

# A Semantic Filter Based on Relations for Knowledge Graph Completion

Zongwei Liang, Junan Yang, Hui Liu, Keju Huang  
National University of Defense Technology, Hefei, China  
zwliang17@nudt.edu.cn, yangjunan@ustc.edu.cn,  
{christ592604, huangkeju}@163.com

## Abstract

Knowledge graph embedding, representing entities and relations in the knowledge graphs with high-dimensional vectors, has made significant progress in link prediction. More researchers have explored the representational capabilities of models in recent years. That is, they investigate better representational models to fit symmetry/antisymmetry and combination relationships. The current embedding models are more inclined to utilize the identical vector for the same entity in various triples to measure the matching performance. The observation that measuring the rationality of specific triples means comparing the matching degree of the specific attributes associated with the relations is well-known. Inspired by this fact, this paper designs **Semantic Filter Based on Relations(SFBR)** to extract the required attributes of the entities. Then the rationality of triples is compared under these extracted attributes through the traditional embedding models. The semantic filter module can be added to most geometric and tensor decomposition models with minimal additional memory. Experiments on the benchmark datasets show that the semantic filter based on relations can suppress the impact of other attribute dimensions and improve link prediction performance. The tensor decomposition models with SFBR have achieved state-of-the-art.

## 1 Introduction

Knowledge Graphs (KGs) are collections of large-scale triples, such as Freebase(Bordes et al., 2013), YAGO (Suchanek et al., 2008) and DBpedia(Auer et al., 2007). KGs play a crucial role in applications such as question answering services, search engines, and medical care. Although there are billions of triples in KGs, they are still incomplete. These incomplete knowledge bases will bring limitations to practical applications. Therefore, knowledge graph completion, known as link prediction, which automatically predicts missing links between

entities based on given links, has recently attracted growing attention.

Inspired by word embedding (Mikolov et al., 2013), researchers try to solve the link prediction through knowledge graph embedding. Knowledge graph embedding models map entities and relations into low-dimensional vectors (or matrices, tensors), measure the rationality of triples through specific functions between entities and relations, and rank the triples with function scores. Since TransE(Bordes et al., 2013) proposes to use relation vectors to represent the geometric distance between entities, many variants emerge. For example, TransH(Wang et al., 2014) first explores the different representations of entities under different relations. TransR(Lin et al., 2015) attempts to map entities to the relational space through a particular matrix. TransD(Ji et al., 2015) tries to incorporate the different representations of the entities under the entity and relation into the calculation. These variants attempt to perform complex transformations based on relations or triples to achieve different representations of entities in different semantic spaces.

Recently, scholars are more inclined to solve link prediction by designing models with more powerful representation, such as ComplEx(Trouillon et al., 2016), Tucker(Balazevic et al., 2019), RotatE(Sun et al., 2019), a method based on vector space rotation, and HAKE(Zhang et al., 2020a). Contrary to the actual semantic description, models in recent research apply identical representation for the same entity in different triples.

Since the invention of TransE(Bordes et al., 2013), early scholars, who realized that we should compare different attributes of entities in different triples, tried to improve the model in this direction. However, most recent studies only focus on investigating the more robust representation of entities, such as AutoETER(Niu et al., 2020) and RotatE(Sun et al., 2019). Surprisingly, the attempt to

find various representations of entities in different semantic spaces is gradually discarded.

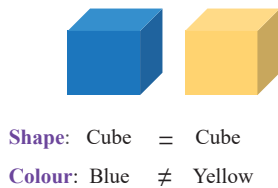


Figure 1: Comparison of boxes with the same shape and different colors.

In practice, entities are collections of attributes, and each entity can contain various semantic attributes. Figure 1 shows the comparison of boxes with the same shape and different colors. When comparing different attributes such as colors or shapes, entities should have different expressions rather than exact representations. The paper believes that each relation describes the links between the head and tail entities in particular attributes. Measuring the plausibility of a given triplet means comparing the matching degree of the attributes associated with the predicate between the entities. Therefore, this paper proposes a semantic filter module to select different attributes of entities in different triples.

This paper designs a semantic filter based on relations. By employing the semantic filter, only the semantics associated with the relations are extracted, and the information of other unneeded dimensions is suppressed. As a result, the head and tail entities are compared under a limited semantic space.

We take the MLP-based semantic filter as the departure. Following the regularization strategy of diagonalization, this paper designs two SFBRs: **Linear-2** and **Diag**. Note that MLP-based SFBR is a general model that can be transformed into most geometric and tensor decomposition models through special regularizations. We analyze several models in Appendix A to show the generality of MLP-based SFBR.

