What Affects the Performance of Models? Sensitivity Analysis of Knowledge Graph Embedding

Han Yang[1], Leilei Zhang[1], Fenglong Su[2], and Jinhui Pang[3](✉)

[1] Peking University, Beijing, China
{hyang001,zhang_leilei}@pku.edu.cn
[2] National University of Defense Technology, Changsha, Hunan, China
sufenglong18@nudt.edu.cn
[3] Beijing Institute of Technology, Beijing, China
pangjinhui@bit.edu.cn

Abstract. Knowledge graph (KG) embedding aims to embed entities and relations into a low-dimensional vector space, which has been an active research topic for knowledge base completion (KGC). Recent researchers improve existing models in terms of knowledge representation space, scoring function, encoding method, etc., have achieved progressive improvements. However, the theoretical mechanism behind them has always been ignored. There are few works on **sensitivity analysis** of embedded models, which is extremely challenging. The diversity of KGE models makes it difficult to consider them uniformly and compare them fairly. In this paper, we first study the internal connections and mutual transformation methods of different KGE models from the generic group perspective, and further propose a unified KGE learning framework. Then, we conduct an in-depth sensitivity analysis on the factors that affect the objective of embedding learning. Specifically, in addition to the impact of the embedding algorithm itself, this article also considers the structural features of the dataset and the strategies of the training method. After a comprehensive experiment and analysis, we can conclude that the Head-to-Tail rate of datasets, the definition of model metric function, the number of negative samples and the selection of regularization methods have a greater impact on the final performance.

Keywords: Knowledge graph embedding · Group theory · Sensitivity analysis

1 Introduction

Knowledge Graphs (KGs) have emerged as a core abstraction for incorporating human knowledge into intelligent systems, which become increasingly popular in various downstream tasks including semantic search [2,28], question answering [1,8], and recommendation system [26,31]. In general, a KG can be seen as a collection of triple facts

H. Yang and L. Zhang—Contributed equally to this research. This work was supported by the National Key RD Program of China under Grant No. 2020AAA0108600.

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13245, pp. 698–713, 2022.
https://doi.org/10.1007/978-3-031-00123-9_55

in the triple format, expressed as *(head entity, relation, tail entity)* also abbreviated as *(h, r, t)*, e.g., (Donald Trump, presidentOf, USA). Knowledge graph embedding aims to compress both relation and entity into continuous low-dimensional embedding spaces while preserving the intrinsic graph properties and its underlying semantic information. These approaches provide a way to perform reasoning in KGs with simple numerical computation in continuous spaces.

Recent years witnessed tremendous research efforts on the KGE models, which can be roughly divided into translation-based models, bilinear models, and other neural network models [25]. These models are dedicated to transferring to more complex representation spaces, designing different scoring functions or loss functions, thus making up for the shortcomings of previous works and improving the performance of embedded learning. Although more and more tailored models have shown promising performance on this task, the theoretical mechanism behind them has been much less well-understood to date. There is still a lack of comprehensive study to explore the influencing factors that lead to the improvements of the results [7,14], which can be helpful to enhance the interpretability of KGE models. In this paper, we focus on the sensitivity analysis of KGE for the first time, which is quite challenging. For one thing, the heterogeneity between different models impedes the proposal of a unified KGE abstract representation, making it difficult to compare existing models and discover which modules of the embedding algorithm lead to progress. Besides, the performance of KGE is also affected by model training, such as regularization and negative sampling method, etc. This makes the independent analysis of factors affecting model performance more complicated, especially when results are reproduced from prior studies that used a different experimental setup. Moreover, the performance of the same KGE model on different datasets may be very different, which also attracts us to analyze how the structural characteristics of the dataset influence the embedded learning objective.

To overcome the aforementioned challenges, we express the popular KGE models into a unified form, that is, a metric space based on the Abelian group. Based on group isomorphism, we further analyzed typical KGE models such as TorusE [6] , RotatE [20], DisMult [29], ComplEx [23], and proposed that they can all be regarded as variants of TransE [3] in terms of metrics, and completely different types of KGE learning algorithms can also be converted to each other. We choose circle group in Sect. 3.3 to illustrate the relation between them in detail and intuitively.

Moreover, we conduct a systematical sensitivity analysis of KGE from three aspects: dataset characteristics, model architecture, and model training. Through statistics and analysis of the structural characteristics of the dataset, we innovatively pointed out that the Head-to-Tail rate will have a significant impact on the effect of KGE models in the knowledge graph completion task. For the KGE model architecture itself, we quantitatively proved the limitations of the commonly used Euclidean metric function, and discussed the impact of model hyperparameters based on the unified KG representation learning framework. We also conduct a large number of experiments by changing the training strategy in common experimental settings to quantify and summarize the impact of different training methods on model performance. Surprisingly, we discovered that the number of negative samples, whether to perform regularization and the choice of the regularization method are important to the embedding effect, but the negative sampling method does not matter. The conclusion of KGE sensitivity analysis is quite helpful to improve the existing KGE models.

