SEEK: Segmented Embedding of Knowledge Graphs

Wentao Xu[1], **Shun Zheng**[2], **Liang He**[2], **Bin Shao**[2], **Jian Yin**[1]*, **and Tie-Yan Liu**[2]
[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China;
Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China
[2] Microsoft Research Asia, Beijing, China
{xuwt6@mail2,issjyin@mail}.sysu.edu.cn
{Shun.Zheng,Liang.He,binshao,tyliu}@microsoft.com

Abstract

In recent years, knowledge graph embedding becomes a pretty hot research topic of artificial intelligence and plays increasingly vital roles in various downstream applications, such as recommendation and question answering. However, existing methods for knowledge graph embedding can not make a proper trade-off between the model complexity and the model expressiveness, which makes them still far from satisfactory. To mitigate this problem, we propose a lightweight modeling framework that can achieve highly competitive relational expressiveness without increasing the model complexity. Our framework focuses on the design of scoring functions and highlights two critical characteristics: 1) facilitating sufficient feature interactions; 2) preserving both symmetry and antisymmetry properties of relations. It is noteworthy that owing to the general and elegant design of scoring functions, our framework can incorporate many famous existing methods as special cases. Moreover, extensive experiments on public benchmarks demonstrate the efficiency and effectiveness of our framework. Source codes and data can be found at https://github.com/Wentao-Xu/SEEK.

1 Introduction

Learning embeddings for a knowledge graph (KG) is a vital task in artificial intelligence (AI) and can benefit many downstream applications, such as personalized recommendation (Zhang et al., 2016; Wang et al., 2018) and question answering (Huang et al., 2019). In general, a KG stores a large collection of entities and inter-entity relations in a triple format, (h, r, t), where h denotes the head entity, t represents the tail entity, and r corresponds to the relationship between h and t. The goal of knowledge graph embedding (KGE) is to project massive interconnected triples into a low-dimensional space and preserve the initial semantic information at the same time.

Although recent years witnessed tremendous research efforts on the KGE problem, existing research did not make a proper trade-off between the model complexity (the number of parameters) and the model expressiveness (the performance in capturing semantic information). To illustrate this issue, we categorize existing research into two categories.

The first category of methods prefers the simple model but suffers from poor expressiveness. Some early KGE methods, such as TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015), fell into this category. It is easy to apply these methods to large-scale real-world KGs, but their performance in capturing semantic information (such as link prediction) is far from satisfactory.

In contrast, the second category pursues the excellent expressiveness but introduces much more model parameters and tensor computations. Typical examples include TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015), Single DistMult (Kadlec et al., 2017), ConvE (Dettmers et al., 2018) and InteractE (Vashishth et al., 2019). However, as pointed out by Dettmers et al. (2018), the high model complexity often leads to poor scalability, which is prohibitive in practice because real-world KGs usually contain massive triples.

To address these drawbacks of existing methods, in this paper, we propose a light-weight framework for KGE that achieves highly competitive expressiveness without the sacrifice in the model complexity. Next, we introduce our framework from three aspects: 1) facilitating sufficient feature interactions, 2) preserving various necessary relation properties, 3) designing both efficient and effective scoring functions.

*Corresponding author.

First, to pursue high expressiveness with the reasonable model complexity, we need to facilitate more sufficient feature interactions given the same number of parameters. Specifically, we divide the embedding dimension into multiple segments and encourage the interactions among different segments. In this way, we can obtain highly expressive representations without increasing model parameters. Accordingly, we name our framework as Segmented Embedding for KGs (SEEK).

Second, it is crucial to preserve different relation properties, especially the symmetry and the antisymmetry. We note that some previous research did not preserve the symmetry or the antisymmetry and thus obtained inferior performance (Bordes et al., 2013; Lin et al., 2015; Yang et al., 2015). Similar to the recent advanced models (Trouillon et al., 2016; Kazemi and Poole, 2018; Sun et al., 2019; Xu and Li, 2019), we also pay close attention to the modeling support of both symmetric and antisymmetric relationships.

Third, after an exhaustive review of the literature, we find that one critical difference between various KGE methods lies in the design of scoring functions. Therefore, we dive deeply into designing powerful scoring functions for a triple (h, r, t). Specifically, we combine the above two aspects (facilitating feature interactions and preserving various relation properties) and develop four kinds of scoring functions progressively. Based on these scoring functions, we can specify many existing KGE methods, including DistMult (Yang et al., 2015), HoIE (Nickel et al., 2016), and ComplEx (Trouillon et al., 2016), as special cases of SEEK. Hence, as a general framework, SEEK can help readers to understand better the pros and cons of existing research as well as the relationship between them. Moreover, extensive experiments demonstrate that SEEK can achieve either state-of-the-art or highly competitive performance on a variety of benchmarks for KGE compared with existing methods.

