A Knowledge Representation Learning Based on the Attention Mechanism Enhanced by Prior Probability

Anonymous Author(s)

ABSTRACT

Representation learning of knowledge graph (KG) aims to encode components of a KG into a continuous low-dimensional vector space, so as to simplify the manipulation while preserving the inherent structure of the KG. Similar to the large-scale data, the knowledge graph also obeys the long-tail distribution, and faces serious data sparsity problems in the entities and relationships in the long-tail part. One of the most effective methods to further alleviate data sparsity is considering entity types. However existing methods integrating entity types ignore an essential information that the type weight of an entity expressing should be different while the entity has multiple types, which is significant for knowledge graph representation learning. In this paper, a novel attention mechanism named Prior-Probability Enhanced Attention Mechanism is proposed to capture the weight between entities and types. Our model is evaluated on link prediction and triplets classification. The experimental results show that our model significantly outperforms all baselines, which indicates that the Prior-Probability of KG for the entity type can be used to enhance the semantic expression ability of the model.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Knowledge graph, Knowledge graph representation learning, Attention mechanism, Embedding

ACM Reference Format:

Anonymous Author(s). 2018. A Knowledge Representation Learning Based on the Attention Mechanism Enhanced by Prior Probability. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recent years have witnessed rapid growth in knowledge graph (KG) construction and application. A large number of KGs, such as Freebase [3], DBpedia [23], YAGO [33], and NELL [8], have been created and successfully applied to many real-world applications[37], from semantic parsing [2],[18] and named entity disambiguation [10], [46], to information extraction [19], [26], [12], [9], [43] and question answering [7], [4], [44]. A KG is a multi-relational graph composed of entities (nodes) and relations (different types of edges). Each edge is represented as a triple of the form (head entity, relation, tail entity), also called a fact, indicating that two entities are connected by a specific relation, e.g., $(AlfredHitchcock, DirectorOf, Psycho)$. Although effective in representing structured data, the underlying symbolic nature of such triples usually makes KGs hard to manipulate.

In order to obtain the underlying semantic information of the entities and relationships in the knowledge graph, and to facilitate downstream tasks to use the knowledge graph, the distributed representations learning technology of the knowledge graph has been widely studied and applied [28], [6], [38], [24], [20], [5], [32]. Knowledge graph representation learning, also known as Knowledge graph embedding is to map entities and relationships into low-dimensional continuous vector spaces, so as to simplify the manipulation while preserving the inherent structure of the KG. Those entity and relation embeddings can further be used to benefit all kinds of tasks, such as knowledge reasoning [6], [38], relation extraction [39], [11], [30], entity classification [28], [29], and entity resolution [28], [5], [13].

Although the knowledge graph embedding can effectively perform knowledge reasoning and is convenient for other tasks, the problem of the knowledge graph itself has data sparseness, which leads to the lack of embedding performance of the learned knowledge graph and cannot encode sparse entities well. Therefore, researchers have proposed some models improving the performance of representation learning by introducing some external information or prior knowledge. The external information includes text description information of entities [40], [41], type information [22], [40], [14] and relations logical rules [31], [15], [45] and so on. Among them, entity type, as a part of the knowledge graph ontology concept, contains rich semantic information, which can well guide the learning of entity and relationship vector representation.

The type information of entity can be represented by `rdf:type` in RDFS (Resource Description Framework Schema) , which is a kind of common language for describing entities and resources.For example, $AlfredHitchcock$ has the type of $Person$, and $Psycho$ the type of $CreativeWork$. This kind of information is available in most KGs, usually encoded by a specific relationship and stored also in the form of triples, e.g., $(Psycho, \texttt{rdf:type}, CreativeWork)$. A straightforward method to model such information, as investigated in [29], is to take `rdf:type` as an ordinary relationship and the corresponding triples as ordinary training examples. In addition to reflecting which type an entity belongs to, it also has two other important characteristics.

First, the type constraint of the relationship. Each relationship plays its own role. It specifies which type the head entity and tail entity associated with the relationship should belong to, which can

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

be represented by `rdfs:domain` and `rdfs:range` in RDFS. As shown in Figure 1, the *works_written* relationship is constrained by (*works_written*, `rdfs:domain`, *Writer*) and (*works_written*, *rdfs : range*, *Written_Work*). In the triples (*William_Shakespeare*, *works_written*, *Romeo_and_Juiet*), obviously, the entity *William_Shakespeare* expresses its characteristics as a *Writer*, and the entity *Romeo_and_Juiet* expresses its characteristics as a *Written_Work*. In the process of representation learning, this information can ensure that the relationship is semantically related to entities that meet its type constraints, but not related to entities that do not meet constraints, and learn their latent semantics more accurately.

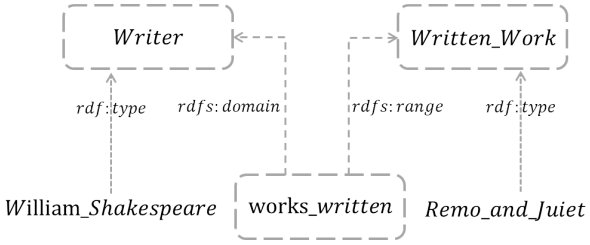

Figure 1

Second, the weight information of the type of the entity can be obtained from the dataset. While there are multiple types of entities, the type's weight of each entity should be different. According to the number of occurrences of an entity in the knowledge base, the frequency of the entity's performance of a certain type can be obtained. For example, *William_Shakespeare* frequently appears as an *Writer*, so the entity *William_Shakespeare* mainly represents the *Writer* characteristic on the whole. The frequency of the entity *William_Shakespeare* representing as an *Poet* is less than the frequency expressing as an *Writer*, so it secondly expresses the *Poet* feature. This information can help the model accurately describe the semantic information of entities, and plays an important role in downstream tasks such as predicting unknown relationships between entities.

