Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction

Zhanqiu Zhang,[*] **Jianyu Cai,**[*] **Yongdong Zhang, Jie Wang**[†]
University of Science and Technology of China
{zzq96, jycai}@mail.ustc.edu.cn
{zhyd73, jiewangx}@ustc.edu.cn

Abstract

Knowledge graph embedding, which aims to represent entities and relations as low dimensional vectors (or matrices, tensors, etc.), has been shown to be a powerful technique for predicting missing links in knowledge graphs. Existing knowledge graph embedding models mainly focus on modeling relation patterns such as symmetry/antisymmetry, inversion, and composition. However, many existing approaches fail to model *semantic hierarchies*, which are common in real-world applications. To address this challenge, we propose a novel knowledge graph embedding model—namely, **H**ierarchy-**A**ware **K**nowledge Graph **E**mbedding (HAKE)—which maps entities into the polar coordinate system. HAKE is inspired by the fact that concentric circles in the polar coordinate system can naturally reflect the hierarchy. Specifically, the radial coordinate aims to model entities at different levels of the hierarchy, and entities with smaller radii are expected to be at higher levels; the angular coordinate aims to distinguish entities at the same level of the hierarchy, and these entities are expected to have roughly the same radii but different angles. Experiments demonstrate that HAKE can effectively model the semantic hierarchies in knowledge graphs, and significantly outperforms existing state-of-the-art methods on benchmark datasets for the link prediction task.

1 Introduction

Knowledge graphs are usually collections of factual triples—(head entity, relation, tail entity), which represent human knowledge in a structured way. In the past few years, we have witnessed the great achievement of knowledge graphs in many areas, such as natural language processing (Zhang et al. 2019), question answering (Huang et al. 2019), and recommendation systems (Wang et al. 2018).

Although commonly used knowledge graphs contain billions of triples, they still suffer from the incompleteness problem that a lot of valid triples are missing, as it is impractical to find all valid triples manually. Therefore, knowledge graph completion, also known as link prediction in knowledge graphs, has attracted much attention recently. Link prediction aims to automatically predict missing links between entities based on known links. It is a challenging task as we not only need to predict whether there is a relation between two entities, but also need to determine which relation it is.

Inspired by word embeddings (Mikolov et al. 2013) that can well capture semantic meaning of words, researchers turn to distributed representations of knowledge graphs (aka, knowledge graph embeddings) to deal with the link prediction problem. Knowledge graph embeddings regard entities and relations as low dimensional vectors (or matrices, tensors), which can be stored and computed efficiently. Moreover, like in the case of word embeddings, knowledge graph embeddings can preserve the semantics and inherent structures of entities and relations. Therefore, other than the link prediction task, knowledge graph embeddings can also be used in various downstream tasks, such as triple classification (Lin et al. 2015), relation inference (Guo, Sun, and Hu 2019), and search personalization (Nguyen et al. 2019).

The success of existing knowledge graph embedding models heavily relies on their ability to model connectivity patterns of the relations, such as symmetry/antisymmetry, inversion, and composition (Sun et al. 2019). For example, TransE (Bordes et al. 2013), which represent relations as translations, can model the inversion and composition patterns. DistMult (Yang et al. 2015), which models the three-way interactions between head entities, relations, and tail entities, can model the symmetry pattern. RotatE (Sun et al. 2019), which represents entities as points in a complex space and relations as rotations, can model relation patterns including symmetry/antisymmetry, inversion, and composition. However, many existing models fail to model *semantic hierarchies* in knowledge graphs.

Semantic hierarchy is a ubiquitous property in knowledge graphs. For instance, WordNet (Miller 1995) contains the triple [arbor/cassia/palm, hypernym, tree], where "tree" is at a higher level than "arbor/cassia/palm" in the hierarchy. Freebase (Bollacker et al. 2008) contains the triple [England, /location/location/contains, Pontefract/Lancaster], where "Pontefract/Lancaster" is at a lower level than "England" in the hierarchy. Although there exists some work that takes the hierarchy structures into account (Xie, Liu, and Sun 2016; Zhang et al. 2018), they usually re-

[*] Equal contribution.

[†] Corresponding author.

quire additional data or process to obtain the hierarchy information. Therefore, it is still challenging to find an approach that is capable of modeling the semantic hierarchy automatically and effectively.

In this paper, we propose a novel knowledge graph embedding model—namely, **H**ierarchy-**A**ware **K**nowledge Graph **E**mbedding (HAKE). To model the semantic hierarchies, HAKE is expected to distinguish entities in two categories: (a) at different levels of the hierarchy; (b) at the same level of the hierarchy. Inspired by the fact that entities that have the hierarchical properties can be viewed as a tree, we can use the depth of a node (entity) to model different levels of the hierarchy. Thus, we use modulus information to model entities in the category (a), as the size of moduli can reflect the depth. Under the above settings, entities in the category (b) will have roughly the same modulus, which is hard to distinguish. Inspired by the fact that the points on the same circle can have different phases, we use phase information to model entities in the category (b). Combining the modulus and phase information, HAKE maps entities into the polar coordinate system, where the radial coordinate corresponds to the modulus information and the angular coordinate corresponds to the phase information. Experiments show that our proposed HAKE model can not only clearly distinguish the semantic hierarchies of entities, but also significantly and consistently outperform several state-of-the-art methods on the benchmark datasets.