In summary, this paper makes the following contributions.

- We propose a light-weight framework (SEEK) for KGE that achieves highly competitive expressiveness without the sacrifice in the model complexity.

- As a unique framework that focuses on designing scoring functions for KGE, SEEK combines two critical characteristics: facilitating sufficient feature interactions and preserving fundamental relation properties.

- As a general framework, SEEK can incorporate many previous methods as special cases, which can help readers to understand and compare existing research.

- Extensive experiments demonstrate the effectiveness and efficiency of SEEK. Moreover, sensitivity experiments about the number of segments also verify the robustness of SEEK.

2 Related Work

We can categorize most of the existing work into two categories according to the model complexity and the model expressiveness.

The first category of methods is the simple but lack of expressiveness, which can easily scale to large knowledge graphs. This kind of methods includes TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015). TransE uses relation r as a translation from a head entity h to a tail entity t for calculating their embedding vectors of (h, r, t); DistMult utilizes the multi-linear dot product as the scoring function.

The second kind of work introduces more parameters to improve the expressiveness of the simple methods. TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015), and ITransF (Xie et al., 2017) are the extensions of TransE, which introduce other parameters to map the entities and relations to different semantic spaces. The Single DistMult (Kadlec et al., 2017) increases the embedding size of the DistMult to obtain more expressive features. Besides, ProjE (Shi and Weninger, 2017), ConvE (Dettmers et al., 2018) and InteractE (Vashishth et al., 2019) leverage neural networks to capture more feature interactions between embeddings and thus improves the expressiveness. However, these neural network-based methods would also lead to more parameters since there are many parameters in the neural network. Although the second kind of methods has a better performance compared with simple methods, they are difficult to apply to real-world KGs due to the high model complexity (a large number of parameters).

Compared with the two types of methods above, our SEEK can achieve high expressiveness without increasing the number of model parameters.

Methods	Scoring Function	Performance	# Parameters	Properties	
				Sym	Antisym
TransE (Bordes et al., 2013)	$\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|$	Low	Small	✗	✓
DistMult (Yang et al., 2015)	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	Low	Small	✓	✗
ComplEx (Trouillon et al., 2016)	$Re(\langle \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} \rangle)$	Low	Small	✓	✓
Single DistMult (Kadlec et al., 2017)	$\langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$	High	Large	✓	✗
ConvE (Dettmers et al., 2018)	$f(\text{vec}(\text{f}([\mathbf{h}, \mathbf{r}] * \omega))\text{W})\mathbf{t}$	High	Large	✗	✓
SEEK	$\sum s_{x,y} \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_{w_{x,y}} \rangle$	High	Small	✓	✓

Table 1: Comparison between our SEEK framework and some representative knowledge graph embedding methods in the aspects of the scoring function, performance, the number of parameters, and the ability to preserve the symmetry and antisymmetry properties of relations.

Table 1 shows the comparison between our framework and some representative KGE methods in different aspects.

Besides, preserving the symmetry and antisymmetry properties of relations is vital for KGE models. Many recent methods devote to preserving these relation properties to improve the expressiveness of embeddings (Trouillon et al., 2016; Nickel et al., 2016; Guo et al., 2018; Ding et al., 2018; Kazemi and Poole, 2018; Sun et al., 2019; Xu and Li, 2019). Motivated by these methods, we also pay attention to preserving symmetry and antisymmetry properties of relations when we design our scoring functions.

3 SEEK

Briefly speaking, we build SEEK by designing scoring functions, which is one of the most critical components of various existing KGE methods, as discussed in the related work. During the procedure of designing scoring functions, we progressively introduce two characteristics that hugely contribute to the model expressiveness: 1) facilitating sufficient feature interactions; 2) supporting both symmetric and antisymmetric relations. In this way, SEEK enables the excellent model expressiveness given a light-weight model with the same number of parameters as some simple KGE counterparts, such as TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015).

3.1 Scoring Functions

In this section, we illustrate our four scoring functions progressively.

3.1.1 f_1: Multi-linear Dot Product

First, we start with the scoring function f_1 developed by Yang et al. (2015), which computes a multi-linear dot product of three vectors:

DistMult

$$f_1(h, r, t) = \langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle = \sum_i r_i \cdot h_i \cdot t_i, \qquad (1)$$

where $\mathbf{r}$, $\mathbf{h}$, and $\mathbf{t}$ are low-dimensional representations of the relation r, the head entity h, and the tail entity t, respectively, and r_i, h_i, and t_i correspond to the i-th dimension of $\mathbf{r}$, $\mathbf{h}$, and $\mathbf{t}$, respectively.

We note that the function f_1 is the building block of much previous research (Trouillon et al., 2016; Kadlec et al., 2017; Kazemi and Poole, 2018). Different from these existing research, we focus on designing more advanced scoring functions with better expressiveness.