How to integrate the entity's type weight information and relation type constraints into the existing knowledge graph representation learning is the main challenge for introducing entity type information. In the existing work, Denis Krompaß et al. [22] combined the relationship type constraints into the three existing knowledge graph embedding models, only entities that meet its category constraints are trained for each relationship. TKRL [40] also considers the hierarchical structure information of the type and the type constraints of the relation. It regards the type as a projection matrix. Use linear combinations of matrices or matrix products to model the hierarchical structure, and then apply the encoded type projection matrix as a constraint to the entity vector. The JOIE [17] maps the types to different low-dimensional vector spaces, maps the entity vector to the space through linear transformation to ensure that it is close to its type and far away from other types which it does not belong to. However, these models only focus on the relation type constraints but ignore the entity's type weight information.

Our contributions in this paper are:

1) By observing the frequency which an entity appearing a certain type in the knowledge base, we proposed an attention mechanism with enhanced prior probability to enhance the types of entities.
2) We propose a new knowledge representation learning method, which integrates entity type and relationship type constraints.
3) In experiments, our approach outperforms baselines and so on in link prediction and triplets classification tasks.

2 RELATED WORK

In this section, we will outline the existing basic knowledge graph representation learning model and the representation learning models combined with type information.

2.1 Basic models

2.1.1 Translation-based Models. These models regard the relationship as a translation operation from the head entity to the tail entity, and measures the authenticity of the triplet by calculating the Euclidean distance between the head entity and the tail entity vector after translation. Among them, TransE[6], TransH[38], TransR[24] and TransD[21] are the most representative models.

TransE[6]. Given a triple (h, r, t), TransE assumes that the tail entity $\mathbf{t} \in \mathbb{R}^d$ is equal to the sum of the head entity $\mathbf{h} \in \mathbb{R}^d$ and the relation $\mathbf{r} \in \mathbb{R}^d$, i.e. $\mathbf{t} \approx \mathbf{h} + \mathbf{r}$. Its scoring function and loss function are shown as below:

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

TransE is suitable for 1-to-1 relations, but it has problems when handling 1-to-N, N-to-1, and N-to-N relations.

TransH[38]. TransH attempts to alleviate the problems of TransE above. It regards a relation vector r as a translation on a hyperplane with w_r as the normal vector. The vector embeddings will be first projected to the hyperplane of relation r and get $\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r$ and $\mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r$. The loss function of TransH is

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

TransR[24]. TransR addresses the issue in TransE and TransH that some entities are similar in the entity space but comparably different in other specific aspects. It sets a transfer matrix $\mathbf{M}_r$ for each relation r to map entity embedding to relation vector space. Its score function is defined as following:

$$f(h, r, t) = \|\mathbf{M}_r \mathbf{h} + \mathbf{r} - \mathbf{M}_r \mathbf{t}\|_{1/2} \tag{3}$$

TransD[21]. TransD simplifies TransR by further decomposing the projection matrix into a product of two vectors. Specifically, for each fact (h, r, t), TransD introduces additional mapping vectors $\mathbf{w}_h, \mathbf{w}_t \in \mathbb{R}^d$ and $\mathbf{w}_r \in \mathbb{R}^d$, along with the entity/relation representations $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ and $\mathbf{r} \in \mathbb{R}^d$. Two projection matrices $\mathbf{M}_r^1$ and $\mathbf{M}_r^2$ are accordingly defined as

$$\mathbf{M}_r^1 = \mathbf{w}_r \mathbf{w}_h^\top + \mathbf{I}, \quad \mathbf{M}_r^2 = \mathbf{w}_r \mathbf{w}_t^\top + \mathbf{I} \tag{4}$$

These two projection matrices are then applied on the head entity vector $\mathbf{h}$ and the tail entity vector $\mathbf{t}$ respectively to get their projections, i.e.,

$$\mathbf{h}_\perp = \mathbf{M}_r^1 \mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r^2 \mathbf{t} \tag{5}$$

With the projected entities, the scoring function is defined in the same way as in TransH as following:

$$f(h,r,t) = \|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_{1/2} \tag{6}$$

2.1.2 Semantic Matching Models. Semantic matching models exploit similarity-based scoring functions. They measure plausibility of facts by matching latent semantics of entities and relations embodied in their vector space representations. Among them, RESCAL[28], DistMult[42] and HolE[27] are the simplest and widely used models.

RESCAL[28]. RESCAL associates each entity with a vector to capture its latent semantics. Each relation is represented as a matrix which models pairwise interactions between latent factors. The score of a fact (h,r,t) is defined by a bilinear function

$$f(h,r,t) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t} = \sum_{i=0}^{d-1}\sum_{j=0}^{d-1} [\mathbf{M}_r]_{ij} \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_j \tag{7}$$

where $\mathbf{h}, \mathbf{r} \in \mathbb{R}^d$ are vector representations of the entities, and $\mathbf{M}^r \in \mathbb{R}^{d\times d}$ is a matrix associated with the relation. This score function captures pairwise interactions between all the components of h and t.

DistMult[42]. DistMult uses a diagonal matrix to represent the relationship vector $\mathbf{r} \in \mathbb{R}^d$, which is regarded as a linear transformation matrix to map the head entity vector $\mathbf{h} \in \mathbb{R}^d$ to the tail entity vector $\mathbf{t} \in \mathbb{R}^d$. Its scoring function and loss function are shown as follow:

$$f(h,r,t) = \mathbf{h}^\top \operatorname{diag}(\mathbf{r})\mathbf{t} = \sum_{i=0}^{d-1} [\mathbf{r}]_i \cdot [\mathbf{h}]_i \cdot [\mathbf{t}]_i \tag{8}$$

This score captures pairwise interactions between only the components of h and t along the same dimension. However, this oversimplified model can only deal with symmetric relations which is clearly not powerful enough for general KGs.