Overall, this paper proposes Semantic Filter Based on Relations (SFBR), which can be added to geometric and tensor decomposition models. SFBR suppresses the interference of useless dimensions and improves the reasoning performance; SFBR occupies minimal additional resources. Experiments on the benchmark datasets show that the tensor decomposition models with SFBR achieve state-of-the-art.

## 2 Related work

In this section, we describe related works and the critical differences between them. We divide knowledge graph embedding models into three leading families (Akrami et al., 2020), including Tensor Decomposition Models, Geometric Models, and Deep Learning Models.

**Tensor Decomposition Models.** These models implicitly consider triples as tensor decomposition. DistMult (Yang et al., 2015) constrains all relation embeddings to be diagonal matrices, which reduces the space of parameters to access a more accessible model to train. RESCAL (Nickel et al., 2011) represents each relationship with a full rank matrix. ComplEx (Trouillon et al., 2016) extends the KG embeddings to the complex space to better model asymmetric and inverse relations. Analogy (Liu et al., 2017) employs the general bilinear scoring function but adds two main constraints inspired by analogical structures. Based on the Tucker decomposition, TuckER (Balazevic et al., 2019) factorizes a tensor into a set of vectors and a smaller shared core matrix.

**Geometric Models.** Geometric Models interpret relations as geometric transformations in the latent space. TransE (Bordes et al., 2013) is the first translation-based method, which treats relations as translation operations from the head entities to the tail entities. Along with TransE (Bordes et al., 2013), multiple variants, including TransH (Wang et al., 2014), TransR (Lin et al., 2015) and TransD (Ji et al., 2015), are proposed to improve the embedding performance of KGs. Recently, RotatE (Sun et al., 2019) defines each relation as a rotation from head entities to tail entities.

**Deep Learning Models.** Deep Learning Models use deep neural networks to perform knowledge graph completion. ConvE (Dettmers et al., 2018) and ConvKB (Nguyen et al., 2018) employ convolutional neural networks to define score functions. CapsE (Nguyen et al., 2019) embeds entities and relations into one-dimensional vectors under the basic assumption that different embeddings encode homologous aspects in the same positions. CompGCN (Vashishth et al., 2020) utilizes graph convolutional networks to update the knowledge graph embedding.

There are also other models, such as DURA (Zhang et al., 2020b), which are proposed to solve overfitting. Together, most of the above studies intend to find a more robust repre-

senting approach. Measuring the effectiveness of certain triples is to compare the matching degree of specific attributes based on relations. Only a few models, such as TransH(Wang et al., 2014), TransR(Lin et al., 2015), and TransD(Ji et al., 2015), consider that entities in different triples should have different representation. However, these variants require many occupations of resources and are limited to particular models.

### 3 Background

In this section, we introduce KG embedding and KG completion tasks. Next, we briefly introduce several models involved in this paper.

**KG Completion.** Knowledge graphs are collections of factual triples  $K = \{(h, r, t), h, t \in \mathcal{E}, r \in \mathcal{R}\}$ , where  $(h, r, t)$  represents a triple in the knowledge graph,  $h, t, r$  are head, tail entities and relations respectively. Knowledge graph embedding associates the entities  $h, t$  and relations  $r$  with vectors  $\mathbf{h}, \mathbf{t}, \mathbf{r}$ . Then we design an appropriate scoring function  $d_r(\mathbf{h}, \mathbf{t}) : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbf{R}$ , to map the embedding of the triple to a certain score. For a particular question  $(h, r, ?)$ , the task of KG completion is ranking all possible answers and obtain the preference of prediction.

**Geometric Models.** The models treat the relations as the transformation of entities in latent spaces. TransE (Bordes et al., 2013) is the first model that uses vectors to represent entities and relations. TransE supposes that entities and relations satisfy  $\mathbf{h} + \mathbf{r} = \mathbf{t}$  where  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbf{R}^n$ . The scoring function can be expressed as:

$$d_r(\mathbf{h}, \mathbf{t}) = - \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \quad (1)$$

RotatE(Sun et al., 2019) defines the relation as a rotation from head entities to tail entities in complex spaces. Given a triple  $\{\mathbf{h}, \mathbf{t}, \mathbf{r}\}$ , we expect that  $\mathbf{t} = \mathbf{h} \circ \mathbf{r}$ , where  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbf{C}^k$  are the embeddings, the modulus for each dimension of relations satisfy  $|\mathbf{r}_i| = 1$  and  $\circ$  denotes the Hadamard product. The score function is :

$$d_r(\mathbf{h}, \mathbf{t}) = - \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|_2 \quad (2)$$

Where  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbf{C}^k, |\mathbf{r}_i| = 1$ .