To sum up, the highlights of the paper can be summarized as:

(1) To the best of our knowledge, we are the first work focusing on the sensitivity analysis of the knowledge graph embedding models, which is of great significance for improving the interpretability of representation learning.
(2) We innovatively provide a unified framework for several popular KGE models, and explored the theoretical and conversion methods between these models, which helps to fairly analyze the influence of various factors of the embedded algorithm from a new perspective.
(3) We define a variety of dataset structural features to better analyze how dataset features affect the goal of embedding learning and pointed out the impact of different training strategies on model performance.

2 Preliminaries: Knowledge Graph Embedding

Various KGE models have been proposed for the KG completion task in recent years. For a more intuitive discussion, we only review the methods that are directly related to our work, without considering the multi-modal embedding with external information.

2.1 General Architecture

Knowledge graph embedding models learn to encode a collection of factual triplets from a knowledge graph $\mathcal{G} = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ into low dimensional, continuous vectors $(\mathbf{h}, \mathbf{r}, \mathbf{t})$, where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^k$ and $\mathbf{r} \in \mathbb{R}^d$. Typical KGE approaches follow a clear workflow that consist of four component:

(1) *Random Initialization.* Randomly initialize the entity and relation vectors, which generally uses an embedding lookup table to convert the sparse discrete one-hot vectors into dense distributed representations;
(2) *Scoring Function.* Define a scoring function to measure the plausibility of facts. The scoring function $s : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$ takes form $s(h, r, t) = f(h, r, t)$ and assigns scores to all potential triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where f may be either a fixed function or a parameterized function;
(3) *Interaction Mechanism.* Design the interaction mechanism to model the interactions of entities and relations to compute the matching score of a triple. The most popular interaction mechanisms include linear or bilinear models, factorization models, and neural networks. This is the main component of a model;
(4) *Training Strategy.* Training the KGE model by maximizing the confidence of triples, with training strategies such as negative sampling and regularization.

2.2 KGE Models

Based on the scoring function and adopted interaction mechanism, we roughly divide previous work into translation-based models, bilinear models and other models.

Translation-Based Model. These models are known for their simplicity and efficiency, which measure the plausibility of a triple as the distance between the head entity and the tail entity. The scoring functions of translation-based models usually adopt L^1 or L^2 distance as the distance metric. Taking TransE [3] as an example, the scoring function is:

$$f_r(h,t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p = -\left(\sum_{d}^{D} |\mathbf{h}_d + \mathbf{r}_d - \mathbf{t}_d|^p\right)^{1/p}, \tag{1}$$

where $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ are the embeddings of h, r, t, respectively, p is the order of Minkowski metric, such as taxicab distance is 1 and Euclidean distance is 2. D is the dimension size of the embedding space, $x = (x_1, x_2, ..., x_D)$ is a point in D-dimensional space.

TransE is the seminal work for translation-based model, which interprets relation as a translation vector $\mathbf{r}$ so that entities can be connected, formally as $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. The follow-up variants of TransE are proposed to overcome the flaws of TransE in dealing with 1-to-N, N-to-1, and N-to-N relations, such as TransH [12] and TransR [12]. TransD [9] and TranSparse [10] simplify the projection matrices, while TorusE [5] and ManifoldE [27] introduce other representation spaces. RotatE [20] defines each relation as a rotation from the head entity to the tail entity in the complex vector.

Bilinear Models. These models, also known as semantic models, use the scoring function in the form of trilinear product between entities and relations to measure the semantic similarity.

The most classical representative method is the RESCAL [18] model, which represents KG as a three-way tensor, then DistMult [29] simplifies RESCAL by restricting relation matrices to be diagonal, HolE [17] further combines the expressive power of RESCAL with the efficiency and simplicity of DistMult. ComplEx [23] entends HolE to the complex space so as to better model asymmetric relations. The analogical embedding framework [13] restricts the embedding dimension and scoring function, thus it can recover or equivalently obtain several models.

Other Models. Traditional translation-based and bilinear models cannot meet the requirements of KGE, there are some works proposed to obtain better and more effective entity and relation embeddings. QuatE [32] takes advantage of quaternion representations to enable rich interactions between entities and relations. ConvE [4] is the first work to use the convolutional neural network (CNN) framework for KG completion. In addition, we notice that some models are significantly better than other KGE models, such as ConvKB [16], CapsE [24], KBAT [15], etc. Recent work [21] has pointed out that their outstanding performances are caused by containing a large number of identical scores. Coper-ConvE [19] only conducted tail entity prediction experiments, which are simpler than head entity prediction, thus it is unfair to them with other models. They will not be discussed in this article.

3 A Unified Knowledge Graph Embedding Framework

In this section, we present a detailed theoretical analysis of the popular and typical KG representation learning models, such as TransE, TorusE, DisMult, ComplEx, RotatE, which have achieved competitive results. We summarized the above five models into the metric space based on the Abelian group, and further discussed the influence of metric methods and group operations on the KGE model performance in Sect. 5.3.

3.1 Abelian Group and Metric Space

Abelian Group: An abelian group, also called a commutative group, is a set, G, together with an operation $*$ that combines any two elements a and b of G to form another element of G, denoted $a * b$. For all a, b in an abelian group, the set and operation,$(G, *)$,

$$a * b = b * a. \tag{2}$$

Metric Space: A metric space is an ordered pair (G, d) where G is a set and d is a metric on G, i.e., a function $d : G \times G \rightarrow \mathbb{R}$ such that for any $x, y, z \in G$, the following holds:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$.