HolE[27]. HolE combines the expressive power of RESCAL with the efficiency and simplicity of DistMult. It represents both entities and relations as vectors in $\mathbb{R}^d$.Given a fact (h,r,t), the entity representations are first composed into $\mathbf{h}\star\mathbf{t} \in \mathbb{R}^d$ by using the circular correlation operation, namely

$$[\mathbf{h}\star\mathbf{t}]_i = \sum_{k=0}^{d-1} [\mathbf{h}]_k \cdot [\mathbf{t}]_{(k+i) \bmod d} \tag{9}$$

The compositional vector is then matched with the relation representation to score that fact, i.e.,

$$f(h,r,t) = \mathbf{r}^\top(\mathbf{h}\star\mathbf{t}) = \sum_{i=0}^{d-1} [\mathbf{r}]_i \sum_{k=0}^{d-1} [\mathbf{h}]_k \cdot [\mathbf{t}]_{(k+i) \bmod d} \tag{10}$$

Circular correlation makes a compression of pairwise interactions and HolE requires less parameters which is more efficient than RESCAL. Meanwhile, since circular correlation is not commutative, HolE is able to model asymmetric relations as RESCAL does.

2.2 Models combined with entity type

As a part of the knowledge graph, entity type information can be used to supplement the semantic information of entities and relationships. Therefore, many researchers have tried to incorporate type information into the knowledge graph representation learning model to improve the performance of representation learning.

Denis Krompaß et al. [22] considered the type constraint of the relationship and introduced it into the three existing knowledge graph embedding models, and only trained entities that meet its type constraint for each relationship. In addition, the authors pointed out that the type constraint information of the relationship is usually missing, so they proposed LCWA (Local Closed-World Assumptions). Using the known triples in the knowledge graph, LCWA regards the relationship's head entity set and tail entity set approximation as its type constraint.

Guo et al.[14]first proposed a method named Semantic Smooth Embedding, which requires entities belonging to the same type to be adjacent to each other in the vector space. The disadvantage of this method is that it assumes that an entity can only belong to one type, and cannot model the type constraints of the relationship.

TKRL[40] considers the hierarchical structure of the type and the type constraints of the relationship. It regards the type as a projection matrix, uses the linear combination of the matrix or the matrix product to model the hierarchical structure, and then uses the encoded type projection matrix as the constraint used on the entity.

The recently proposed JOIE model [17] integrates the knowledge graph and its ontology concept information, and uses each other's semantic information to learn a more comprehensive knowledge graph embedding. JOIE uses a new method to model type information. It maps the type to a different low-dimensional vector space, and maps the entity vector to the type space through a linear transformation. The mapped entity is required to be as close as possible to the position of its type and far away from the type that it does not belong to. Similarly, for the hierarchical structure of categories, the affiliation between types is also realized through this linear transformation.But this way of modeling hierarchical structures with linear transformations is not intuitive, and JOIE does not make use of the type constraints of relations.

3 PROBLEM FORMULATION

In this section, we will introduce in detail how to combine type information and relational type constraints into the existing knowledge graph representation learning model.

3.1 Definitions

In this section, we give the definitions related to the proposed model as bellow and the basic mathematical symbols used in the model as Table 1.

Definition 1.Knowledge Graph. Given a symbol$G = (E,R,C,T)$ represents the knowledge graph. Among them, E represents the set of entities, R represents the set of relations, C represents the set of type, and T represents the truple set, that is, $T = \{(h,r,t) \mid h,t \in E, r \in R\} \subseteq E \times R \times E$.

*Definition 2.Entity Type.*We use C to denote the set of all types. The set $I_e = \{(e,c) \mid e \in E, c \in C\}$ denotes the set of entities and their according categories.

Definition 3.Relation's Type Constraint. When the relation in the triple is constrained by `rdfs:domain` and $rdfs:range$, the type of the head entity and the type of the tail entity corresponding to

Notation	Description
e	The entity
c	The type
E	The set of entities
R	The set of relations between entities
(h, r, t)	Triplet indicating a relation r from head entity h to tail entity t
S	The set of triplets
C	The set of types
I_e	The set of entity and its corresponding types
$Domain_r$	The set of relations and its corresponding head entity type constrain
$Range$	The set of relations and its corresponding tail entity type constrain
r_c	The radius of the type sphere c
$\alpha\,(e, c)$	The weight between entity e and type c
$freq\,(e, c)$	The frequency of entity e expressing type c
W_c	A weight parameter used to extract the type c features of the entity
b_c	A bias parameter used to extract the type c features of the entity

Table 1: Notation used in this slide.

this relation are determined. Given $Domain = \{(c_h, r) \mid c_h \in C, r \in R\}$ represents the relation and its corresponding head entity type set; $Range = \{(r, c_t) \mid r \in R, c_t \in C\}$ represents the set of the relation and its corresponding tail entity type.