与现有的研究不同，我们专注于设计更先进的，拥有更好表达的评分函数

3.1.2 f_2: Multi-linear Dot Product Among Segments

细粒度

Next, we introduce fine-grained feature interactions to improve the model expressiveness further. To be specific, we develop the scoring function f_2 that conducts the multi-linear dot product among different segments of the entity/relation embeddings. First, we uniformly divide the d-dimensional embedding of the head h, the relation r, and the tail t into k segments, and the dimension of each segment is d/k. For example, we can write the embedding of relation $\mathbf{r}$ as:

$$\mathbf{r} = [\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{k-1}], \quad \mathbf{r}_x \in \mathbb{R}^{d/k},$$

where $\mathbf{r}_x$ is the x-th segment of the embedding $\mathbf{r}$.

Then, we define the scoring function f_2 as follows:

$$f_2(h, r, t) = \sum_{0 \le x, y, w < k} \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_w \rangle . \qquad (2)$$

Compared with the scoring function f_1, where the interactions only happen among the same positions of $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ embeddings, the scoring function f_2 can exploit more feature interactions among different segments of embeddings.

不同片段之间也交互

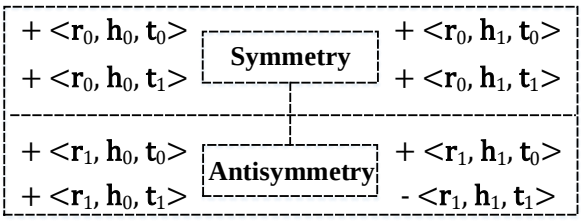

Figure 1: Scoring function f_3 with $k = 2$.

3.1.3 f_3: Modeling both Symmetric and Antisymmetric Relations

Although the scoring function f_2 can facilitate fine-grained feature interactions, it can only preserve the symmetry property of relations and can not support the modeling of antisymmetric relations. For example, given a symmetric relation r, we have $f_2(h, r, t) = f_2(t, r, h)$, but for an antisymmetric relation r', the value of $f_2(h, r', t)$ is also equal to $f_2(t, r', h)$, which is unreasonable because (t, r', h) is a false triple.

奇数-反对称

偶数-对称

To preserve the antisymmetry property of relations, we divide the segments of relation embedding $\mathbf{r}$ into odd and even parts. Then we define a variable $s_{x,y}$ to enable the even parts of segments to capture the symmetry property of relations and the odd parts to capture the antisymmetry property. We define the scoring function after adding $s_{x,y}$ as:

$$f_3(h, r, t) = \sum_{0 \le x,y,w < k} s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_w \rangle , \quad (3)$$

where

$$s_{x,y} = \begin{cases} -1, & \text{if } x \text{ is odd and } x + y \ge k, \\ 1, & \text{otherwise.} \end{cases}$$

In the scoring function f_3, $s_{x,y}$ indicates the sign of each dot product term $\langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_w \rangle$. Figure 1 depicts an example of the function f_3 with $k = 2$. When $\mathbf{r}_x$ is the even part of $\mathbf{r}$ (the index x is even), $s_{x,y}$ is positive, and the summation $\sum_{s_{x,y}==1} s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_w \rangle$ of $f_3(h, r, t)$ equals to the corresponding one $\sum_{s_{x,y}==1} s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{t}_y, \mathbf{h}_w \rangle$ of $f_3(t, r, h)$. Therefore, the function f_3 can model symmetric relations via the even segments of $\mathbf{r}$. When $\mathbf{r}_x$ is the odd part of r (the index x is odd), $s_{x,y}$ can be either negative or positive depending on whether $x + y \ge k$. Then, the summation of odd parts of $f_3(h, r, t)$ is differ from that of $f_3(t, r, h)$. Accordingly, $f_3(h, r, t)$ can support antisymmetric relations with the odd segments of $\mathbf{r}$.

The scoring function f_3 can support both symmetric and antisymmetric relations inherently because of the design of segmented embeddings.

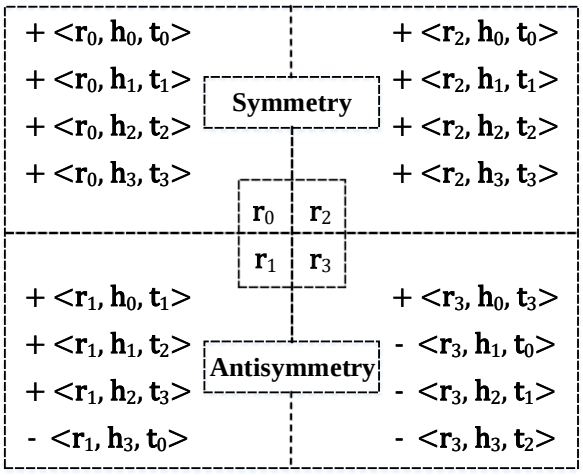

Figure 2: Scoring function f_4 with $k = 4$.