**Tensor Factorization Models.** Models in this family interpret link prediction as a task of tensor decomposition, where triples are decomposed into a combination (e.g., a multi-linear product) of

low-dimensional vectors for entities and relations. CP(Lacroix et al., 2018) represents triples with canonical decomposition. Note that the same entity has different representations at the head and tail of the triplet. The score function can be expressed as :

$$d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}^T \mathbf{r} \mathbf{t}\| \quad (3)$$

Where  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbf{R}^k$ . RESCAL(Nickel et al., 2011) represents a relation as a matrix  $\mathbf{M}_r \in \mathbf{R}^{d \times d}$  that describes the interactions between latent representations of entities. The score function is defined as:

$$d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h}^T \mathbf{M}_r \mathbf{t}\| \quad (4)$$

ComplEx(Trouillon et al., 2016) extends the real space to complex spaces and constrains the embeddings for relation to be diagonal matrixs. The bilinear product becomes a Hermitian product in complex spaces. The score function can be expressed as:

$$d_r(\mathbf{h}, \mathbf{t}) = \text{Re}(\mathbf{h}^T \text{diag}(\mathbf{r}) \mathbf{t}) \quad (5)$$

where  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbf{C}^k$ .

### 4 SFBR model

This section introduces a novel module—A **Semantic Filter Based on Relations** for knowledge graph completion. We first introduce the basic framework of SFBR in Section 4.1 and the specific filter design in Section 4.2. Finally, we introduce several cases on several models in Section 4.3.

#### 4.1 Framework of SFBR

As is shown in the left of Figure 2, the mainstream KG embedding model depends on the unique representation of entities and relations. The rationality of possible triples is compared through the rankings calculated by the score function.

It is widely accepted that an entity may contain various attributes. This paper believes that each relation describes the relationship between entities in specific attributes. In different triples with different relations, the attributes compared by the triples should also be unique. The comparison requires the choice of needed attributes. For a given triplet, this paper filters out the needed attributes of the triplet by special functions and ranks the triples with the scores calculated by filtered attributes.

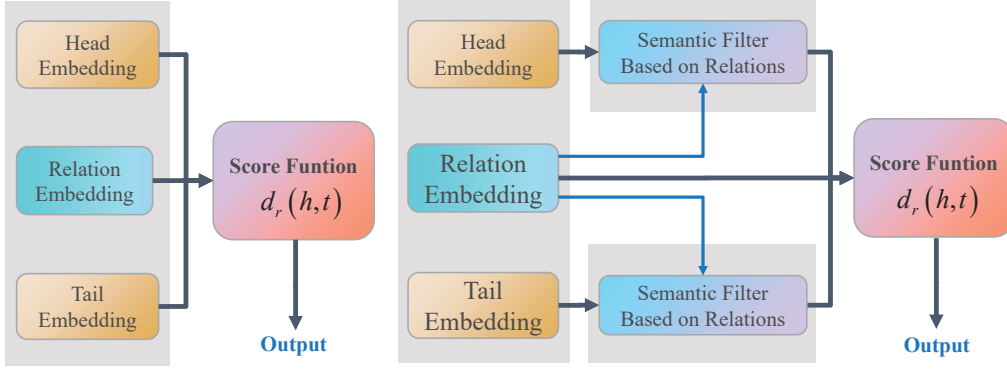


Figure 2: The framework of traditional embeddings models(left) and the framework of embedding models with SFBR(right).

As shown in the right of Figure 2, based on the traditional embedding method, this paper designs a relation-based function for the entities. This function reinforces the dimensions associated with the relations and suppresses the information of other unrelated dimensions. This operation is similar to filters used in signal processing, so the module is named the relation-based semantic filter. The score function can be express as :

$$d_r(\mathbf{h}, \mathbf{t}) = d_r(\mathbf{h}, \mathbf{t}) \quad (6)$$

$$\Rightarrow d_r^f(\mathbf{h}, \mathbf{t}) = d_r(f_r^h(\mathbf{h}), f_r^t(\mathbf{t})) \quad (7)$$

Where  $d_r(\mathbf{h}, \mathbf{t})$  is the traditional scoring function,  $d_r^f(\mathbf{h}, \mathbf{t})$  is the modified scoring function and  $f_r(*)$  is the semantic filter.