Definition 4.Knowledge Representation Learning Based on Attention Mechanism Enhanced By Prior-Probability. In real life, the weight of type each entities express should be different. According to the frequency an entity appears in a certain relationship which has type constrains in the knowledge base, we can obtain the weight of a certain type the entity express. Representation learning Based on Attention Mechanism Enhanced By Prior-Probability aims to project both entities $h, t \in E$ and relations $r \in R$ as vectors $\mathbf{h}, \mathbf{t}$ and $\mathbf{r}$ in a continuous low-dimensional type space $\mathbb{R}^d$, which can be written as $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$. The entity type $c \in C$ is modeled as an sphere consisted of a sphere center point $\mathbf{c} \in \mathbb{R}^d$ and a radius $\mathbf{r}_c \in \mathbb{R}$ in the continuous low-dimensional type space. The knowledge graph represents the learning model by designing a scoring function $f_1 : E \times C \rightarrow \mathbb{R}$ to model the `rdf:type` relationship considering types' weight information and a scoring function $f_2 : E \times R \times E \rightarrow \mathbb{R}$ to measure the authenticity of the input triples. The less the value of the scoring function, the more likely the triple is to be true. The vectors of entities, types and relations are learned by minimizing the jointly learning loss function value of all known triples in datasets and negative triples created by replacing head and tail entities.

4 OUR MODEL

In this section, we introduce our proposed model, which jointly embed entities, relations and concepts using two model components:Type Enhanced Component and Specific-relation Type Constraints Based Embedding Component, as show as Figure 3.

Specifically, Section 4.1 gives the definition of the prior probability which can be obtained form the datasets; Section 4.2 introduces how to integrate the prior probability into the attention mechanism; Section 4.3 shows how to use prior probability to enhance entity's type; Section 4.4 describes how to combine the type constraints of the relationship into the knowledge graph embedding model; Section 4.5 give the definition how to jointly train and learn the vector representation of entities, relationships, and types.

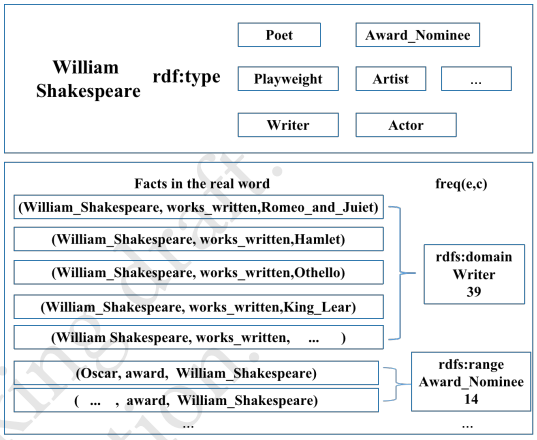

Figure 2: *WilliamShakespeare*'s Prior Probability In Real World

4.1 Prior Probability In Knowledge Graph

The Knowledge Graph describes the concepts, entities and their relationships in the objective world in a structured form, and expresses the information on the Internet in a form closer to the human cognitive world. So the same as the real world, which type *William_Shakespeare* mainly plays is depended on its type frequency appeared in the Knowledge Graph as shown in the Figure 2. If *William* appears a lot in the *works_written* relation as a head entity than other relation, then he will mainly play the *works_written*'s $rfds$: $domain$ constrain which is *Writer*. Inspired by this, prior probability of each entity can be obtained, which can help entities accurately express their semantics. $freq(e, c)$ represents the occurrences of entity e expressing as type c

4.2 Attention Mechanism Enhanced By Prior-Probability

Here we propose an attention mechanism enhanced by priorprobability as shown in Figure 4. The particular attentional setup utilized by us closely follows the work of Graph Attention Network [36]—but the framework is agnostic to the particular choice of attention mechanism. The input to attention mechanism is a set which consists of entity and its according types. For each entity $\mathbf{e} \in \mathbb{R}^d$, calculate attention coefficients

$$k\,(e, c) = a([\mathbf{We} \| \mathbf{Wc}]) \tag{11}$$

between itself and its each type $\mathbf{c} \in \mathbb{R}^d$.

In order to obtain sufficient expressive power to transform the input features into higher-level features, at least one learn-able