3.2 Group Representation of KGE Models

Following the definition of Abelian group and metric space, we transfer the process of knowledge graph embedding models into a three-state workflow on the group space:

(1) The group operation of the head entity h and the relation r on the abelian group G, aiming at generating a target characteristic $\tilde{t}$ in the group:

$$\tilde{t} = h * r, h, r \in G. \tag{3}$$

(2) Calculate the distance between the generated target characteristic $\tilde{t}$ and the ground-truth tail entity t on the metric space $< G, * >$.

$$d(\tilde{t}, t), d : G \times G \rightarrow \mathbb{R}. \tag{4}$$

(3) Design the loss function $F(d)$ and use it to train the whole KGE model.

We discuss the characteristics of the several selected typical models and further summarize their group representations in Table 1. TransE interprets relation as a translation vector r, formally, it calculates the distance between the characteristics of entity $\tilde{t}$ and the tail entity t in the metric space $< R^n, *, d >$, where d is the Euclidean distance. ComplEx and RotatE belong to the semantic model and the translation-based rotation model respectively, but they are highly similar from the perspective of group representation, both of them act in the high-dimensional complex number field $\mathbb{C}^n$ and perform

Table 1. The transformations and metrics of several typical KGE models

Models	Scoring function	Group operation	Metric	Transformation function
TransE	$-\ \|\mathbf{h}+\mathbf{r}-\mathbf{t}\|$	+	Euclidean	–
TorusE	$-\min_{(x,y)\in([h]+[r])\times[t]} \|\mathbf{x}-\mathbf{y}\|_i$	+(Lie Group)	Euclidean	$0.5*\sin(2*\pi*(\mathbf{h}+\mathbf{r}-\mathbf{t})+1.5\pi)$ $+0.5 \sim \sin(\mathbf{h}+\mathbf{r}-\mathbf{t})$
DisMult	$-\sum_{k=1}^{K} \mathbf{h}_k \mathbf{r}_k \mathbf{t}_k$	*	Dot product	$e^{(\mathbf{h}+\mathbf{r}-\mathbf{t})}$
ComplEx	$-\ \mathrm{Re}\left(\sum_{k=1}^{K} \mathbf{r}_k \mathbf{h}_k \overline{\mathbf{t}}_k\right)$	* (Complex)	Dot product	$\cos(\mathbf{h}+\mathbf{r}-\mathbf{t})$
RotatE	$-\ \|\|\mathbf{h}\circ\mathbf{r}-\mathbf{t}\|\|$	* (Complex)	Euclidean	$0.5 * \sin(0.5(\mathbf{h}+\mathbf{r}-\mathbf{t}))$

group operations such as complex multiplication. The difference between them is the metric function in the metric space. ComplEx applies the inner product of two vectors, while RotatE uses Euclidean distance. TorusE performs Lie group addition operation in the multidimensional torus $\mathbb{T}^n$, then computes the distance between the characteristic entity and the real entity by Euclidean distance. DisMult performs the Hadamard product of vectors in $\mathbb{R}^n$, and then measures the characteristic entity and the tail entity through the inner-product operation of vectors.

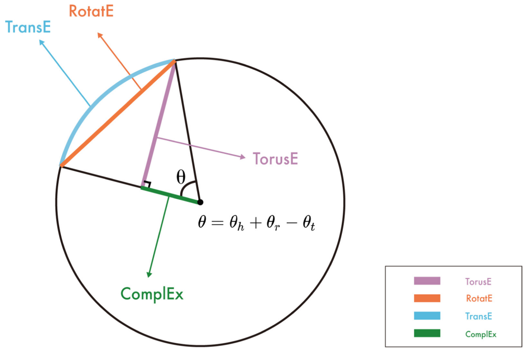

Fig. 1. The unification of KGE models.

3.3 Model Transformation and Unification

We have shown that there is a conditional isomorphism of models, i.e. a kind of mapping relation that maps models to a uniform representation space. In order to enhance the interpretability of the model and facilitate the sensitivity analysis in the following section, we choose the most commonly used circle group to summarize different KGE models. As shown in Fig. 1, we compare different KGE models in the circle group more vividly. The details of the transformation function are described in Table 1, and we give the proof as follow,

Theorem 1. *TransE can be represented as an angle with the size of θ, or as a arc segment(the bleu arc in Fig. 1), then TorusE, ComplEx, RotatE and DistMult can all be regarded as transformations of TransE based on trigonometric functions.*

Proof. (1) **TorusE:** By setting $d = \mathbf{h} + \mathbf{r} - \mathbf{t}$, we get $d - \lfloor d \rfloor \in [0,1)^n$. Then,

$$\begin{aligned} f(\mathbf{h},\mathbf{r},\mathbf{t}) &= \frac{1}{4}\|(2 - 2\cos(2\pi(\min(d - \lfloor d \rfloor, 1 - d + \lfloor d \rfloor)))\| \\ &= 0.5\|\sin(2\pi d + 1.5\pi)\| \\ &\sim \|\sin(\mathbf{h} + \mathbf{r} - \mathbf{t})\|. \end{aligned} \tag{5}$$