Notations Throughout this paper, we use lower-case letters h, r, and t to represent 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. The i-th entry of a vector $\mathbf{h}$ is denoted as $[\mathbf{h}]_i$. Let k denote the embedding dimension.

Let $\circ : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ denote the Hadamard product between two vectors, that is,

$$[\mathbf{a} \circ \mathbf{b}]_i = [\mathbf{a}]_i \cdot [\mathbf{b}]_i,$$

and $\|\cdot\|_1$, $\|\cdot\|_2$ denote the ℓ_1 and ℓ_2 norm, respectively.

2 Related Work

In this section, we will describe the related work and the key differences between them and our work in two aspects—the model category and the way to model hierarchy structures in knowledge graphs.

Model Category

Roughly speaking, we can divide knowledge graph embedding models into three categories—translational distance models, bilinear models, and neural network based models. Table 1 exhibits several popular models.

Translational distance models describe relations as translations from source entities to target entities. TransE (Bordes et al. 2013) supposes that entities and relations satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^n$, and defines the corresponding score function as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{1/2}$. However, TransE does not perform well on 1-N, N-1 and N-N relations (Wang et al. 2014). TransH (Wang et al. 2014) overcomes the many-to-many relation problem by allowing entities to have distinct representations given different relations. The score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2$, where $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ are the projections of entities onto relation-specific hyperplanes. ManifoldE (Xiao, Huang, and Zhu 2016) deals with many-to-many problems by relaxing the hypothesis $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ to $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2 \approx \theta_r^2$ for each valid triple. In this way, the candidate entities can lie on a manifold instead of exact point. The corresponding score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -(\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_2^2 - \theta_r^2)^2$. More recently, to better model symmetric and antisymmetric relations, RotatE (Sun et al. 2019) defines each relation as a rotation from source entities to target entities in a complex vector space. The score function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_1$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$ and $|[\mathbf{r}]_i| = 1$.

Bilinear models product-based score functions to match latent semantics of entities and relations embodied in their vector space representations. RESCAL (Nickel, Tresp, and Kriegel 2011) represents each relation as a full rank matrix, and defines the score function as $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$, which can also be seen as a bilinear function. As full rank matrices are prone to overfitting, recent works turn to make additional assumptions on $\mathbf{M}_r$. For example, DistMult (Yang et al. 2015) assumes $\mathbf{M}_r$ to be a diagonal matrix, and ANALOGY (Liu, Wu, and Yang 2017) supposes that $\mathbf{M}_r$ is normal. However, these over-simplified models are usually less expressive and not powerful enough for general knowledge graphs. Differently, ComplEx (Trouillon et al. 2016) extends DistMult by introducing complex-valued embeddings to better model asymmetric and inverse relations. HolE (Nickel, Rosasco, and Poggio 2016) combines the expressive power of RESCAL with the efficiency and simplicity of DistMult by using the circular correlation operation.

Neural network based models have received greater attention in recent years. For example, MLP (Dong et al. 2014) and NTN (Socher et al. 2013) use a fully connected neural network to determine the scores of given triples. ConvE (Dettmers et al. 2018) and ConvKB (Nguyen et al. 2018) employ convolutional neural networks to define score functions. Recently, graph convolutional networks are also introduced, as knowledge graphs obviously have graph structures (Schlichtkrull et al. 2018).

Our proposed model HAKE belongs to the translational distance models. More specifically, HAKE shares similarities with RotatE (Sun et al. 2019), in which the authors claim that they use both modulus and phase information. However, there exist two major differences between RotatE and HAKE. Detailed differences are as follows.

(a) The aims are different. RotatE aims to model the relation patterns including symmetry/antisymmetry, inversion, and composition. HAKE aims to model the semantic hierarchy, while it can also model all the relation patterns mentioned above.

(b) The ways to use modulus information are different. RotatE models relations as rotations in the complex space, which encourages two linked entities to have the same modulus, no matter what the relation is. The different moduli in RotatE come from the inaccuracy in training.

Table 1: Details of several knowledge graph embedding models, where $\circ$ denotes the Hadamard product, f denotes a activation function, $*$ denotes 2D convolution, and ω denotes a filter in convolutional layers. $\bar{\cdot}$ denotes conjugate for complex vectors in ComplEx model and 2D reshaping for real vectors in ConvE model.