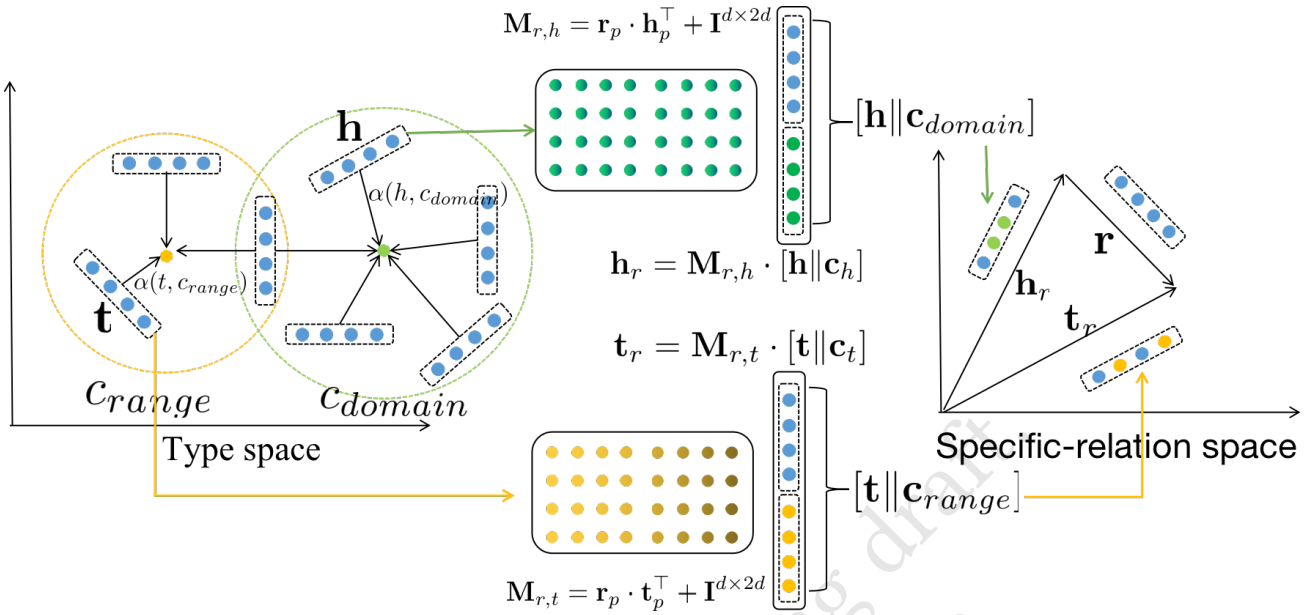

Figure 3: First, the Type Enhanced Component encodes each entity as a vector in the type space, and encodes the type as a sphere in the same space. Using the relative position to model the `rdf:type` relation between entities and types. The priori-probability enhanced attention mechanism captures the weight information of entities and different types before, which can be described as the distance from the entity vector e to the center of the sphere c. Through the type enhancement component, the semantics of the entity can be accurately expressed. Then the translation-based model integrates the type constraints of the relationship into the representation learning model, so that the entity expresses the different characteric in the specific-relation.

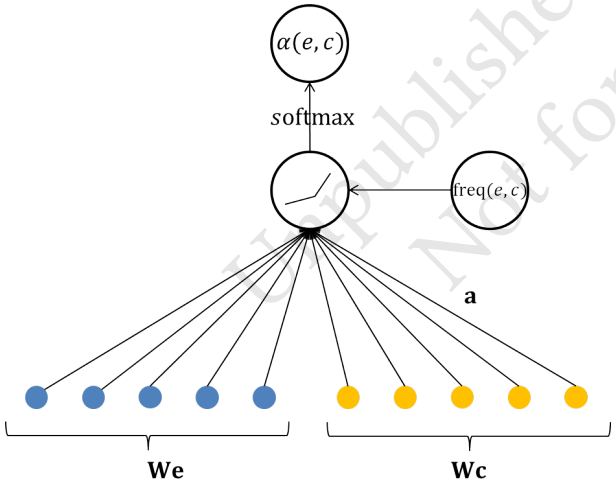

Figure 4: The Attention Mechanism Enhanced By Prior-Probability

linear transformation is required. First, as an initial step, a shared linear transformation, parametrized by a weight matrix **W**, is applied to each entity and its according type to augment features of them. Then concatenate the augmented entity and its type by using $[\cdot\|\cdot]$. Finally, $a(\cdot)$ maps the concatenated high-dimensional features to a real number which is the attention coefficient. Here, the attention mechanism $a(\cdot)$ is a single-layer feedforward neural network. **We use** *LeakyReLU* **nonlinearity (with negative input slope** $\alpha = 0.2$**).** Finally, to make coefficients easily comparable across different types, we normalize them across all types of this entity using the $softmax$ function:

$$\begin{aligned} \alpha(e,c) &= \text{softmax}[freq(e,c)\cdot k(e,c)] \\ &= \frac{\exp(freq(e,c)\cdot k(e,c))}{\sum_{(e,c)\in I_e}\exp(freq(e,c)\cdot k(e,c))} \end{aligned} \tag{12}$$

where $\alpha(e,c)$ represents the weight between entity e and its according type c.

4.3 Type Enhanced Component

Inspired by the SSE [14] and the TransC[25], entities of the same type will be close to each other in the vector space. However, these models do not notice the situation that the type weight of each entity should be different while there are multiple types for the entity. Hence, we propose a Type Enhanced Component (TEC) based on attention mechanism enhanced by prior-probability. Similiar to TransC, we encode each type in knowledge graph as a sphere and each entity as a vector in the same semantic space. The improvement is that the Euclidean distance between the entity and the center of the type sphere is calculated by attention mechanism. The greater the weight, the closer the distance to the center of this

type sphere. If the entity does not have a certain type, the entity does not appear in the spherical range of the type. We define the score functions and loss function as follows.

$$f_r^1(e,c) = \|\mathbf{e}-\mathbf{c}\|_2 - \alpha(e,c)\cdot r_c \tag{13}$$

$$f_r^2(e,c') = \|\mathbf{e}-\mathbf{c}'\|_2 - r_c \tag{14}$$

$f_r^1(e,c)$ corresponds to the situation which entity e is an instance of type c. As shown in the Figure 5, the more $\alpha(e,c)$ is, the less distance between entity $\mathbf{e}$ and the type's corresponding sphere center $\mathbf{c}$. For the situation $(e,c) \notin I_e$, the score function $f_r^2(e,c')$ defines that entity e's representation $\mathbf{e}$ should be not in the sphere belonged to type c.

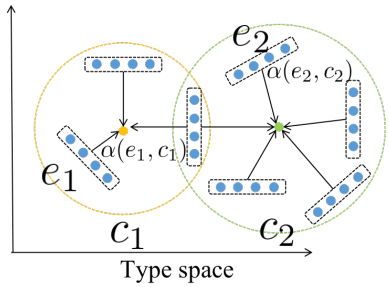

Figure 5: Type Enhanced Component

4.4 Specific-relation Type Constraints Based Embedding Component

Existing translation-based models perform well in knowledge graphs, but few of them make full use of the rich information located in relation type constrains. We believe that every entity in different relations, as the reflections of itself from various angles, should have different representations.

