250	.381	.542	.545	.462	.596	.694
TransE	.222	.014	.399	.528	.330	.232	.369	.526	.510	.413	.574	.681
RotatE	.476	.428	.492	.571	.338	.241	.375	.533	.495	.402	.550	.670
DistMult	.455	.410	.467	.544	.358	.264	.395	.550	.566	.491	.608	.704
ComplEx	.489	.443	.502	.580	.365	.270	.403	.558	.577	.502	.621	.709
TransE + AT	.479	.434	.495	.571	.351	.257	.386	.538	.543	.462	.596	.690
RotatE + AT	.488	.438	.509	.583	.348	.253	.384	.537	.545	.459	.601	.701
DistMult + AT	.478	.433	.490	.563	**.372**	**.276**	**.412**	**.564**	**.584**	**.513**	.625	.709
ComplEx + AT	**.500**	**.455**	.514	**.592**	.369	.273	.407	.559	.582	.507	**.627**	**.712**

Table 4: Evaluation results on WN18RR, FB15k-237 and YAGO3-10 datasets. We reimplement TransE using self-adversarial negative sampling Sun et al. (2019), DistMult and ComplEx using the "reciprocal" setting Kazemi and Poole (2018); Lacroix et al. (2018), which leads to better results than the reported results in the original paper.

average MRR increase of 2.5% on the three data sets. Especially for the YAGO3-10 data set, RotatE + AT exceeds the retrained TransE and closes to HAKE.

For DistMult + AT and ComplEx + AT, We creatively introduce the translation component into the bilinear model. Interestingly, this kind of application works and makes improvements then the original models. Compared with the retrained DistMult, our results of DistMult + AT on the three data sets have an average MRR increase of 1.8%. Similarly, compare with the retrained ComplEx, our results of ComplEx + AT on the three data sets have an average MRR increase of 0.7%. In three data sets, DistMult + AT and ComplEx + AT exceed other affine transformation methods, and mostly outperform MuRE, QuatE, InteractE and HAKE, reaching the state of art results.

4.3 Ablation Studies

In this section, we conduct ablation studies on different models. Based on the structural differences of models, we split the affine transformation into different combinations, including only make affine transformation on head entities AT_h and only make affine transformation on tail entities AT_t; only keep the scale parameter of affine transformation AT_scale and only keep the translation parameters of the affine transformation AT_trans. For DistMult + AT and ComplEx + AT, we choose the first combination as it can easily split the affine transformation of the head and tail entities. And we chose the second combination for RotatE.

From Table 5, we can see that for most models, better results are obtained by using a complete affine transformation. There are some results where H@10 is higher than the final models, such as RotatE + AT_scale gains a 0.2% higher H@10 than RotatE + AT on the FB15k-237 data set, DistMult + AT_t gains a 0.1% higher H@10 than DistMult + AT on the YAGO-10 data set. We infer that the use of a complete affine transformation will have stronger constraints, which makes the accurate prediction H@1 higher, while the rough prediction H@10 decreases. On the contrary, under weak constraints, the accurate prediction H@1 will be lower, while the rough prediction H@10 will increase.

4.4 Visualize Embedded Parameters

In this part, we visualize the some instances of TransE + AT, RotatE + AT, DisMult + AT and ComplEx + AT models on three data sets. Refer to Sun et al. (2019), we display the histogram of the k-dimensional embedding vector with different relations.

The first column is a symmetry relations {_similar_to, _derivationally_related_form}. From Figure 2 we can see that the parameter of TransE + AT diag($\mathbf{a}_r$) can help TransE deal with symmetric relations. For RotatE, the value of diag($\mathbf{a}_r$) is 1 or -1, while the value of $\mathbf{b}_r$ is close to zero for the symmetric relations. For DistMult + AT and ComplEx + AT, the $\mathbf{b}_r$ and $\mathbf{d}_r$ parameters may affect the model's representation of the symmetric relations, so their values here are close to zero.