Model	**Score Function** $f_r(\mathbf{h}, \mathbf{t})$	**Parameters**
TransE (Bordes et al. 2013)	$-\Vert\mathbf{h} + \mathbf{r} - \mathbf{t}\Vert_{1/2}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransR (Lin et al. 2015)	$-\Vert\mathbf{M}_r\mathbf{h} + \mathbf{r} - \mathbf{M}_r\mathbf{t}\Vert_2$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$, $r \in \mathbb{R}^k$, $\mathbf{M}_r \in \mathbb{R}^{k\times d}$
ManifoldE (Xiao, Huang, and Zhu 2016)	$-(\Vert\mathbf{h} + \mathbf{r} - \mathbf{t}\Vert_2^2 - \theta_r^2)^2$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
RotatE (Sun et al. 2019)	$-\Vert\mathbf{h} \circ \mathbf{r} - \mathbf{t}\Vert_2$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$, $\vert r_i\vert = 1$
RESCAL (Nickel, Tresp, and Kriegel 2011)	$\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$, $\mathbf{M}_r \in \mathbb{R}^{k\times k}$
DistMult (Yang et al. 2015)	$\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\mathbf{t}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ComplEx (Trouillon et al. 2016)	$\mathrm{Re}(\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\bar{\mathbf{t}})$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$
ConvE (Dettmers et al. 2018)	$f(\mathrm{vec}(f([\bar{\mathbf{r}}, \bar{\mathbf{h}}] * \omega))\mathbf{W})\mathbf{t}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
HAKE	$-\Vert\mathbf{h}_m \circ \mathbf{r}_m - \mathbf{t}_m\Vert_2 - \lambda\Vert\sin((\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p)/2)\Vert_1$	$\mathbf{h}_m, \mathbf{t}_m \in \mathbb{R}^k$,$\mathbf{r}_m \in \mathbb{R}_+^k$, $\mathbf{h}_p, \mathbf{r}_p, \mathbf{t}_p \in [0, 2\pi)^k$, , $\lambda \in \mathbb{R}$

Instead, HAKE explicitly models the modulus information, which significantly outperforms RotatE in distinguishing entities at different levels of the hierarchy.

The Ways to Model Hierarchy Structures

Another related problem is how to model hierarchy structures in knowledge graphs. Some recent work considers the problem in different ways. Li et al. (2016) embed entities and categories jointly into a semantic space and designs models for the concept categorization and dataless hierarchical classification tasks. Zhang et al. (2018) use clustering algorithms to model the hierarchical relation structures. Xie, Liu, and Sun (2016) proposed TKRL, which embeds the type information into knowledge graph embeddings. That is, TKRL requires additional hierarchical type information for entities.

Different from the previous work, our work

(a) considers the link prediction task, which is a more common task for knowledge graph embeddings;

(b) can automatically learn the semantic hierarchy in knowledge graphs without using clustering algorithms;

(c) does not require any additional information other than the triples in knowledge graphs.

3 The Proposed HAKE

In this section, we introduce our proposed model HAKE. We first introduce two categories of entities that reflect the semantic hierarchies in knowledge graphs. Afterwards, we introduce our proposed HAKE that can model entities in both of the categories.

Two Categories of Entities

To model the semantic hierarchies of knowledge graphs, a knowledge graph embedding model must be capable of distinguishing entities in the following two categories.

(a) Entities at different levels of the hierarchy. For example, “mammal” and “dog”, “run” and ”move”.

(b) Entities at the same level of the hierarchy. For example, “rose” and “peony”, “truck” and ”lorry”.

Hierarchy-Aware Knowledge Graph Embedding

To model both of the above categories, we propose a hierarchy-aware knowledge graph embedding model—HAKE. HAKE consists of two parts—the modulus part and the phase part—which aim to model entities in the two different categories, respectively. Figure 1 gives an illustration of the proposed model.

To distinguish embeddings in the different parts, we use $\mathbf{e}_m$ ($\mathbf{e}$ can be $\mathbf{h}$ or $\mathbf{t}$) and $\mathbf{r}_m$ to denote the entity embedding and relation embedding in the modulus part, and use $\mathbf{e}_p$ ($\mathbf{e}$ can be $\mathbf{h}$ or $\mathbf{t}$) and $\mathbf{r}_p$ to denote the entity embedding and relation embedding in the phase part.

The modulus part aims to model the entities at different levels of the hierarchy. Inspired by the fact that entities that have hierarchical property can be viewed as a tree, we can use the depth of a node (entity) to model different levels of the hierarchy. Therefore, we use modulus information to model entities in the category (a), as moduli can reflect the depth in a tree. Specifically, we regard each entry of $\mathbf{h}_m$ and $\mathbf{t}_m$, that is, $[\mathbf{h}_m]_i$ and $[\mathbf{t}_m]_i$, as a modulus, and regard each entry of $\mathbf{r}_m$, that is, $[\mathbf{r}]_i$, as a scaling transformation between two moduli. We can formulate the modulus part as follows:

$$\mathbf{h}_m \circ \mathbf{r}_m = \mathbf{t}_m, \text{ where } \mathbf{h}_m, \mathbf{t}_m \in \mathbb{R}^k, \text{ and } \mathbf{r}_m \in \mathbb{R}_+^k.$$

The corresponding distance function is:

$$d_{r,m}(\mathbf{h}_m, \mathbf{t}_m) = \Vert\mathbf{h}_m \circ \mathbf{r}_m - \mathbf{t}_m\Vert_2.$$

Note that we allow the entries of entity embeddings to be negative but restrict the entries of relation embeddings to be positive. This is because that the signs of entity embeddings can help us to predict whether there exists a relation between two entities. For example, if there exists a relation r between h and t_1, and no relation between h and t_2, then (h, r, t_1) is a positive sample and (h, r, t_2) is a negative sample. Our goal is to minimize $d_r(\mathbf{h}_m, \mathbf{t}_{1,m})$ and maximize $d_r(\mathbf{h}_m, \mathbf{t}_{2,m})$, so as to make a clear distinction between positive and negative samples. For the positive sample, $[\mathbf{h}]_i$ and $[\mathbf{t}_1]_i$ tend to