Moreover, the optimization of relation embeddings is entirely data-driven, and thus we focus on providing the proper mechanism to capture common relation properties.

3.1.4 f_4: Reducing Computing Overheads

However, though capturing various relation properties, the function f_3 suffers from huge computation overheads. The time complexity of function f_3 is $O(k^2 d)$ because there are k^3 dot product terms $\langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_w \rangle$ in total. Therefore, the scoring function f_3 needs k^3 times of dot product to compute the score of a triple (h, r, t). Recall that the dimension of each segment is d/k, so each multi-linear dot product requires $O(d/k)$ times of multiplication. As a conclusion, the time complexity of the function f_3 is $O(k^2 d)$, which can be calculated by $O(k^3 \times d/k)$. To reduce the computation overheads of the function f_3, we introduce another variable $w_{x,y}$ for the index of tail entity t. Accordingly, we define the scoring function f_4 as follows.

$$f_4(h, r, t) = \sum_{0 \le x,y < k} s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_{w_{x,y}} \rangle , \quad (4)$$

where

$$w_{x,y} = \begin{cases} y, & \text{if } x \text{ is even,} \\ (x + y) \,\%\, k, & \text{if } x \text{ is odd.} \end{cases}$$

The scoring function f_4 reduces the number of dot product terms to k^2, so its time complexity is $O(kd)$ (calculated by $O(k^2 \times d/k)$). Moreover, the scoring function f_4 can also preserve symmetry property in the even parts of $\mathbf{r}$ and preserve antisymmetry property in the odd parts of $\mathbf{r}$.

Figure 2 shows the example of the scoring function f_4 with $k = 4$. The dot product terms in

Figure 2 can be categorized into four groups according to the segment indexes of $\mathbf{r}$. In the groups of $\mathbf{r}_0$ and $\mathbf{r}_2$, which are the even parts of $\mathbf{r}$, the segment $\mathbf{t}_{w_{x,y}}$'s index $w_{x,y}$ is same as the segment $\mathbf{h}_y$'s index y, and $s_{x,y}$ is always positive. Thus, the summation $\sum s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_{w_{x,y}} \rangle$ of the even parts of $f_4(h,r,t)$ is equal to the corresponding one $\sum s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{t}_y, \mathbf{h}_{w_{x,y}} \rangle$ of $f_4(t,r,h)$. In the groups of $\mathbf{r}_1$ and $\mathbf{r}_3$, which are the odd parts of $\mathbf{r}$, the segment indexes of $\mathbf{t}$ are $(x+y)\ \%\ k$, where x and y are the indexes of $\mathbf{r}$ and $\mathbf{h}$, respectively. When $x+y \geq k$, the variable $s_{x,y}$ will change from positive to negative. So the summation of the odd parts of $f_4(h,r,t)$ and $f_4(t,r,h)$ will not be the same. Besides, it is apparent that the number of feature interactions on h, r and t are increasing k times since each segment has k interactions with other segments. k次交互

当x+y>=k时，f4头和f4尾是不一样的

In summary, the scoring function f_4 of our SEEK framework has the following characteristics:

- *Tunable Computation.* The scoring function exactly involves each segment of $\mathbf{r}$, $\mathbf{h}$, and $\mathbf{t}$ k times. Thus the number of feature interactions and the computation cost are fully tunable with a single hyperparameter k.

- *Symmetry and Antisymmetry Preservation.* The even parts of $\mathbf{r}$ can preserve the symmetry property of relations, and the odd parts of $\mathbf{r}$ can preserve the antisymmetry property.

- *Dimension Isolation.* The dimensions within the same segment are isolated from each other, which will prevent the embeddings from excessive correlations.

同一段内的维度是相互隔离的，这将防止嵌入的过度关联。

3.2 Discussions

Complexity analysis As described before, the number of dot product terms in scoring function f_4 is k^2, and each term requires $O(d/k)$ times of multiplication. So the time complexity of our SEEK framework is $O(kd)$ (calculated by $O(k^2 \times d/k)$), where k is a small constant such as 4 or 8. For the space complexity, the dimension of entity and relation embeddings is d, and there are no other parameters in our SEEK framework. Thus, the space complexity of SEEK is $O(d)$. The low time and space complexity of our framework demonstrate that our SEEK framework has high scalability, which is vital for large-scale real-world knowledge graphs.

Connection with existing methods Our SEEK framework is a generalized framework of some existing methods, such as DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), and HolE (Nickel et al., 2016). In the following, we will prove that these methods are special cases of our framework when we set $k=1$ and $k=2$, respectively.