For other visualization of models' instances, such as the last four columns of TransE + AT, we use similar relationships {/film/film/genre,

	WN18RR				FB15k-237				YAGO3-10			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
RotatE + AT_scale	.474	.426	.497	.567	.342	.243	.381	**.539**	.538	.449	.596	.694
RotatE + AT_trans	.480	.434	.497	.577	.346	.250	.381	.534	.535	.448	.591	.693
RotatE + AT	**.485**	**.436**	**.505**	**.581**	**.348**	**.253**	**.384**	.537	**.545**	**.459**	**.601**	**.701**
DistMult + AT_h	.475	.431	.487	.562	.368	.272	.407	.562	.582	.511	**.626**	.709
DistMult + AT_t	.459	.416	.473	.545	.369	.274	.406	.561	.582	.511	.625	**.710**
DistMult + AT	**.478**	**.433**	**.490**	**.563**	**.372**	**.276**	**.412**	**.564**	**.584**	**.513**	.625	.709
ComplEx + AT_h	.498	.454	.513	.588	.368	.272	.406	**.561**	.580	.505	.625	**.713**
ComplEx + AT_t	.495	.451	.508	.585	.367	.271	.404	.558	.579	.504	.626	.712
ComplEx + AT	**.500**	**.455**	**.514**	**.592**	**.369**	**.273**	**.407**	.559	**.582**	**.507**	**.627**	.712

Table 5: Ablation results on WN18RR, FB15k-237 and YAGO3-10 datasets. The symbols AT_scale and AT_trans represent only keep the scale parameter of affine transformation and only keep the translation parameters of the affine transformation, respectively; AT_h and AT_t represent only make affine transformation on head entities and only make affine transformation on tail entities, respectively.

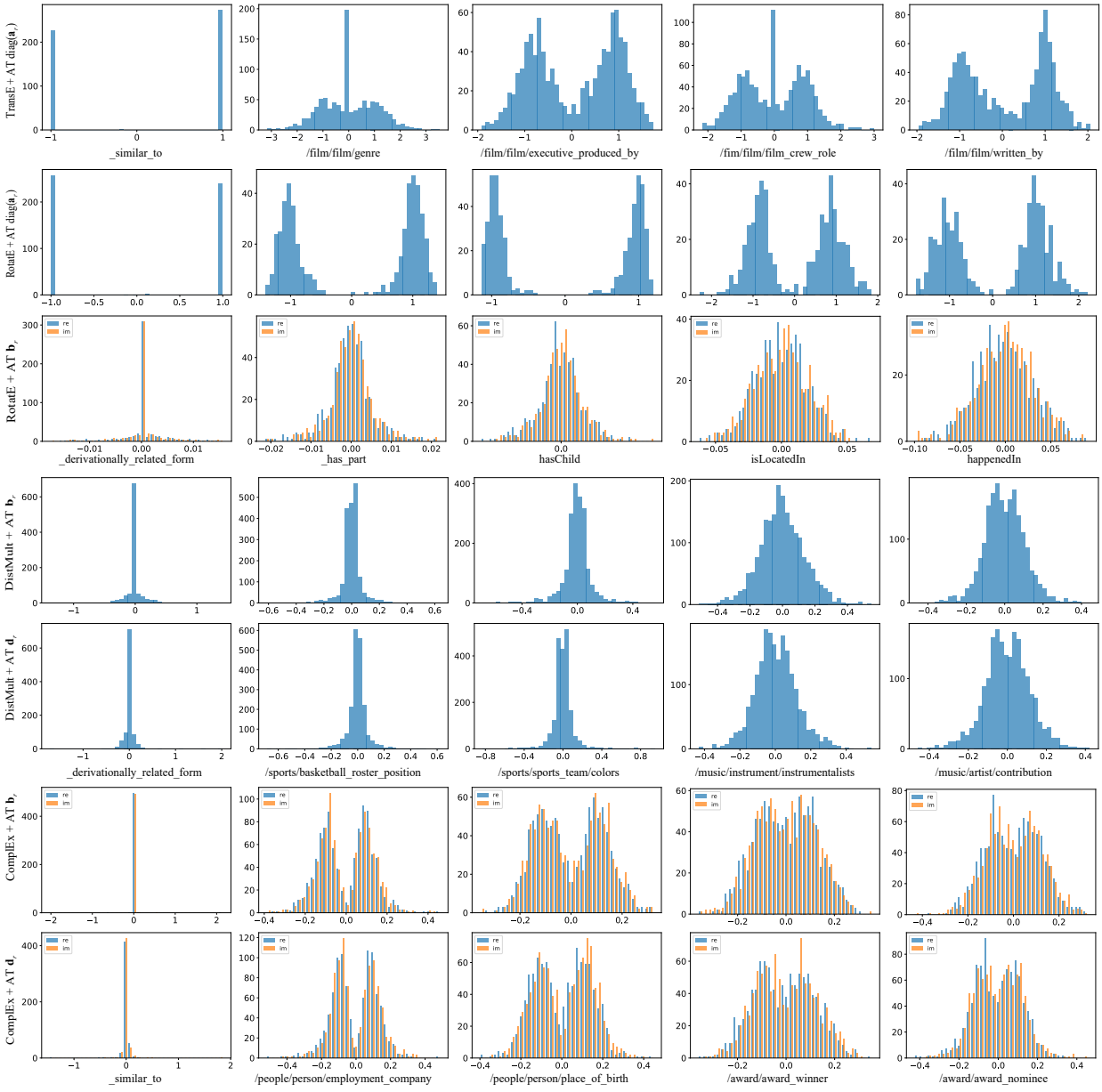

Figure 2: Visualization of some instances of TransE + AT, RotatE + AT, DistMult + AT and ComplEx + AT on WN18RR, FB15k-237 and YAGO3-10 data sets