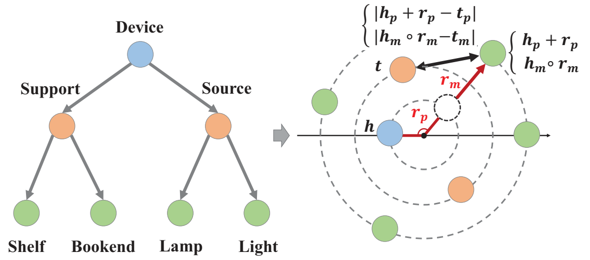


Figure 1: Simple illustration of HAKE. In a polar coordinate system, the radial coordinate aims to model entities at different levels of the hierarchy, and the angular coordinate aims to distinguish entities at the same level of the hierarchy.

share the same sign, as $[\mathbf{r}_m]_i > 0$. For the negative sample, the signs of $[\mathbf{h}_m]_i$ and $[\mathbf{t}_{2,m}]_i$ can be different if we initialize their signs randomly. In this way, $d_r(\mathbf{h}_m, \mathbf{t}_{2,m})$ is more likely to be larger than $d_r(\mathbf{h}_m, \mathbf{t}_{1,m})$, which is exactly what we desire.

Further, we can expect the entities at higher levels of the hierarchy to have smaller modulus, as these entities are more close to the root of the tree.

If we use only the modulus part to embed knowledge graphs, then the entities in the category (b) will have the same modulus. Moreover, suppose that r is a relation that reflects the same semantic hierarchy, then $[\mathbf{r}]_i$ will tend to be one, as $h \circ r \circ r = h$ holds for all h. Hence, embeddings of the entities in the category (b) tend to be the same, which makes it hard to distinguish these entities. Therefore, a new module is required to model the entities in the category (b).

The phase part aims to model the entities at the same level of the semantic hierarchy. Inspired by the fact that points on the same circle (that is, have the same modulus) can have different phases, we use phase information to distinguish entities in the category (b). Specifically, we regard each entry of $\mathbf{h}_p$ and $\mathbf{t}_p$, that is, $[\mathbf{h}_p]_i$ and $[\mathbf{t}_p]_i$ as a phase, and regard each entry of $\mathbf{r}_p$, that is, $[\mathbf{r}_p]_i$, as a phase transformation. We can formulate the phase part as follows:

$$(\mathbf{h}_p + \mathbf{r}_p) \bmod 2\pi = \mathbf{t}_p, \text{where } \mathbf{h}_p, \mathbf{r}_p, \mathbf{t}_p \in [0, 2\pi)^k.$$

The corresponding distance function is:

$$d_{r,p}(\mathbf{h}_p, \mathbf{t}_p) = \|\sin((\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p)/2)\|_1,$$

where $\sin(\cdot)$ is an operation that applies the sine function to each element of the input. Note that we use a sine function to measure the distance between phases instead of using $\|\mathbf{h}_p + \mathbf{r}_p - \mathbf{t}_p\|_1$, as phases have periodic characteristic. This distance function shares the same formulation with that of pRotatE (Sun et al. 2019).

Combining the modulus part and the phase part, HAKE maps entities into the **polar coordinate system**, where the radial coordinate and the angular coordinates correspond to the modulus part and the phase part, respectively. That is, HAKE maps an entity h to $[\mathbf{h}_m; \mathbf{h}_p]$, where $\mathbf{h}_m$ and $\mathbf{h}_p$ are generated by the modulus part and the phase part, respectively, and $[\,\cdot\,;\,\cdot\,]$ denotes the concatenation of two vectors. Obviously, $([\mathbf{h}_m]_i, [\mathbf{h}_p]_i)$ is a 2D point in the polar coordinate system. Specifically, we formulate HAKE as follows:

$$\begin{cases} \mathbf{h}_m \circ \mathbf{r}_m = \mathbf{t}_m, \text{ where } \mathbf{h}_m, \mathbf{t}_m \in \mathbb{R}^k, \mathbf{r}_m \in \mathbb{R}^k_+, \\ (\mathbf{h}_p + \mathbf{r}_p) \bmod 2\pi = \mathbf{t}_p, \text{ where } \mathbf{h}_p, \mathbf{t}_p, \mathbf{r}_p \in [0, 2\pi)^k. \end{cases}$$

The distance function of HAKE is:

$$d_r(\mathbf{h}, \mathbf{t}) = d_{r,m}(\mathbf{h}_m, \mathbf{t}_m) + \lambda d_{r,p}(\mathbf{h}_p, \mathbf{t}_p),$$

where $\lambda \in \mathbb{R}$ is a parameter that learned by the model. The corresponding score function is

$$f_r(\mathbf{h}, \mathbf{t}) = d_r(\mathbf{h}, \mathbf{t}) = -d_{r,m}(\mathbf{h}, \mathbf{t}) - \lambda d_{r,p}(\mathbf{h}, \mathbf{t}).$$

When two entities have the same moduli, then the modulus part $d_{r,m}(\mathbf{h}_m, \mathbf{t}_m) = 0$. However, the phase part $d_{r,p}(\mathbf{h}_p, \mathbf{t}_p)$ can be very different. By combining the modulus part and the phase part, HAKE can model the entities in both the category (a) and the category (b). Therefore, HAKE can model semantic hierarchies of knowledge graphs.