Proposition 1. *SEEK ($k=1$) is equivalent to DistMult.*

Proof. The proof is trivial. Given $k=1$, we have $x=0$ and $y=0$ in scoring function f_4 and $\mathbf{r}_0=\mathbf{r}$, $\mathbf{h}_0=\mathbf{h}$, and $\mathbf{t}_0=\mathbf{t}$. Thus the function f_4 can be written as $f_4^{k=1}(h,r,t) = \langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$, which is the same scoring function of DistMult. □

Proposition 2. *SEEK ($k=2$) is equivalent to the ComplEx and HolE.*

Proof. Given $k=2$, function f_4 can be written as:

$$f_4^{k=2}(h,r,t) = \sum_{x=0,1} \sum_{y=0,1} s_{x,y} \cdot \langle \mathbf{r}_x, \mathbf{h}_y, \mathbf{t}_{w_{x,y}} \rangle ,$$

then we expand the right part of the equation:

$$\langle \mathbf{r}_0, \mathbf{h}_0, \mathbf{t}_0 \rangle + \langle \mathbf{r}_0, \mathbf{h}_1, \mathbf{t}_1 \rangle + \langle \mathbf{r}_1, \mathbf{h}_0, \mathbf{t}_1 \rangle - \langle \mathbf{r}_1, \mathbf{h}_1, \mathbf{t}_0 \rangle .$$

If we consider $\mathbf{r}_0, \mathbf{h}_0, \mathbf{t}_0$ as the real part of $\mathbf{r}, \mathbf{h}, \mathbf{t}$, and $\mathbf{r}_1, \mathbf{h}_1, \mathbf{t}_1$ as the imaginary part, then $f_4^{k=2}(h,r,t)$ is exactly the scoring function of ComplEx framework. Since (Hayashi and Shimbo, 2017) has already discussed the equivalence of ComplEx and HolE, the SEEK ($k=2$) is also equivalent to the HolE framework. □

3.3 Training

SEEK takes the negative log-likelihood loss function with L_2 regularization as its objective function to optimize the parameters of entities and relations:

$$\min_{\Theta} \sum_{(h,r,t) \in \Omega} -\log(\sigma(Y_{hrt} f_4(h,r,t))) + \frac{\lambda}{2d} ||\Theta||_2^2, \tag{5}$$

where σ is a sigmoid function defined as $\sigma(x) = \frac{1}{1+e^{-x}}$, and Θ represents the parameters in the embeddings of entities and relations in knowledge graphs; Ω is the triple set containing the true triples in the knowledge graphs and the false triples generated by negative sampling. In the negative sampling, we generate a false triple (h', r, t) or (h, r, t') by replacing the head or tail entity of a true triple

with a random entity. Y_{hrt} is the label of (h, r, t), which is 1 for the true triples and -1 for the false triples. λ is the L_2 regularization parameter.

The gradients of Equation 5 are then given by:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial f_4} \frac{\partial f_4}{\partial \theta} + \frac{\lambda \theta}{d}, \tag{6}$$

where $\mathcal{L}$ represents the objective function of SEEK, and θ is the parameters in the segments. Specifically, the partial derivatives of function f_4 for the x-th segment of $\mathbf{r}$ and the y-th segment of $\mathbf{h}$ are:

$$\frac{\partial f_4}{\partial \mathbf{r}_x} = \sum_{0 \le y < k} s_{x,y} \cdot (\mathbf{h}_y \odot \mathbf{t}_{w_{x,y}}),$$

$$\frac{\partial f_4}{\partial \mathbf{h}_y} = \sum_{0 \le x < k} s_{x,y} \cdot (\mathbf{r}_x \odot \mathbf{t}_{w_{x,y}}),$$

where $\odot$ is the entry-wise product of two vectors, e.g. $\mathbf{c} = \mathbf{a} \odot \mathbf{b}$ results in the i-th dimension of $\mathbf{c}$ is $\mathbf{a}_i \cdot \mathbf{b}_i$. The derivative of scoring function f_4 for $\mathbf{t}_w$ is different from those of the above two:

$$\frac{\partial f_4}{\partial \mathbf{t}_w} = \sum_{0 \le x,y < k} \mathbb{1}_{[w=w_{x,y}]} \cdot s_{x,y} \cdot (\mathbf{r}_x \odot \mathbf{h}_y),$$

where $\mathbb{1}_{[w=w_{x,y}]}$ has value 1 if $w = w_{x,y}$ holds, otherwise it is 0.

4 Experimental Evaluation

In this section, we present thorough empirical studies to evaluate and analyze our proposed SEEK framework. We first introduce the experimental setting. Then we evaluate our SEEK framework on the task of link prediction. Then, we study the influence of the number of segments k to the SEEK framework, and present the case studies to demonstrate why our SEEK framework has high effectiveness.