/film/film/executive_produced_by, /fim/film/film _crew_role, /film/film/written_by}, they show a certain difference, and the last three related to people are more similar; In DistMult + AT and ComplEx + AT, we choose two similar relations to form different groups. The results show that different relationships have large differences in histograms, while similar relationships have smaller differences; similar phenomena also appears in the RotatE + AT.

5 Conclusion

We propose a novel knowledge graph embedding approach which firstly introduces a parametric mapping that projects entity vectors into a new space by an affine transformation corresponding to each relation, and then employs previous embedding methods that map the entities and relations into the embedding space. This algorithm enforces the embedding to be approximately uniformly distributed around the original entity vectors by adjusting the scaling and translation parameters of the affine transformation, which requires considerably less additional computational effort. Extensive experimental results show that the affine-transformation-based algorithms outperform the original TransE, RotatE, Distmult and ComplEx, respectively. Experiments on three benchmark data sets also show that the proposed affine-transformation-based algorithms outperform several other state-of-the-art algorithms in some quality indexes. We believe that knowledge graph embedding based on affine transformations is very promising and has the potential of being used for many applications. However, more comparison with other embedding methods are needed to fully understand its advantages and disadvantages.

Acknowledgement

This work was supported in part by State Key Development Program Grand No. 2020YFB1708002, and NNSFC Grant No. 61971008.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. In *Proc. of Annual Conference on Neural Information Processing Systems*, volume 32, pages 4463–4473.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of ACM International Conference on Management of Data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proc. of Annual Conference on Neural Information Processing Systems*, pages 1–9.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proc. of AAAI Conference on Artificial Intelligence*, volume 32, pages 1811–1818.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proc. of ACM International Conference on Knowledge Discovery and Data Mining*, pages 601–610.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proc. of International Conference on Machine Learning*, pages 2505–2514.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proc. of ACM International Conference on Web Search and Data Mining*, pages 105–113.

Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. Adaptive convolution for multi-relational learning. In *Proc. of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 978–987.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Proc. of Annual Conference on Neural Information Processing Systems*, pages 4289–4300.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *Proc. of International Conference on Machine Learning*, pages 2863–2872.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proc. of AAAI Conference on Artificial Intelligence*, volume 29, pages 2181–2187.

Shuqi Lu, Zhicheng Dou, Chenyan Xiong, Xiaojie Wang, and Ji-Rong Wen. 2020. Knowledge enhanced personalized search. In *Proc. of ACM International Conference on Research and Development in Information Retrieval*, pages 709–718.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *Proc. of Conference on Innovative Data Systems Research*.

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

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proc. of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 460–466.

Tu Nguyen, Dinh Phung, et al. 2020. A relational memory-based embedding model for triple classification and search personalization. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 3429–3435.

Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proc. of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–333.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proc. of International Conference on Machine Learning*, volume 11, pages 809–816.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2019. You can teach an old dog new tricks! on training knowledge graph embeddings. In *Proc. of International Conference on Learning Representations*, pages 1–20.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proc. of European Semantic Web Conference*, pages 593–607.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proc. of Annual Conference on Neural Information Processing Systems*, pages 926–934.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proc. of International Conference on World Wide Web*, pages 697–706.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proc. of International Conference on Learning Representations*, pages 1–18.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proc. of Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proc. of International Conference on Machine Learning*, volume 48, pages 2071–2080.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proc. of AAAI Conference on Artificial Intelligence*, volume 34, pages 3009–3016.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proc. of AAAI Conference on Artificial Intelligence*, volume 28, pages 1112–1119.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. of International Conference on Learning Representations*, pages 1–12.

Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. 2020a. Few-shot knowledge graph completion. In *Proc. of AAAI Conference on Artificial Intelligence*, volume 34, pages 3041–3048.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Proc. of Annual Conference on Neural Information Processing Systems*, volume 32, pages 2735–2745.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020b. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 3065–3072.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proc. of ACM International Conference on Knowledge Discovery and Data Mining*, pages 1006–1014.