TransE has issues while modeling N-to-1, 1-to-N and N-to-N relations, since each entity has only one representation in every scenario. We believe that every entity in different scenarios, as the reflections of itself from various angles, should have different representations. According to the Definition 3.1, for a specific-relation, the type of the head entity and the type of the tail entity corresponding to this relation are determined. Given a triple (h,r,t) decorated with $(c_h, \texttt{rdfs:domain}, r)$ and $(r, rdfs:range, c_t)$, the head entity position should be an entity with type c_h, and the projection matrix corresponding to head entity type c_h for relation r is $\mathbf{M}_{r,h}$. The tail entity position should be an entity with type c_t, and the projection matrix corresponding to tail entity type c_t for the specific-relation r is represented as $\mathbf{M}_{r,t}$.

$$\mathbf{M}_{r,h} = \mathbf{r}_p \cdot \mathbf{h}_p^\top + \mathbf{I}^{d\times 2d} \tag{15}$$

$$\mathbf{M}_{r,t} = \mathbf{r}_p \cdot \mathbf{t}_p^\top + \mathbf{I}^{d\times 2d} \tag{16}$$

In the formula (15) and (16), $\mathbf{h}_p, \mathbf{t}_p \in \mathbb{R}^{2d}$ and $\mathbf{r}_p \in \mathbb{R}^d$ are parameters related to entity type c_h, tail type c_t and relations r respectively, which can be learned by the model. And $\mathbf{I}^{d\times 2d}$ is an identity matrix.

Then concatenate the head entity vector $\mathbf{h}$ and its $\texttt{rdfs:domain}$ constrained type vector $\mathbf{c}_h$ by using $[\cdot\|\cdot]$ so as to integrating type information into entity. The same operation is performed on the tail entity. To implement the multiple representations of entities in different scenarios, we set type-specific projection $\mathbf{M}_{r,c}$ constructed by specific-relation and its type constrains, and then represent both $[\mathbf{h}\|\mathbf{c}_h]$ and $[\mathbf{t}\|\mathbf{c}_t]$ under the projections of the specific types $\mathbf{M}_{r,h}$ and $\mathbf{M}_{r,t}$ which head and tail should belong to in this relation.

$$\mathbf{h}_r = \mathbf{M}_{r,h} \cdot [\mathbf{h}\|\mathbf{c}_h] \tag{17}$$

$$\mathbf{t}_r = \mathbf{M}_{r,t} \cdot [\mathbf{t}\|\mathbf{c}_t] \tag{18}$$

The whole process is as shown in the Figure 6. The score function

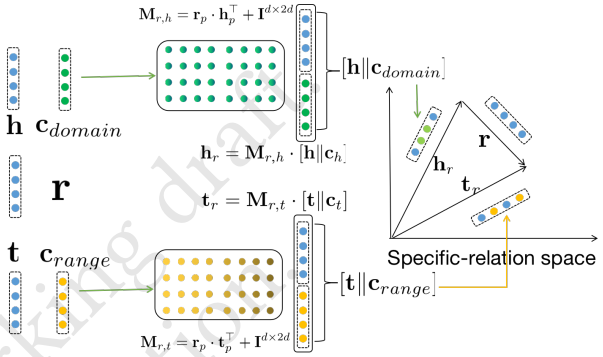

Figure 6: Embedding Component

and loss function are defined as follows:

$$f_r^3(h,r,t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2 \tag{19}$$

4.5 Joint Training

We define a margin based ranking loss for Type Enhanced Component:

$$\mathrm{L}_1 = \sum_{(e,c)\in I_e} \sum_{(e,c')\notin I_e} \|\gamma_1 + f_r^1(e,c) - f_r^2(e,c')\|_+ \tag{20}$$

where $[x]_+ \triangleq max(0,x)$ and γ_1 is the margin separating positive $\texttt{rdf:type}$ relation and negative $\texttt{rdf:type}$ relation. Similarly, for Specific-relation Type Constrains Based Embedding Component, we will have a ranking loss:

$$L_2 = \sum_{(h,r,t)\in T} \sum_{(h',r,t')\in T'} [\gamma_2 + f_r^3(h,r,t) - f_r^3(h',r,t')] \tag{21}$$

Finally, we define the overall loss function as linear combinations of these two functions:

$$L = \beta L_1 + L_2 \tag{22}$$

where β which is a hyper-parameter represents the weight of Type Enhanced Component. The goal of training model is to minimize the above function, and iteratively update embeddings of types, entities, and relations.

5 EXPERIMENTS

We evaluate our method on two typical tasks commonly used in knowledge graph embedding: link prediction [6] and triple classification [32].

Dataset	Rel	Ent	Train	Valid	Test
FB15K	1,345	14,951	483,142	50,000	59,071
FB15K-237	237	14,541	271,115	17,535	20,466

Table 2: Statistics of datasets.

5.1 Datasets and Experiment Settings

Datasets

The experiments are conducted on two standard benchmarks used in related works: FB15k and FB15k-237, which are both extracted from Freebase. Although the FB15K has been used as a standard dataset in many previous knowledge representation learning methods, Toutanova et al. [35] found that 81% of the triples in the FB15K test set can be obtained by reversible relationship inference from the triples of the training set. The experimental results obtained by using such a dataset as an experimental dataset are not very convincing, because the knowledge representation learning method that can achieve good performance on such a dataset is likely to be good at modeling reciprocal relationships but not able to represent the complex knowledge graph in the real world.

Therefore, Toutanova et al.[35] avoid this problem by constructing a subset of the FB15K, that is, the FB15K-237 dataset. If the set of head and tail entity pairs corresponding to the two relationships are exactly the same or completely opposite, then they will be considered as a repetitive or reciprocal relationship. After this treatment, the number of relationships has dropped to 237. At the same time, the training set, validation set, and test set are also updated. Only the triples containing these 237 relationships are retained in the FB15K-237 dataset. Table 2 gives a summary of these datasets.