(2) **ComplEx:** By further restricting $|h_i| = |r_i| = |t_i| = C$, we can rewrite $\mathbf{h}, \mathbf{r}, \mathbf{t}$ by

$$\begin{aligned} \mathbf{h} &= Ce^{i\boldsymbol{\theta}_h} = C\cos\boldsymbol{\theta}_h + iC\sin\boldsymbol{\theta}_h \\ \mathbf{r} &= Ce^{i\boldsymbol{\theta}_r} = C\cos\boldsymbol{\theta}_r + iC\sin\boldsymbol{\theta}_r \\ \mathbf{t} &= Ce^{i\boldsymbol{\theta}_t} = C\cos\boldsymbol{\theta}_t + iC\sin\boldsymbol{\theta}_t. \end{aligned} \tag{6}$$

Then, we can get

$$\begin{aligned} f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|\operatorname{RE}(\mathbf{h} \circ \mathbf{r} \circ \bar{\mathbf{t}})\| &= C\left\|\operatorname{RE}\left(e^{i(\boldsymbol{\theta}_h + \boldsymbol{\theta}_r)} \circ e^{-i\boldsymbol{\theta}_t}\right)\right\| \\ &= C\left\|\operatorname{RE}\left(e^{i(\boldsymbol{\theta}_h + \boldsymbol{\theta}_r - \boldsymbol{\theta}_t)}\right)\right\| \\ &= C\left\|\cos\left(\boldsymbol{\theta}_h + \boldsymbol{\theta}_r - \boldsymbol{\theta}_t\right)\right\| \\ &\sim \left\|\cos\left(\boldsymbol{\theta}_h + \boldsymbol{\theta}_r - \boldsymbol{\theta}_t\right)\right\|. \end{aligned} \tag{7}$$

(3) **RotatE:** Since the transformation of the trigonometric function calculation for RotatE [20] has been given in previous work, we directly use the conclusion in this paper.

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| \sim \|\sin(\boldsymbol{\theta}_h + \boldsymbol{\theta}_r - \boldsymbol{\theta}_t)\|. \tag{8}$$

(4) **DistMult:** Prior work [30] proofed $\mathrm{Trans\,E} \cong$ DistMult $/\mathbb{Z}_2$. By further restricting $h, r, t > 0$, and then we can rewrite

$$\begin{aligned} \mathbf{h} &= e^{\boldsymbol{\theta}_h} \\ \mathbf{r} &= e^{\boldsymbol{\theta}_r}\ , \\ \mathbf{t} &= e^{\boldsymbol{\theta}_t} \end{aligned}$$

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|\mathbf{h} \cdot \mathbf{r} \cdot \mathbf{t}\| = \left\|e^{\boldsymbol{\theta}_h + \boldsymbol{\theta}_r + \boldsymbol{\theta}_t}\right\| \tag{9}$$

Let $\boldsymbol{\theta}'_t = -\boldsymbol{\theta}_t$, then we can get

$$f(\mathbf{h},\mathbf{r},\mathbf{t}) = \|e^{\boldsymbol{\theta}_h + \boldsymbol{\theta}_r + \boldsymbol{\theta}_t}\| \sim \|e^{\boldsymbol{\theta}_h + \boldsymbol{\theta}_r - \boldsymbol{\theta}'_t}\|. \tag{10}$$

□

4 Influencing Factors of Knowledge Graph Models

Sensitivity Analysis quantitatively studies the uncertainty in the output of a black-box model or system, thus can greatly enhance the interpretability of neural networks, which is still blank in the field of knowledge graph embedding. Since the performance of a KGE model is not only determined by the embedding algorithm itself, but also affected by the structural features of the experimental dataset, and various strategies adopted in the training method. We will first review these influencing factors follow, and then analyze and discuss them in detail in the next section.

4.1 Dataset Structural Features

Table 2. Definition of the structural features of KGE datasets

Definition	Description
First-level absolute features	
Number of entities	The number of entities and their proportion to the total entities
Number of relations	The number of relations and their proportion to the total entities
Entity category	The entity is divided into four categories: 1−1, 1−n, n−1, n−n
Relation category	The relation is divided into four categories: 1−1, 1−n, n−1, n−n
Secondary-level absolute features	
Head-to-Tail Rate	The ratio between the number of head entity category and the number of tail entity category
Test-train relative features	
Head-In Rate	Given a relation r in the test dataset, the proportion of its head entities appearing as head entities for the same relation r in the training dataset
Tail-In Rate	Given a relation r in the test dataset, the proportion of its tail entities appearing as tail entities for the same relation r in the training dataset
Avg-In Rate	Given a relation r in the test dataset, the proportion of its head or tail entities that occur with the same relation in the training dataset

The same KGE model performs quite differently on different datasets. It is difficult for a model with better performance on the benchmark dataset to maintain its superiority in the new dataset, which greatly limits the popularization and application of knowledge graph representation learning model in downstream tasks. This paper pioneered a variety of characteristic indicators to describe the structure of the knowledge graph dataset to further study how the dataset affects the performance of the KGE models. We introduced and described the definition of the structural characteristics of KGE datasets in Table 2. The dataset structural features include two types: the absolute structure characteristics of the dataset itself and the relative characteristics describing the relation between the test dataset and the training dataset. Among them, the absolute characteristics include not only the first-level features that can be directly obtained by statistics, such as the number of unique entities, relations, etc., but also the secondary-level features calculated based on the first-level features, such as Head-to-Tail rate, that is the rate of the number of types of head entities h to the number of types of tail entities t for a given relation r.