When evaluating the models, we find that adding a **mixture bias** to $d_{r,m}(\mathbf{h}, \mathbf{t})$ can help to improve the performance of HAKE. The modified $d_{r,m}(\mathbf{h}, \mathbf{t})$ is given by:

$$d'_{r,m}(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_m \circ \mathbf{r}_m + (\mathbf{h}_m + \mathbf{t}_m) \circ \mathbf{r}'_m - \mathbf{t}_m\|_2,$$

where $0 < \mathbf{r}'_m < 1$ is a vector that have the same dimension with $\mathbf{r}_m$. Indeed, the above distance function is equivalent to

$$d'_{r,m}(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_m \circ ((1 - \mathbf{r}'_m)/(\mathbf{r}_m + \mathbf{r}'_m)) - \mathbf{t}_m\|_2,$$

where $/$ denotes the element-wise division operation. If we let $\mathbf{r}_m \leftarrow (1 - \mathbf{r}'_m)/(\mathbf{r}_m + \mathbf{r}'_m)$, then the modified distance function is exactly the same as the original one when compare the distances of different entity pairs. For notation convenience, we still use $d_{r,m}(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}_m \circ \mathbf{r}_m - \mathbf{t}_m\|_2$ to represent the modulus part. We will conduct ablation studies on the bias in the experiment section.

Loss Function

To train the model, we use the negative sampling loss functions with self-adversarial training (Sun et al. 2019):

$$\begin{aligned} L = &- \log \sigma(\gamma - d_r(\mathbf{h}, \mathbf{t})) \\ &- \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(d_r(\mathbf{h}'_i, \mathbf{t}'_i) - \gamma), \end{aligned}$$

where γ is a fixed margin, σ is the sigmoid function, and (h'_i, r, t'_i) is the ith negative triple. Moreover,

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(\mathbf{h}'_j, \mathbf{t}'_j)}{\sum_i \exp \alpha f_r(\mathbf{h}'_i, \mathbf{t}'_i)}$$

is the probability distribution of sampling negative triples, where α is the temperature of sampling.

4 Experiments and Analysis

This section is organized as follows. First, we introduce the experimental settings in detail. Then, we show the effectiveness of our proposed model on three benchmark datasets. Finally, we analyze the embeddings generated by HAKE, and show the results of ablation studies. The code of HAKE is available on GitHub at https://github.com/MIRALab-USTC/KGE-HAKE.

Experimental Settings

We evaluate our proposed models on three commonly used knowledge graph datasets—WN18RR (Toutanova and Chen 2015), FB15k-237 (Dettmers et al. 2018), and YAGO3-10 (Mahdisoltani, Biega, and Suchanek 2013). Details of these datasets are summarized in Table 2.

WN18RR, FB15k-237, and YAGO3-10 are subsets of WN18 (Bordes et al. 2013), FB15k (Bordes et al. 2013), and YAGO3 (Mahdisoltani, Biega, and Suchanek 2013), respectively. As pointed out by Toutanova and Chen (2015) and Dettmers et al. (2018), WN18 and FB15k suffer from the test set leakage problem. One can attain the state-of-the-art results even using a simple rule based model. Therefore, we use WN18RR and FB15k-237 as the benchmark datasets.

Evaluation Protocol Following Bordes et al. (2013), for each triple (h, r, t) in the test dataset, we replace either the head entity h or the tail entity t with each candidate entity to create a set of candidate triples. We then rank the candidate triples in descending order by their scores. It is worth noting that we use the "Filtered" setting as in Bordes et al. (2013), which does not take any existing valid triples into accounts at ranking. We choose Mean Reciprocal Rank (MRR) and Hits at N (H@N) as the evaluation metrics. Higher MRR or H@N indicate better performance.

Training Protocol We use Adam (Kingma and Ba 2015) as the optimizer, and use grid search to find the best hyperparameters based on the performance on the validation datasets. To make the model easier to train, we add an additional coefficient to the distance function, i.e., $d_r(\mathbf{h}, \mathbf{t}) = \lambda_1 d_{r,m}(\mathbf{h}_m, \mathbf{t}_m) + \lambda_2 d_{r,p}(\mathbf{h}_p, \mathbf{t}_p)$, where $\lambda_1, \lambda_2 \in \mathbb{R}$.

Baseline Model One may argue that the phase part is unnecessary, as we can distinguish entities in the category (b) by allowing $[\mathbf{r}]_i$ to be negative. We propose a model—ModE—that uses only the modulus part but allow $[\mathbf{r}]_i < 0$. Specifically, the distance function of ModE is

$$d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_2, \text{where } \mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k.$$

Main Results

In this part, we show the performance of our proposed models—HAKE and ModE—against existing state-of-the-art methods, including TransE (Bordes et al. 2013), DistMult (Yang et al. 2015), ComplEx (Trouillon et al. 2016), ConvE (Dettmers et al. 2018), and RotatE (Sun et al. 2019).