Experiment Settings

The experimental comparison methods of the proposed model include some typical translation-based models, namely: TransE[6], TransH[38], TransR[24] , TransD[21], RotatE [34] and three typical semantic matching models including RESCAL[28], DistMult[42] and HolE[27]. In evaluation, most of the experimental data of the comparison method are derived from the published results of TransD and HolE. They are trained with the best parameters reported in their papers. For other baselines including RESCAL and HolE, we directly use the code released in OpenKE [16] which is an open-source framework for knowledge embedding.

Every triple in our training set has a label to indicate whether the triple is positive or negative. But existing knowledge graph only contains positive triples. We need to generate negative triples by corrupting positive triples. For a relational triple (h, r, t), we replace h or t to generate a negative triple (h', r, t) or (h, r, t'). For the Type Enhanced Component, we create negative `rdf:type` triples by randomly selecting a type that does not have `rdf:type` relationship with the entity to replace the current type in the triple. For the Embedding Component, negative triples are generated by using the entity which satisfy the relation type constrains to replace h or t. We use the "bern" strategy described in [38] to replace baselines' head entities and tail entities for its rationality.

We train the proposed model with mini-batch SGD. As for parameters, we select the batch size B among $\{20, 240, 1200, 4800\}$, and margin among $\{0.5, 1.0, 1.5, 2.0\}$. We also set the dimensions of entity and relation to be the same n. For learning rate, a fixed rate was selected following (Bordes et al. 2013). The optimal configurations of our models are: $B = 20$, $\omega = 1.0$, $\lambda = 0.001$. For fair comparison, all models are trained under the same dimension $n = 100$.

5.2 Link Prediction

Evaluation

Link Prediction aims to predict the missing h or t for a relational triple (h, r, t), i.e., predict t given (h, r) or predict h given (r, t). Rather than requiring one best answer, this task emphasizes more on ranking a set of candidate entities from the knowledge graph. We follow the same protocol in TransE: for each testing triple (h, r, t), we replace the tail t by every entity $e \in E$ in the knowledge graph and calculate a dissimilarity score (according to the scoring function f_r^3) on the corrupted triple (h, r, e). Ranking the scores in ascending order, we then get the rank of the original correct triple. Similarly, we can get another rank for (h, r, t) by corrupting the head h. Let $rank_h(h, r, t)$ be the ranking of (h, r, t) among all head corrupted relations and $rank_t(h, r, t)$ denotes a similar ranking with tail corruptions.

We use two evaluation metrics: Mean Reciprocal Rank (MRR) and Hits@N. MRR is the mean of the reciprocal rank:

$$\text{MRR} = \frac{1}{2 * |T|} \sum_{(h,r,t) \in T} \frac{1}{\text{rank}_h(h, r, t)} + \frac{1}{\text{rank}_t(h, r, t)} \tag{23}$$

Hits@N measures the proportion of triples in T that rank among top t after corrupting both heads and tails. The above is called the "raw" setting. Notice that if a corrupted triple exists in the knowledge graph, ranking it before the test triple will have a bad effect on the experimental results. To eliminate this factor, we remove those corrupted triples which exist in either train, valid, or test set before getting the rank of each test triple. This setting is called "filter".

Results

The experimental results are shown in Table 3. On the FB15K dataset, the experimental results of the proposed model are improved compared to the all classic knowledge representation learning baseline model such as TransE, TransH, TransR and so on. The proposed model increases by 3.3% and 0.2% on the "filter" setting and "raw" setting of MRR respectively; And on Hits@1 and Hits@3, it also increased by 3.2% and 1.3% respectively. On the FB-15K237 dataset, the proposed model are also improved compared to the baslines except TransR. From the results, we observe that:

(1) Our proposed model significantly outperforms all baselines in terms of both MRR and Hits@N on FB15k dataset and only lower than TransR on FB15K-237 datasets, which indicates that the Attention Mechanism Enhanced by Prior Probability can successfully encodes the type information into entity and relation embeddings and could improve the representation learning of knowledge graphs.

(2) The prior probability in the data set is greatly affected by the size of the dataset. According to the previous analysis of the two data sets FB15K and FB15K-237, we can know that with the amount of data decreasing, the type's prior probability information of the entity is not particularly significant which makes the model not very good at processing the FB15K-237 dataset.

(3) The "unif" sampling trick and the "bern" sampling trick works well for FB15K.

	FB-15K					FB15K-237				
	MRR		Hits@N(%)			MRR		Hits@N(%)		
Model	**Filter**	**Raw**	**1**	**3**	**10**	**Filter**	**Raw**	**1**	**3**	**10**
TransE	0.463	0.222	29.7	57.8	74.9	0.267	0.172	17.6	30.1	46.5
TransH	0.416	0.255	28.7	49.2	64.4	0.262	0.170	17.1	29.5	44.5
TransR	0.346	0.198	21.8	40.4	58.2	**0.286**	**0.182**	**19.4**	**32.0**	**46.7**
TransD	0.405	0.251	21.5	47.9	**77.3**	0.265	0.173	17.4	29.7	44.5
RESCAL	0.354	0.189	23.5	40.9	58.7	0.202	0.142	13.4	22.0	33.8
HolE	0.524	0.232	40.9	61.3	73.9	0.230	0.131	16.4	25.1	35.9
Mine(unif)	0.544	0.242	42.5	61.4	76.6	0.241	0.125	15.2	26.5	42.3
Mine(bern)	**0.557**	**0.257**	**44.1**	**62.6**	**77.3**	0.274	0.143	18.7	29.9	44.9

Table 3: Evaluation results on link prediction. Best results are in bold.

(4) The negative sample generation adopts the "bern" setting experimental effect to be better.