## 4.2 Semantic Filter Module

We first try to design the filter based on multilayer perceptron (MLP) for SFBR.

$$\begin{aligned} f_r(\mathbf{h}) &= MLP(\mathbf{h}) \\ &= \mathbf{h} \times \mathbf{W}_r + \mathbf{b} \end{aligned} \quad (8)$$

In order to guarantee each relation filters out different semantics, each relation uses a separate  $f_r(*)$ . However, the semantic filters based on the MLP will bring enormous parameters, and the matrix multiplication requires many resources. As shown in Figure 3, The paper attempts to regularize MLP: diagonalization.

Notice that SFBR is introduced as a module into the existing models in this paper. However, we must be clear about the theoretical status of the SFBR based on MLP. MLP-based SFBR is a general model that can be transformed into most geometric and tensor decomposition models through different regularizations. This paper selects TransE,

RotatE, RESCAL, and ComplEx as examples and conducts these regularization analyses in Appendix A.

$$\mathbf{W}_r = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 \\ \mathbf{W}_3 & \mathbf{W}_4 \end{bmatrix} \quad (9)$$

where  $\mathbf{W}_r \in \mathbf{R}^{n \times n}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4 \in \mathbf{R}^{n/2 \times n/2}$ .

$$\mathbf{W}_r^{Linear-2} = \begin{bmatrix} diag(\mathbf{w}_1) & diag(\mathbf{w}_2) \\ diag(\mathbf{w}_3) & diag(\mathbf{w}_4) \end{bmatrix} \quad (10)$$

where  $\mathbf{W}_r^{Linear-2} \in \mathbf{R}^{n \times n}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4 \in \mathbf{R}^{n/2}$ .

Too many parameters of MLP make the model hard to train, which promotes regularization. First, we ignore the bias. As shown in Eq.(9)and Eq.(10), We decompose the semantic filter matrix of MLP into four square matrices of equal size and diagonalize four square matrices to reduce the parameter quantity of the relational filter. Since this diagonalization equals a linear combination of two parts of entities, we call this SFBR **Linear-2**.

$$\mathbf{W}_r^{Diag} = \begin{bmatrix} diag(\mathbf{w}_1) & \mathbf{O} \\ \mathbf{O} & diag(\mathbf{w}_4) \end{bmatrix} \quad (11)$$

where  $\mathbf{W}_r^{Diag} \in \mathbf{R}^{n \times n}$  and all elements in  $\mathbf{O}$  equal zero.

To further lessen the number of parameters, the paper directly diagonalizes the filter matrix, taking a one-dimensional vector as the semantic filter. The paper names this SFBR **Diag**.

$$f_r(\mathbf{h}) = \mathbf{h} \odot \mathbf{w} + \mathbf{b} \quad (12)$$

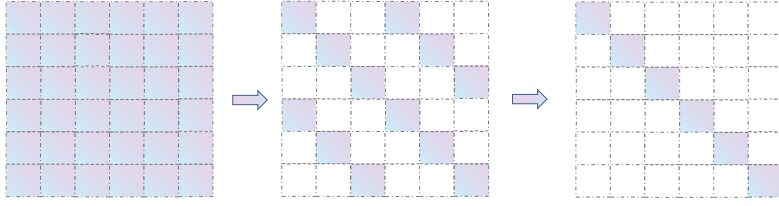


Figure 3: The design route of matrix regularization for SFBR.

Where  $\mathbf{w}, \mathbf{b} \in \mathbf{R}^n$ ,  $\odot$  denotes Hadamard (or element-wise) product,  $\times$  denotes matrix multiplication.

### 4.3 Special Cases with SFBR

This section will introduce the examples of SFBR for different models, including TransE, RotatE, and RESCAL.

The corresponding score function of SFBR based on TransE can be expressed as:

$$d_r^f(\mathbf{h}, \mathbf{t}) = \left\| f_r^h(\mathbf{h}) + \mathbf{r} - f_r^t(\mathbf{t}) \right\| \quad (13)$$

where  $f_r(\mathbf{e}) = \mathbf{e} \times \mathbf{W}_r + \mathbf{b}$ ,  $\mathbf{e}, \mathbf{W}_r, \mathbf{b} \in \mathbf{R}^n$  and  $\mathbf{e}$  represents the entity vectors  $\mathbf{h}, \mathbf{t}$ .