Table 3 shows the performance of HAKE, ModE, and several previous models. Our baseline model ModE shares similar simplicity with TransE, but significantly outperforms it on all datasets. Surprisingly, ModE even outperforms more complex models such as DistMult, ConvE and Complex on all datasets, and beats the state-of-the-art model—RotatE—on FB15k-237 and YAGO3-10 datasets, which demonstrates the great power of modulus information. Table 3 also shows that our HAKE significantly outperforms existing state-of-the-art methods on all datasets.

WN18RR dataset consists of two kinds of relations: the symmetric relations such as $\_similar\_to$, which link entities in the category (b); other relations such as $\_hypernym$ and $\_member\_meronym$, which link entities in the category (a). Actually, RotatE can model entities in the category (b) very well (Sun et al. 2019). However, HAKE gains a 0.021 higher MRR, a 2.4% higher H@1, and a 2.4% higher H@3 against RotatE, respectively. The superior performance of HAKE compared with RotatE implies that our proposed model can better model different levels in the hierarchy.

Table 2: Statistics of datasets. The symbols #E and #R denote the number of entities and relations, respectively. #TR, #VA, and #TE denote the size of train set, validation set, and test set, respectively.

Dataset	#E	#R	#TR	#VA	#TE
WN18RR	40,493	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

FB15k-237 dataset has more complex relation types and fewer entities, compared with WN18RR and YAGO3-10. Although there are relations that reflect hierarchy in FB15k-237, there are also lots of relations, such as "/location/location/time_zones" and "/film/film/prequel", that do not lead to hierarchy. The characteristic of this dataset accounts for why our proposed models doesn't outperform the previous state-of-the-art as much as that of WN18RR and YAGO3-10 datasets. However, the results also show that our models can gain better performance so long as there exists semantic hierarchies in knowledge graphs. As almost all knowledge graphs have such hierarchy structures, our model is widely applicable.

YAGO3-10 datasets contains entities with high relation-specific indegree (Dettmers et al. 2018). For example, the link prediction task $(?, hasGender, male)$ has over 1000 true answers, which makes the task challenging. Fortunately, we can regard "male" as an entity at higher level of the hierarchy and the predicted head entities as entities at lower level. In this way, YAGO3-10 is a dataset that clearly has semantic hierarchy property, and we can expect that our proposed models is capable of working well on this dataset. Table 3 validates our expectation. Both ModE and HAKE significantly outperform the previous state-of-the-art. Notably, HAKE gains a 0.050 higher MRR, 6.0% higher H@1 and 4.6% higher H@3 than RotatE, respectively.

Analysis on Relation Embeddings

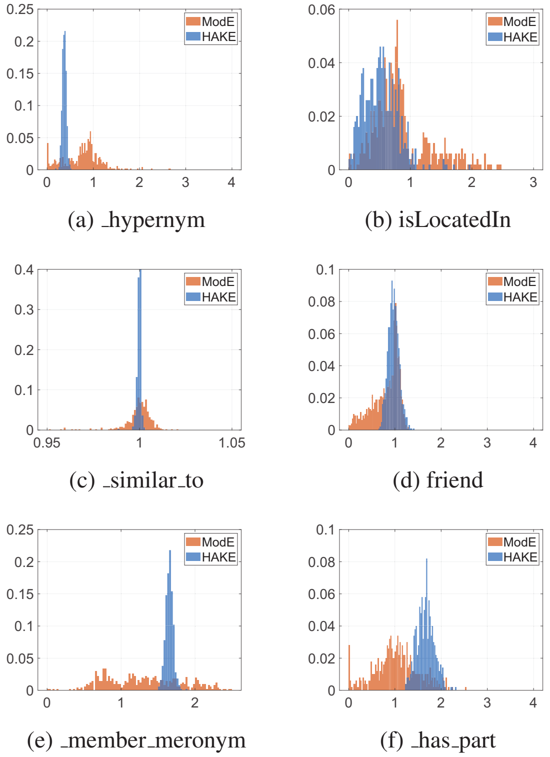
In this part, we first show that HAKE can effectively model the hierarchy structures by analyzing the moduli of relation embeddings. Then, we show that the phase part of HAKE can help us to distinguish entities at the same level of the hierarchy by analyzing the phases of relation embeddings.

In Figure 2, we plot the distribution histograms of moduli of six relations. These relations are drawn from WN18RR, FB15k-237, and YAGO3-10. Specifically, the relations in Figures 2a, 2c, 2e and 2f are drawn from WN18RR. The relation in Figure 2d is drawn from FB15k-237. The relation in Figure 2b is drawn from YAGO3-10. We divide the relations in Figure 2 into three groups.

(A) Relations in Figures 2c and 2d connect the entities at the same level of the semantic hierarchy;

Table 3: Evaluation results on WN18RR, FB15k-237 and YAGO3-10 datasets. Results of TransE and RotatE are taken from Nguyen et al. (2018) and Sun et al. (2019), respectively. Other results are taken from Dettmers et al. (2018).