5.3 Triple Classification

Evaluation

Triple classification is to determine whether a given triple (h, r, t) is correct or not, which is a binary classification task. Negative triples are required for the evaluation of triple classification. Hence, we construct some negative triples following the same setting in [32]. There are as many true triples as negative triples in both valid and test set.

The classification strategy is conducted as follows: We set different relation-specific thresholds r for each relation. For a triple (h, r, t), if the dissimilarity score of $f_r(h, t)$ is below δ_r, the triple is then predicted to be positive and otherwise negative. The relation-specific thresholds δ_r are optimized by maximizing the classification accuracies in all triples with the corresponding r on the valid sets.

Results

Similar to the link prediction experiment setting, the baselines still use the "bern" setting in the negative sampling stage. As shown in the Figure 4, it can be found that the proposed model has achieved the best experimental results in the FB15K dataset but not in the FB15K-237 dataset, and the experimental effect of the bern setting is better in the negative sampling stage.

0.51.9Analyzing the reasons for the good performance on the FB15K dataset but poor on the FB15K-237 dataset, the prior probability in the dataset is greatly affected by the size of the dataset. According to Jacob Bernoulli's Law of Large Numbers[1], in a large number of repeated occurrences of random events, an almost inevitable law is often present. Under the condition of the same experiment, repeat the experiment many times, and the frequency of random events is close to its probability. Only when the scale of dataset is large enough, the frequency of occurrence of a certain type of an entity can be expressed as the weight of the entity's performance of that type.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel model for representation learning of knowledge graphs based on attention mechanism enhanced by prior-probability. More specifically, each type in knowledge graph is encoded into a sphere in the type space and each entity is encoded as a vector in the same space. The Euclidean distance between the entity and the center of the type sphere is calculated by attention mechanism enhanced by prior-probability, which can be described as the weigh information. The less the distance between entity and type is, the larger weigh between entity and type. Then we integrate relation type constrain into embedding component by creating projection matrix related to specific-relation type constrain and concatenating the entity and its type. In experiments, we evaluate our model on two typical tasks including link prediction and triple classification. Experimental results show that the prior-probability of type information implicited in the dataset and the attention mechanism are helpful for representation learning of knowledge graphs, and the proposed embedding component is capable of encoding type information and relation type constrains into KG embeddings.

	Accuracy(%)	
Model	**FB-15K**	**FB15K-237**
TransE	79.8	77.5
TransH	79.9	79.9
TransR	82.1	82.1
TransD	88.0	**85.4**
RESCAL	73.1	71.0
HolE	77.4	75.2
Mine(unif)	80.6	79.2
Mine(bern)	**88.7**	81.6

Table 4: Evaluation results on triple classification. Best results are in bold.

We will explore the following research directions in future:

(1) The proposed model only considers type information into representation learning of KGs, while there is rich information in the form of images and texts which could be integrated to our model. We will explore the advantages of those rich information in future.

(2) The radius of type sphere is randomly initialized and learned by the model. However, the radius has rich semantic information that we have ignored. For the type with larger semantic scope such as *things*, *people* and *animals*, its radius should be larger than the radius of *furniture*, *Chinese* and *birds*. In the future we can make full use of semantic information of radius.

REFERENCES

[1] Leonard E Baum and Melvin Katz. 1965. Convergence rates in the law of large numbers. *Trans. Amer. Math. Soc.* 120, 1 (1965), 108–123.

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1533–1544.

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (Vancouver, Canada) *(SIGMOD '08)*. Association for Computing Machinery, New York, NY, USA, 1247–1250. https://doi.org/10.1145/1376616.1376746

[4] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676* (2014).

[5] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 94, 2 (2014), 233–259.

[6] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*. 1–9.

[7] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 165–180.

[8] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24.

[9] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*. 121–124.

[10] Danica Damljanovic and Kalina Bontcheva. 2012. Named entity disambiguation using linked data. In *Proceedings of the 9th extended semantic web conference*. 231–240.

[11] Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2016. Lifted rule injection for relation embeddings. *arXiv preprint arXiv:1606.08359* (2016).

[12] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Fang Zheng, and Edward Y Chang. 2014. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 839–849.

[13] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*. 413–422.

[14] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 84–94.

[15] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[16] Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An Open Toolkit for Knowledge Embedding. In *Proceedings of EMNLP*.

[17] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1709–1719.

[18] Larry Heck, Dilek Hakkani-Tür, and Gokhan Tur. 2013. Leveraging knowledge graphs for web-scale unsupervised semantic parsing. (2013).

[19] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*. 541–550.

[20] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. 2012. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*. 3176–3184.

[21] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*. 687–696.

[22] Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *International semantic web conference*. Springer, 640–655.

[23] 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 (2015), 167–195.

[24] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[25] Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Differentiating concepts and instances for knowledge graph embedding. *arXiv preprint arXiv:1811.04588* (2018).

[26] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 1003–1011.

[27] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[28] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*.

[29] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*. 271–280.

[30] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 74–84.

[31] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 conference of the north American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1119–1129.

[32] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*. Citeseer, 926–934.

[33] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*. 697–706.

[34] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* (2019).

[35] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*. 57–66.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[37] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[38] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[39] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. *arXiv preprint arXiv:1307.7973* (2013).

[40] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[41] Jiacheng Xu, Kan Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Knowledge graph representation with jointly structural and textual encoding. *arXiv preprint arXiv:1611.08661* (2016).

[42] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[43] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 283–292.

[44] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.

[45] Jindou Zhang and Jing Li. 2019. Enhanced knowledge graph embedding by jointly learning soft rules and facts. *Algorithms* 12, 12 (2019), 265.

[46] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y Chang, and Xiaoyan Zhu. 2012. Entity disambiguation with freebase. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Vol. 1. IEEE, 82–89.