4.1 Experimental Setting

Datasets In our experiments, we firstly use a *de facto* benchmark dataset: FB15K. FB15K is a subset of the Freebase dataset (Bollacker et al., 2008), and we used the same training, validation and test set provided by (Bordes et al., 2013). We also use another two new datasets proposed in recent years: DB100K (Ding et al., 2018) and YAGO37 (Guo et al., 2018). DB100K was built from the mapping-based objects of core DBpedia (Bizer et al., 2009); YAGO37 was extracted from the core facts of YAGO3 (Mahdisoltani et al., 2013). Table 2 lists the statistics of the three datasets.

Dataset	#Ent	#Rel	#Train	#Valid	#Test
FB15K	14,951	1,345	483,142	50,000	59,071
DB100K	99,604	470	597,572	50,000	50,000
YAGO37	123,189	37	989,132	50,000	50,000

Table 2: Statistics of datasets.

Compared Methods There are many knowledge graph embedding methods developed in recent years. We categorize the compared methods as the following groups:

- Some simple knowledge graph embedding methods that have low time and space complexity, like TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), HolE (Nickel et al., 2016), ComplEx (Trouillon et al., 2016), and Analogy (Liu et al., 2017). Specifically, TransE is a translation based method, and others are the multi-linear dot product-based framework.

- Some methods that achieve state-of-the-art performance on DB100K and YAGO37, which include RUGE (Guo et al., 2018) and ComplEx-NNE+AER (Ding et al., 2018).

- Some latest methods that achieve current state-of-the-art performance on FB15K, including Single DistMult (Kadlec et al., 2017), ConvE (Dettmers et al., 2018), SimplE (Kazemi and Poole, 2018), RotatE (Sun et al., 2019), and DihEdral (Xu and Li, 2019).

- We evaluate the scoring function f_2 to apply an ablation study for our approach. Then we can observe the respective effect of facilitating sufficient feature interactions and preserving the relation properties. Since the scoring function f_2 can only preserve the symmetric property, we refer to it as Sym-SEEK.

Since our framework does not use additional information like text (Toutanova and Chen, 2015), relational path (Ebisu and Ichise, 2019), or external memory (Shen et al., 2017), we do not compare the methods with additional information. Moreover, we only compare our method with single models, and the Ensemble DistMult (Kadlec et al., 2017) is a simple ensemble of multiple different methods, so we do not compare with it.

Experimental Details We use the asynchronous stochastic gradient descent (SGD) with the learning rate adapted by AdaGrad (Duchi et al., 2011)

Methods	DB100K				YAGO37			
	MRR	Hits@N			MRR	Hits@N		
		1	3	10		1	3	10
TransE (Bordes et al., 2013)	0.111	1.6	16.4	27.0	0.303	21.8	33.6	47.5
DistMult (Yang et al., 2015)	0.233	11.5	30.1	44.8	0.365	26.2	41.1	57.5
HolE (Nickel et al., 2016)	0.260	18.2	30.9	41.1	0.380	28.8	42.0	55.1
ComplEx (Trouillon et al., 2016)	0.242	12.6	31.2	44.0	0.417	32.0	47.1	60.3
Analogy (Liu et al., 2017)	0.252	14.2	32.3	42.7	0.387	30.2	42.6	55.6
RUGE (Guo et al., 2018)	0.246	12.9	32.5	43.3	0.431	34.0	48.2	60.3
ComplEx-NNE+AER (Ding et al., 2018)	0.306	24.4	33.4	41.8	–	–	–	–
Sym-SEEK*	0.306	22.5	34.3	46.2	0.452	36.7	**49.8**	60.6
SEEK*	**0.338**	**26.8**	**37.0**	**46.7**	**0.454**	**37.0**	**49.8**	**62.2**

* Statistically significant improvements by independent t-test with $p = 0.01$.

Table 3: Results of link prediction on DB100K and YAGO37.

to optimize our framework. The loss function of our SEEK framework is given by Equation 5. We conducted a grid search to find hypeparameters which maximize the results on validation set, by tuning number of segments $k \in \{1, 2, 4, 8, 16, 20\}$, the dimension of embeddings $D \in \{100, 200, 300, 400\}$, L_2 regularization parameter $\lambda \in \{0.1, 0.01, 0.001, 0.0001\}$ and the number of negative samples per true triple $\eta \in \{10, 50, 100, 500, 1000\}$. The optimal combinations of hyperparameters are $k = 8$, $D = 400$, $\lambda = 0.001$, $\eta = 1000$ on FB15K; $k = 4$, $D = 400$, $\lambda = 0.01$, $\eta = 100$ on DB100K; and $k = 4$, $D = 400$, $\lambda = 0.001$, $\eta = 200$ on YAGO37. We set the initial learning rate lr to 0.1 and the number of epochs to 100 for all datasets.