	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
TransE	.226	-	-	.501	.294	-	-	.465	-	-	-	-
DistMult	.43	.39	.44	.49	.241	.155	.263	.419	.34	.24	.38	.54
ConvE	.43	.40	.44	.52	.325	.237	.356	.501	.44	.35	.49	.62
ComplEx	.44	.41	.46	.51	.247	.158	.275	.428	.36	.26	.40	.55
RotatE	.476	.428	.492	.571	.338	.241	.375	.533	.495	.402	.550	.670
ModE	.472	.427	.486	.564	.341	.244	.380	.534	.510	.421	.562	.660
HAKE	**.497**	**.452**	**.516**	**.582**	**.346**	**.250**	**.381**	**.542**	**.545**	**.462**	**.596**	**.694**



(a) _hypernym

(b) isLocatedIn

(c) _similar_to

(d) friend

(e) _member_meronym

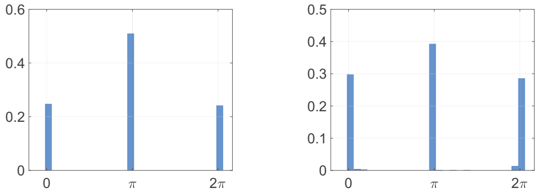
(f) _has_part

Figure 2: Distribution histograms of moduli of some relations. The relations are drawn from WN18RR, FB15k-237 and YAGO3-10 dataset. The relation in (d) is */celebrities/celebrity/celebrity_friends/celebrities/friendship/friend*. Let *friend* denote the relation for simplicity.

(B) Relations in Figures 2a and 2b represent that tail entities are at higher levels than head entities of the hierarchy;

(C) Relations in Figures 2e and 2f represent that tail entities are at lower levels than head entities of the hierarchy.

As described in the model description section, we expect entities at higher levels of the hierarchy to have small moduli. The experiments validate our expectation. For both ModE and HAKE, most entries of the relations in the group (A) take values around one, which leads to that the head entities and tail entities have approximately the same moduli. In the group (B), most entries of the relations take values less than one, which results in that the head entities have smaller moduli than the tail entities. The cases in the group (C) are contrary to that in the group (B). These results show that our model can capture the semantic hierarchies in knowledge graphs. Moreover, compared with ModE, the relation embeddings' moduli of HAKE have lower variances, which shows that HAKE can model hierarchies more clearly.



(a) _similar_to

(b) friend

Figure 3: Distribution histograms of phases of two relations that reflect the same hierarchy. The relations in Figure (a) and (b) are drawn from WN18RR and FB15k-237, respectively.

As mentioned above, relations in the group (A) reflect the same semantic hierarchy, and are expected to have the moduli of about one. Obviously, it is hard to distinguish entities linked by these relations only using the modulus part. In Figure 3, we plot the phases of the relations in the group (A). The results show that the entities at the same level of the hierarchy can be distinguished by their phases, as many phases have the values of π.

Analysis on Entity Embeddings

In this part, to further show that HAKE can capture the semantic hierarchies between entities, we visualize the embeddings of several entity pairs.

We plot the entity embeddings of two models: the previous state-of-the-art RotatE and our proposed HAKE. RotatE regards each entity as a group of complex numbers. As a complex number can be seen as a point on a 2D plane, we

Table 4: Ablation results on WN18RR, FB15k-237 and YAGO3-10 datasets. The symbols **m**, **p**, and **b** represent the modulus part, the phase part, and the mixture bias term, respectively.

			WN18RR				**FB15k-237**				**YAGO3-10**			
m	**p**	**b**	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
✓			.240	.047	.404	.527	.258	.121	.333	.508	.476	.374	.541	.658
	✓		.465	.423	.480	.550	.324	.226	.361	.519	.480	.383	.532	.664
✓	✓		.496	.449	**.517**	**.584**	.336	.239	.373	.533	.522	.429	.581	.693
✓	✓	✓	**.497**	**.452**	.516	.582	**.346**	**.250**	**.381**	**.542**	**.545**	**.462**	**.596**	**.694**

Table 5: Comparison results with TKRL models (Xie, Liu, and Sun 2016) on FB15k dataset. RHE, WHE, RHE+STC, and WHE+STC are four versions of TKRL model , of which the results are taken from the original paper.

	HAKE	RHE	WHE	RHE+STC	WHE+STC
H@10	**.884**	.694	.696	.731	.734

can plot the entity embeddings on a 2D plane. As for HAKE, we have mentioned that it maps entities into the polar coordinate system. Therefore, we can also plot the entity embeddings generated by HAKE on a 2D plane based on their polar coordinates. For a fair comparison, we set $k = 500$. That is, each plot contains 500 points, and the actual dimension of entity embeddings is 1000. Note that we use the logarithmic scale to better display the differences between entity embeddings. As all the moduli have values less than one, after applying the logarithm operation, the larger radii in the figures will actually represent smaller modulus.