As for the relative characteristics describing the relation between the training dataset and the test dataset, we give three conceptual definitions: Head-In Rate, Tail-In Rate, and Avg-In Rate. Given a specific triple (h_i, r_i, t_i) which appears in the test dataset, Head-In (Tail-In) Rate refers to the probability of the head entity h_i (tail entity t_i) and relation r_i appear in the triple that contains this relation in the training test set at the same time. Avg-In Rate is the average of Head-In rate and Tail-In rate.

4.2 Embedding Algorithm

We have unified some typical embedding learning models into the form of transE with trigonometric functions. Specifically, the KGE model can be expressed in the form of the trigonometric function $A\sin(\omega*(\theta)+b)$. In this paper, we construct a new model based on the unified framework Sin E as follows:

$$f_{\sin E}=A\sin(\omega*(\mathrm{h}+\mathrm{r}-\mathrm{t})+b). \tag{11}$$

Then, we analyze the effect of three elements of the trigonometric function including amplitude, frequency, and phase on the entity linking task in Sect. 5.3.

4.3 Model Training

The most commonly used strategies for training the KE model are *negative sampling* and *regularization*.

Negative Sampling Methods: We analyzed not only the method of negative sampling, but also the influence of the number of negative sampling on the objective of embedding learning. For the negative sampling method, we mainly study the impact of three methods including uniform sampling, self-adversarial sampling, and NSCaching sampling [33] on the performance of KGE models. Uniform sampling refers to randomly selecting negative sample entities from all candidate entities. On the basis of uniform sampling, self-adv increases the weight of samples with higher scores in the same batch. NSCaching considers negative examples to be good or bad, and uses a caching mechanism to obtain high-quality negative examples. Moreover, we also analyze the impact of the number of negative samples on the performance of the representation model.

Regularization Method: Regularization method constrains the parameters to be optimized, which helps prevent over-fitting. Prior work [11] has studied the regularization method of the embedded models, and proposed the N3 regularization method. This paper further analyzes the effects of five common matrix norms on the KGE model, and compares them with the pure model that does not use regularization.
1-Norm:

$$\|\mathrm{A}\|_1=\max_{1\leq j\leq n}\sum_{i=1}^{m}|a_{ij}|. \tag{12}$$

∞-Norm:

$$\|\mathrm{A}\|_\infty=\max_{1\leq i\leq m}\sum_{j=1}^{n}|a_{ij}|. \tag{13}$$

2-Norm:

$$\|\mathrm{A}\|_2=\sqrt{\lambda_{\max}(\mathrm{A}^{\mathrm{T}}\mathrm{A})}. \tag{14}$$

Nuclear Norm(Nuc-Norm):

$$\begin{gathered}\|\mathrm{A}\|_*=\operatorname{tr}\left(\sqrt{\mathrm{A}^{\mathrm{T}}\mathrm{A}}\right)=\operatorname{tr}(\varSigma)\\ \mathrm{A}=U\sum V^{\mathrm{T}}.\end{gathered} \tag{15}$$

Fro Regularization:

$$\|\mathrm{A}\|_F=\sqrt{\operatorname{tr}(\mathrm{A}^{\mathrm{T}}\mathrm{A})}=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}^2}. \tag{16}$$

5 Sensitivity Analysis of the Influencing Factors in KGE Models

In this section, we perform experiments to revisit the contribution of the various influencing factors in KGE learning models, and hope to answer the following questions through an objective sensitivity analysis of the experimental results.

Q1: How does the dataset structure affect knowledge graph embedding learning?
Q2: How does the model architecture influence the objective of embedding learning?
Q3: What are the effects of different training strategies on the KGE models?

We first introduce the relevant datasets and evaluation tasks used in the sensitivity analysis experiment, and then discuss them in detail.

5.1 Experimental Settings

Table 3. Statistics of the datasets used in this paper.

Dataset	Train	Valid	Test	Ent	Rel
FB15k	483,1442	50,000	59071	14951	1345
FB15k-237	272,115	17,535	20,466	14,541	237
WN18	141,442	5000	5000	40,943	18
WN18RR	86,835	3,3034	3,3134	40,943	11

Experimental Dataset: We conducted experiments on some common datasets including FB15k, WN18 [3], FB15k-237 [22] and WN18RR [4]. FB15k is a subset of Freebase, a large-scale knowledge graph containing general knowledge facts. WN18 is extracted from WordNet3, where words are interlinked by means of conceptual-semantic and lexical relations. FB15k-237 and WN18RR are their corresponding updated version, with inverse relations removed. Statistics of the datasets are provided in Table 3.