4.2 Link Prediction

We study the performance of our method on the task of link prediction, which is a prevalent task to evaluate the performance of knowledge graph embeddings. We used the same data preparation process as (Bordes et al., 2013). Specifically, we replace the head/tail entity of a true triple in the test set with other entities in the dataset and name these derived triples as *corrupted triples*. The goal of the link prediction task is to score the original true triples higher than the corrupted ones. We rank the triples by the results of the scoring function.

We use the MRR and Hit@N metrics to evaluate the ranking results: a) MRR: the mean reciprocal rank of original triples; b) Hits@N: the percentage rate of original triples ranked at the top n in prediction. For both metrics, we remove some of the corrupted triples that exist in datasets from the ranking results, which is also called *filtered* setting

Methods	FB15K			
	MRR	Hits@N		
		1	3	10
TransE	0.380	23.1	47.2	47.1
DistMult	0.654	54.6	73.3	72.8
HolE	0.524	40.2	61.3	73.9
ComplEx	0.692	59.9	75.9	84.0
Analogy	0.725	64.6	78.5	85.4
RUGE	0.768	70.3	81.5	86.5
ComplEx-NNE+AER	0.803	76.1	83.1	87.4
Single DistMult	0.798	–	–	**89.3**
ConvE	0.745	67.0	80.1	87.3
SimplE	0.727	66.0	77.3	83.8
RotatE	0.797	74.6	83.0	88.4
DihEdral	0.733	64.1	80.3	87.7
Sym-SEEK*	0.796	74.7	82.9	88.2
SEEK*	**0.825**	**79.2**	**84.1**	88.6

* Statistically significant improvements by independent t-test with $p = 0.01$.

Table 4: Results of link prediction on FB15K.

in (Bordes et al., 2013). We use Hits@1, Hits@3, and Hits@10 for the metrics of Hits@N.

Table 3 summarizes the results of link prediction on DB100K and YAGO37, and Table 4 shows the results on FB15K. Note, the results of compared methods on DB100K and YAGO37 are taken from (Ding et al., 2018; Guo et al., 2018); the results on FB15K are taken from (Kadlec et al., 2017; Ding et al., 2018; Kazemi and Poole, 2018; Sun et al., 2019; Xu and Li, 2019).

On the DB100K, SEEK outperforms the compared methods in all metrics, and the Sym-SEEK also can achieve a good performance. On the YAGO37, the SEEK and Sym-SEEK have a similar result and outperform other previous methods. The results on YAGO37 show that exploiting more

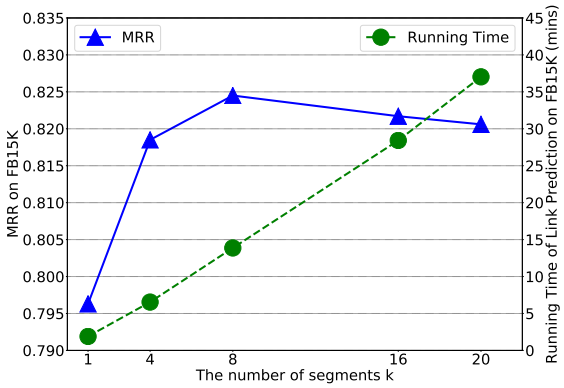

Figure 3: The influence of the number of segments k to the MRR and the running time of link prediction on FB15K.

YAGO37上，交互比关系属性更重要

feature interactions can significantly improve the performance of the embeddings on YAGO37 while preserving the semantic properties have a slight improvement. On FB15K, SEEK achieves the best performance on MRR, Hit@1 and Hit@3. Although SEEK is worse than the Single DistMult on the metrics of Hit@10, the Single DistMult is just a higher dimensional version of DistMult. The Single DistMult uses 512-dimensional embeddings, which is larger than the 400-dimensional embeddings of the SEEK framework. On the whole, our method's improvements on these datasets demonstrate that our method has a higher expressiveness.

4.3 Influence of the Number of Segments k

In the SEEK framework, a larger number of segments k implies more feature interactions and higher computational cost. To empirically study the influence of the number of segments k to the performance and computation time of SEEK, we let k vary in $\{1, 4, 8, 16, 20\}$ and fix all the other hyperparameters, then we observe the MRR and time costs for the link prediction task on the test set of FB15K.

Figure 3 shows the MRR and time costs of different segment counts k on FB15K. As we can see, changing k affects the performance of knowledge graph embeddings significantly. When k varies from 1 to 8, the performance is increased steadily. However, when k becomes even larger, no consistent and dramatic improvements observed on the FB15K dataset. This phenomenon suggests that excessive feature interactions cannot further improve performance. Therefore, k is a sensitive hyperparameter that needs to be tuned for the best performance given a dataset. Figure 3 also illustrates that the running time of SEEK is linear in k, and it verifies that the time complexity of SEEK is $O(kd)$.