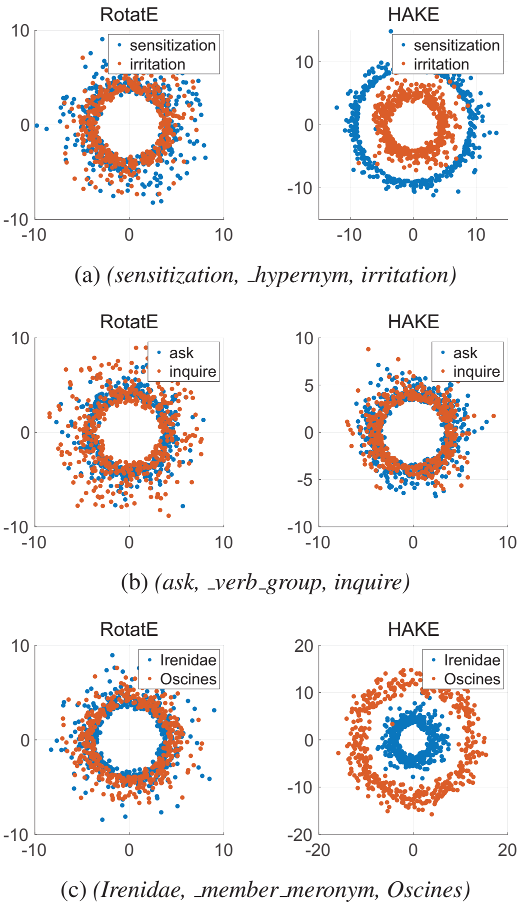
Figure 4 shows the visualization results of three triples from the WN18RR dataset. Compared with the tail entities, the head entities in Figures 4a, 4b, and 4c are at lower levels, similar levels, higher levels in the semantic hierarchy, respectively. We can see that there exist clear concentric circles in the visualization results of HAKE, which demonstrates that HAKE can effectively model the semantic hierarchies. However, in RotatE, the entity embeddings in all three subfigures are mixed, making it hard to distinguish entities at different levels in the hierarchy.

Ablation Studies

In this part, we conduct ablation studies on the modulus part and the phase part of HAKE, as well as the mixture bias item. Table 4 shows the results on three benchmark datasets.

We can see that the bias can improve the performance of HAKE on nearly all metrics. Specifically, the bias improves the H@1 score of 4.7% on YAGO3-10 dataset, which illustrates the effectiveness of the bias.

We also observe that the modulus part of HAKE does not perform well on all datasets, due to its inability to distinguish the entities at the same level of the hierarchy. When only using the phase part, HAKE degenerates to the pRotatE model (Sun et al. 2019). It performs better than the modulus part, because it can well model entities at the same level of the hierarchy. However, our HAKE model significantly outperforms the modulus part and the phase part on all datasets, which demonstrates the importance to combine the two parts for modeling semantic hierarchies in knowledge graphs.



(a) *(sensitization, _hypernym, irritation)*

(b) *(ask, _verb_group, inquire)*

(c) *(Irenidae, _member_meronym, Oscines)*

Figure 4: Visualization of the embeddings of several entity pairs from WN18RR dataset.

Comparison with Other Related Work

We compare our models with TKRL models (Xie, Liu, and Sun 2016), which also aim to model the hierarchy structures. For the difference between HAKE and TKRL, please refer to the Related Work section. Table 5 shows the H@10 scores of

HAKE and TKRLs on FB15k dataset. The best performance of TKRL is .734 obtained by the WHE+STC version, while the H@10 score of our HAKE model is .884. The results show that HAKE significantly outperforms TKRL, though it does not require additional information.

5 Conclusion

To model the semantic hierarchies in knowledge graphs, we propose a novel hierarchy-aware knowledge graph embedding model—HAKE—which maps entities into the polar coordinate system. Experiments show that our proposed HAKE significantly outperforms several existing state-of-the-art methods on benchmark datasets for the link prediction task. A further investigation shows that HAKE is capable of modeling entities at both different levels and the same levels in the semantic hierarchies.

Acknowledgments

We would like to thank all the anonymous reviewers for their insightful comments. This work was supported in part by NSFC (61525206, 61822604, 61836006).

References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.

Dettmers, T.; Pasquale, M.; Pontus, S.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.

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

Guo, L.; Sun, Z.; and Hu, W. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *ICML*.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *WSDM*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Li, Y.; Zheng, R.; Tian, T.; Hu, Z.; Iyer, R.; and Sycara, K. 2016. Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In *COLING*.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.

Liu, H.; Wu, Y.; and Yang, Y. 2017. Analogical inference for multi-relational embeddings. In *ICML*.

Mahdisoltani, F.; Biega, J.; and Suchanek, F. M. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM*.

Nguyen, D. Q.; Nguyen, T. D.; Nguyen, D. Q.; and Phung, D. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *NAACL*.

Nguyen, D. Q.; Vu, T.; Nguyen, T. D.; Nguyen, D. Q.; and Phung, D. 2019. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. In *NAACL*.

Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic embeddings of knowledge graphs. In *AAAI*.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *ICML*.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *ESWC*.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NeurIPS*.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR*.

Toutanova, K., and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *The Workshop on CVSC*.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *ICML*.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; and Guo, M. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*.

Xiao, H.; Huang, M.; and Zhu, X. 2016. From one point to a manifold: Knowledge graph embedding for precise link prediction. In *IJCAI*.

Xie, R.; Liu, Z.; and Sun, M. 2016. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*.

Yang, B.; Yih, S. W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

Zhang, Z.; Zhuang, F.; Qu, M.; Lin, F.; and He, Q. 2018. Knowledge graph embedding with hierarchical relation structure. In *EMNLP*.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced language representation with informative entities. In *ACL*.