Evaluation Protocol. We evaluate the performance of KGE models on the entity linking task, which predicts the missing entity in the triple by minimizing the loss function, that is, given $(h, r, ?)$, predict the tail entity ?.

$$t = \underset{e}{\text{argmin}} \cdot f(h, r, e). \tag{17}$$

We employ three popular metrics to evaluate the performance of link prediction, including Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hit ratio with cut-off values n = 1, 3, 10. MR measures the average rank of all correct entities with a lower value representing better performance. MRR is the average inverse rank for correct entities. Hit@n measures the proportion of correct entities in the top n entities.

5.2 Sensitivity Analysis of Dataset Structural Features

Table 4. The link prediction results of some relations

Relation	HeadInRate	tailInRate	AvgInRate	HeadToTailRate	ErorrRate
dated_money_value/currency	0.00000	1.00000	0.03226	553.91670	0.00000
location_of_ceremony	1.00000	0.01888	0.03704	0.00197	0.88679
food/diet/followers	1.00000	0.00000	0.11764	0.02173	0.46667
athlete_salary/team	0.25000	0.25000	0.25000	1.37500	0.25000

We perform sensitivity analysis of dataset structural characteristics on the FB15K dataset, because it has the largest amount of training triplets and contains various types of relations. As shown in Table 4, we list the characteristic statistical values and error rate of missing triples corresponding to some specific relations in the FB15K dataset. Here we use the hit@10 indicator to represent the error rate. Figure 2 shows the scatter plots of Hit@10 error rate with Avg-In rate and Head-to-Tail rate, respectively. It shows that Hit@10 error rate and Head-to-Tail rate are negatively correlated, while Avg-In Rate has little effect on Hit@10 error rate.

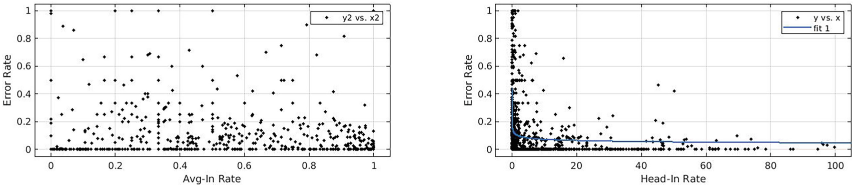

Fig. 2. Error Rate to Avg-In Rate, Head-In Rate, respectively.

Table 5. Error Rate to Head-to-Tail Rate.

HeadTailRate	Hit10 ErrorRate	NumberOfTriples	HeadTailRate	Hit10 ErrorRate	NumberOfTriples
0−0.5	0.171	568	3−3.5	0.058	44
0.5−1	0.142	218	3.5−4	0.089	29
1−1.5	0.161	273	4−4.5	0.035	37
1.5−2	0.099	66	4.5−5	0.025	42
2−2.5	0.098	97	≥5	0.029	523
2.5−3	0.084	25			

The Head-to-Tail Rate is not a uniform distribution, in order to analyze the relations between the two more intuitively, we segment the Head-to-Tail Rate with a step size of 0.5, and calculate the average error rate of all relations in the segment to make the histogram. Table 5 and Fig. 3 further confirm the conclusion that the larger the Head-to-Tail Rate, the lower the Hit@10 error rate.

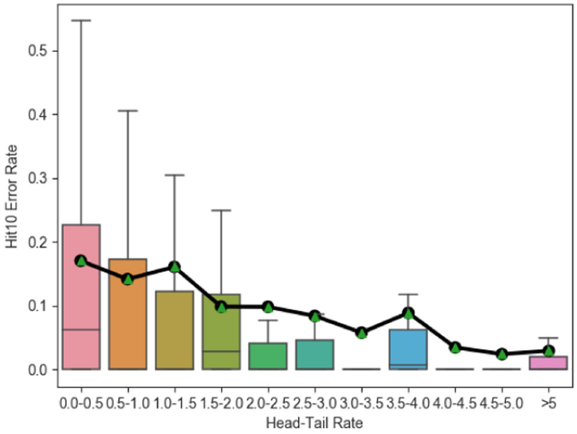

Fig. 3. Error Rate to Head-to-Tail Rate.

5.3 Sensitivity Analysis of KGE Model Architecture

We analyzed the influence of period, margin, amplitude, and phase of SinE. The results are shown in Fig. 4. Among the four factors, the period has the greatest impact. ω can be regarded as the reciprocal of the cycle. With the increase of ω, the performance of the KGE model greatly improves until it reaches the peak, then stabilizes or decreases slowly. The margin has a similar impact, but it is not as obvious as the period. The impact of amplitude on the dataset is relatively small. At first, a large range of amplitude has minimal impact on model performance, and then as the amplitude increases, the model performance gradually decreases. Phase is the parameter that has the least impact on the performance of the model among the four factors.

5.4 Sensitivity Analysis of Model Training Strategies

Table 6. Sensitivity analysis of the number of NS, where NS represents the negative sampling.