交互太多也不好

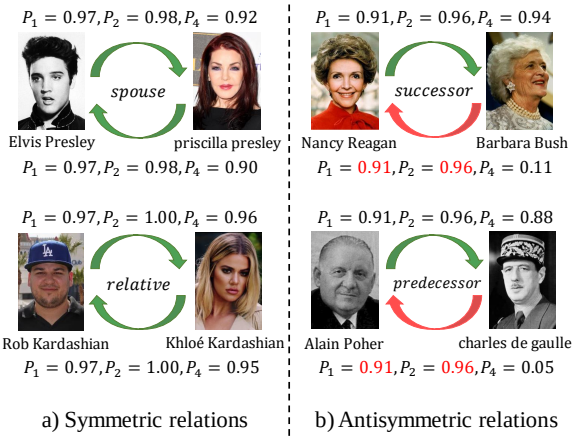

Figure 4: The correct probabilities of four triples in DB100K and their reverse triples. The probabilities P_1, P_2 and P_4 are corresponding to the scoring functions f_1, f_2 and f_4, respectively.

4.4 Case Studies

We employ case studies to explain why our framework has a high expressiveness. Specifically, we utilize the scoring functions f_1, f_2 and f_4 to train the embeddings of DB100K, respectively. Then we use the corresponding scoring functions to score the triples in the test set and their reverse triples, and we feed the scores to the sigmoid function to get the correct probabilities P_1, P_2 and P_4 of each triple. Figure 4 shows the correct probabilities of some triples. In these triples, two triples have symmetric relations, and the other two have antisymmetric relations. On the triples with symmetric relations, the original triples in the test set and their reverse triples are true triples, and the scoring functions f_1, f_2, f_4 can result in high probabilities on original and reverse triples. On the triples with antisymmetric relations, the reverse triples are false. Since the values of $f_1(h, r, t)$ or $f_2(h, r, t)$ are equal to $f_1(t, r, h)$ or $f_2(t, r, h)$, the scoring functions f_1 and f_2 result in high probabilities on the reverse triples. But the scoring function f_4, which can model both symmetric and antisymmetric relations, results in low probabilities on the reverse triples. Meanwhile, we can also find that function f_2 have higher probabilities than function f_1 on the true triples. This phenomenon further explains that facilitating sufficient feature interactions can improve the expressiveness of embeddings.

5 Conclusion and Future Work

In this paper, we propose a lightweight KGE framework (SEEK) that can improve the expressiveness of embeddings without increasing the model complexity. To this end, our framework focuses on designing scoring functions and highlights two critical characteristics: 1) facilitating sufficient feature interactions and 2) preserving various relation properties. Besides, as a general framework, SEEK can incorporate many existing models, such as DistMult, ComplEx, and HolE, as special cases. Our extensive experiments on widely used public benchmarks demonstrate the efficiency, the effectiveness, and the robustness of SEEK. In the future, we plan to extend the key insights of segmenting features and facilitating interactions to other representation learning problems.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (U1711262, U1611264, U1711261, U1811261, U1811264, U1911203), National Key R&D Program of China (2018YFB1004404), Guangdong Basic and Applied Basic Research Foundation (2019B1515130001), Key R&D Program of Guangdong Province (2018B010107005).

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165.

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 SIGMOD*, pages 1247–1250.

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

Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of AAAI*, pages 1811–1818.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. In *Proceedings of ACL*, pages 110–121.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Takuma Ebisu and Ryutaro Ichise. 2019. Graph pattern entity ranking model for knowledge graph completion. *arXiv preprint arXiv:1904.02856*.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of AAAI*, pages 4816–4823.

Katsuhiko Hayashi and Masashi Shimbo. 2017. On the equivalence of holographic and complex embeddings for link prediction. *arXiv preprint arXiv:1702.05563*.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of WSDM*, pages 105–113.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL-IJCNLP*, pages 687–696.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744*.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Proceedings of NIPS*, pages 4289–4300.

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

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *Proceedings of ICML*, pages 2168–2178.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *Proceedings of CIDR*.

Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI*, pages 1955–1961.

Yelong Shen, Po-Sen Huang, Ming-Wei Chang, and Jianfeng Gao. 2017. Modeling large-scale structured relationships with shared memory for knowledge base completion. In *Proceedings of the 2nd Workshop on RepL4NLP*, pages 57–68.

Baoxu Shi and Tim Weninger. 2017. Proje: Embedding projection for knowledge graph completion. In *Proceedings of AAAI*, pages 1236–1242.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of ICLR*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on CVSC*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of ICML*, pages 2071–2080.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2019. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. *arXiv preprint arXiv:1911.00219*.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of WWW*, pages 1835–1844.

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

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of ACL*, pages 950–962.

Canran Xu and Ruijiang Li. 2019. Relation embedding with dihedral group in knowledge graph. In *Proceedings of ACL*, pages 263–272.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of SIGKDD*, pages 353–362.