Number of NS	1	3	5	10
MRR/Hit@1	0.633/0.510	0.696/0.581	0.721/0.611	0.752/0.654
Number of NS	50	150	200	256
MRR/Hit@1	0.788/0.723	0.795/0.741	0.791/0.738	0.791/0.739

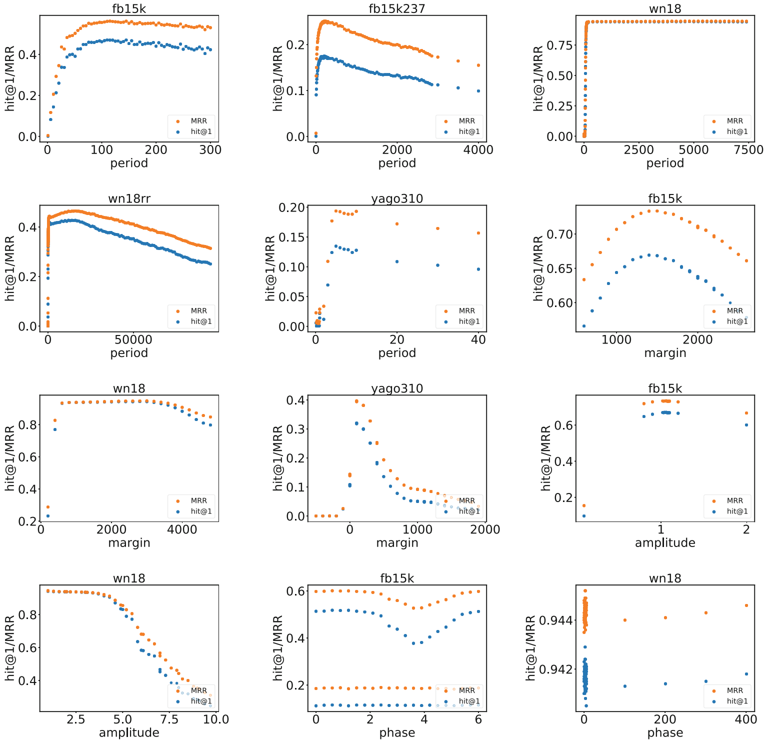

Fig. 4. Sensitivity Analysis of period, margin, amplitude, and phase of SinE.

In order to verify whether the number of negative samples has an impact on the objective of embedding learning, we use RotatE as the basic model, and adopt uniform, self-adversarial, and NSCaching negative sampling methods to train the KGE model on the FB15k dataset. At the same time, we also set the number of negative samples is 1, 3, 5, 10, 50, 100, 150, 200, and 256 to observe the influence of the number of negative samples. The experimental results are shown in Table 6. We can conclude that compared to the negative sample sampling method, the number of samples has a greater impact on downstream tasks.

In addition, We conduct experiments on the selection of negative sampling methods, and the results are shown in Table 7. We can conclude that the impact of negative sampling methods on model performance depends on the dataset. For example, in WN18RR, the uniform sampling performs better, but the performance of uniform sampling is worse in the FB15k-237 dataset. NSCaching is the opposite, with the worst effect on WN18RR, but it has the best performance on FB15k-237 among the three negative sampling methods. The performance of self-adversarial negative sampling is relatively stable on the two datasets. Therefore, this article believes that the choice of negative sampling method should be determined by the structural characteristics of the dataset. Different datasets should be equipped with different negative sampling methods.

Table 7. Sensitivity analysis of NS methods, where NS represents negative sampling.

NS method	WN18RR				FB15K237			
	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
ComplEx + random	0.449	0.387	0.486	**0.556**	0.272	0.191	0.295	0.432
ComplEx + self-adversarial	**0.470**	0.429	0.484	0.555	0.322	0.229	0.352	**0.511**
ComplEx + NSCaching	0.302	–	–	0.481	**0.446**	–	–	0.509

Table 8. Sensitivity analysis of regularization methods on FB15K and WN18 datasets.

Model	FB15K				WN18			
	MRR	H@1	H@3	Hit@10	MRR	H@1	H@3	Hit@10
ComplEx-None	0.83	0.79	0.86	0.90	0.95	0.95	0.95	0.95
ComplEx-FRO	0.84	0.80	0.86	0.90	0.95	0.95	0.95	0.96
ComplEx-N3	0.84	0.79	0.86	0.91	0.95	0.95	0.95	0.96
ComplEx-1	0.83	0.80	0.86	0.90	0.95	0.94	0.95	0.95
ComplEx-2	0.83	0.80	0.86	0.90	0.95	0.94	0.95	0.95
ComplEx-∞	0.83	0.80	0.86	0.90	0.95	0.94	0.95	0.95
ComplEx-nuc	0.83	0.79	0.86	0.90	0.95	0.94	0.95	0.95

Table 9. Sensitivity analysis of regularization methods on FB15k-237 and WN18RR datasets.

Model	FB15k-237				WN18RR			
	MRR	H@1	H@3	Hit@10	MRR	H@1	H@3	Hit@10
ComplEx-None	0.34	0.25	0.37	0.51	0.46	0.43	0.47	0.52
ComplEx-FRO	0.33	0.24	0.36	0.52	0.45	0.42	0.46	0.51
ComplEx-N3	0.35	0.26	0.38	0.53	0.47	0.43	0.48	0.53
ComplEx-1	0.34	0.25	0.37	0.51	0.46	0.43	0.47	0.51
ComplEx-2	0.33	0.25	0.37	0.51	0.45	0.43	0.46	0.51
ComplEx-∞	0.31	0.22	0.34	0.49	0.41	0.39	0.41	0.44
ComplEx-nuc	0.33	0.25	0.37	0.51	0.45	0.43	0.46	0.51

We also implement the six regularization methods mentioned in Sect. 4.3 to train the ComplEx separately, and the experimental results are shown in Table 8 and 9. After analysis, it can be found that (1). The choice of the regularization method is important to the performance of the KGE models. The N3 regularization method has achieved excellent performance on all datasets. Especially for the WN18RR dataset, except for N3, the other regularization methods lead to reduces in the performance of ComplEx, which shows that the regularization method cannot be used arbitrarily. (2). The structural features of KGE datasets also affect the influence of regularization methods. In the denser FB15K and WN18 datasets, whether to use the regularization method has less impact on the model. In the sparse FB237 and WN18 datasets, the opposite is true, which also proves that when the dataset is small, the use of regularization can prevent overfitting.

6 Conclusion

This paper first conducts sensitivity analysis to improve the interpretability of the KGE models. We give a unified representation of several typical KGE models based on TransE + trigonometric functions, and further analyze the transformation methods between them. On this basis, we concluded that the different parameters of the trigonometric function have a significant impact on the objective of embedding learning. Moreover, we discussed the effect of features of the data structure, different model implementation strategies on the KGE models. we found that the Head-to-Tail rate of datasets, the definition of model metric function, the number of negative samples and the selection of regularization methods have a greater impact on the final performance.

References

1. Abujabal, A., Yahya, M., Riedewald, M., Weikum, G.: Automated template generation for question answering over knowledge graphs. In: Proceedings of the 26th International Conference on World Wide Web (2017)
2. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid search: effectively combining keywords and semantic searches. In: European Semantic Web Conference (2008)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: (NIPS) (2013)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: AAAI (2018)
5. Ebisu, T., Ichise, R., Torus, E.: Knowledge graph embedding on a lie group. AAAI, Toruse (2018)
6. Ebisu, T., Ichise, R.: TorusE: knowledge graph embedding on a lie group. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
7. Gesese, G.A., Biswas, R., Alam, M., Sack, H.: A survey on knowledge graph embeddings with literals: which model links better literally? Semantic Web (2021)
8. Hu, S., Zou, L., Yu, J.X., Wang, H., Zhao, D.: Answering natural language questions by subgraph matching over knowledge graphs. IEEE Trans. Knowl. Data Eng. **30**, 824–837 (2017)
9. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: ACL (2015)
10. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: AAAI (2016)
11. Lacroix, T., Usunier, N., Obozinski, G.: Canonical tensor decomposition for knowledge base completion. In: International Conference on Machine Learning (2018)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI (2015)
13. Liu, H., Wu, Y., Yang, Y.: Analogical inference for multi-relational embeddings. In: International conference on machine learning. PMLR (2017)
14. Mohamed, S.K., Novácek, V., Vandenbussche, P.Y., Muñoz, E.: Loss functions in knowledge graph embedding models. In: DL4KG@ ESWC (2019)

15. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. arXiv preprint arXiv:1906.01195 (2019)
16. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121 (2017)
17. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: AAAI (2016)
18. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
19. Stoica, G., Stretcu, O., Platanios, E.A., Mitchell, T., Póczos, B.: Contextual parameter generation for knowledge graph link prediction. In: AAAI (2020)
20. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2018)
21. Sun, Z., Vashishth, S., Sanyal, S., Talukdar, P., Yang, Y.: A re-evaluation of knowledge graph completion methods. In: ACL (2020)
22. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality (2015)
23. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning. PMLR (2016)
24. Vu, T., Nguyen, T.D., Nguyen, D.Q., Phung, D., et al.: A capsule network-based embedding model for knowledge graph completion and search personalization. In: NAACL (2019)
25. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**, 2724–2743 (2017)
26. Wang, X., Wang, D., Xu, C., He, X., Cao, Y., Chua, T.S.: Explainable reasoning over knowledge graphs for recommendation. In: AAAI (2019)
27. Xiao, H., Huang, M., Zhu, X.: From one point to a manifold: knowledge graph embedding for precise link prediction. arXiv preprint arXiv:1512.04792 (2015)
28. Xiong, C., Power, R., Callan, J.: Explicit semantic ranking for academic search via knowledge graph embedding. In: WWW (2017)
29. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
30. Yang, H., Liu, J.: Knowledge graph representation learning as groupoid: unifying TransE, RotatE, QuatE, ComplEx. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management (2021)
31. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: SIGKDD (2016)
32. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. Adv. Neural Inf. Process. Syst. **32** (2019)
33. Zhang, Y., Yao, Q., Shao, Y., Chen, L.: NSCaching: simple and efficient negative sampling for knowledge graph embedding. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 614–625. IEEE (2019)