The corresponding score function of SFBR based on RotatE can be expressed as:

$$d_r^f(\mathbf{h}, \mathbf{t}) = \left\| f_r^h(\mathbf{h}) \circ \mathbf{r} - f_r^t(\mathbf{t}) \right\| \quad (14)$$

where  $f_r(\mathbf{e}) = \mathbf{e} \times \mathbf{W}_r + \mathbf{b}$ ,  $\mathbf{e}, \mathbf{W}_r, \mathbf{b} \in \mathbf{C}^n$  and  $\mathbf{e}$  represents the entity vector  $\mathbf{h}, \mathbf{t}$ .

The corresponding score function of SFBR based on RESCAL can be expressed as:

$$d_r^f(\mathbf{h}, \mathbf{t}) = \left\| f_r^h(\mathbf{h})^T \mathbf{M}_r f_r^t(\mathbf{t}) \right\| \quad (15)$$

where  $f_r^h(\mathbf{h}) = \mathbf{h} \times \mathbf{W}_r + \mathbf{b}$ ,  $f_r^t(\mathbf{t}) = \mathbf{t} + \mathbf{b}$  and  $\mathbf{h}, \mathbf{t}, \mathbf{W}_r, \mathbf{b} \in \mathbf{R}^n$ .

Notice that using  $f_r^h(\mathbf{t}) = \mathbf{t} \times \mathbf{W}_r + \mathbf{b}$  for tails is more in line with our design. However, the prediction is to rank the scores of all entities. There are hundreds of thousands of entities. If we apply Hadamard operation for all tails, it will take up enormous resources. The paper simplified SFBR for tails. This simplification can effectively reduce resource occupation. Although the performance is sacrificed, there is still a certain improvement to basic models.

## 5 Experiment

This section is organized as follows. First, we introduce the experimental settings in Section 5.1.

Then, we show the effectiveness of SFBR on three benchmark datasets in Section 5.2. Finally, we visualize and analyze the embeddings generated by SFBR in Section 5.3.

### 5.1 Experimental Settings

**Dataset.** In order to evaluate the proposed module, we consider three common knowledge graph datasets—WN18RR (Toutanova and Chen, 2015), FB15k-237 (Dettmers et al., 2018) and YAGO3-10 (Mahdisoltani et al., 2015). Details of these datasets are listed in Table 1.

FB15k-237 is obtained by eliminating the inverse and equal relations in FB15K, making it more difficult for simple models to do well. WN18RR is achieved by excluding inverse and equal relations in WN18. The main relation patterns are symmetry/antisymmetry and composition. YAGO3-10 is a subset of YAGO3, which is produced to alleviate the test set leakage problem.

**Evaluation Settings.** We use evaluation metrics standard across the link prediction literature: mean reciprocal rank (MRR) and Hits@k,  $k=1,3,10$ . Mean reciprocal rank is the average of the inverse of the mean rank assigned to the true triple over all candidate triples. Hits@k measures the percentage of times a true triple is ranked within the top k candidate triples. We evaluate the performance of link prediction in the filtered setting (Bordes et al., 2013), i.e., all known true triples are removed from the candidate set except for the current test triple. In both settings, higher MRR or higher Hits@1/3/10 indicate better performance.

**Baselines and Training Protocol.** In this section, we compare the performance of SFBR against two categories of KGC models: (1) Geometric Models including TransE(Bordes et al., 2013), RotatE(Sun et al., 2019), TucKer(Balazevic et al., 2019), AutoERTR (Niu et al., 2020) and HAKE(Zhang et al., 2020a), (2) models based on tensor decomposition including CP(Lacroix et al., 2018), RESCAL(Nickel et al., 2011), Com-

Dataset	#entity	#relation	#training	#validation	#test
WN18RR	40943	11	141442	5000	5000
FB15K-237	14505	237	272115	17535	20466
YAGO3-10	123182	37	1079040	5000	5000

Table 1: The number of entities, relations and observed triples in each split for four benchmarks.

plEx(Trouillon et al., 2016) and DURA(Zhang et al., 2020b).

Because SFBR is a module based on existing models, the parameters of our experiments are consistent with those in original papers, and no additional hyperparameters are designed. The parameters of TransE-SFBR and RotatE-SFBR are consistent with the hyper-parameters in RotatE(Sun et al., 2019). CP-SFBR, RESCAL-SFBR and ComplEx-SFBR take the same parameters in DURA(Zhang et al., 2020b). For tensor decomposition models, we find that the model’s output is related to the initialization, so SFBR is trained based on initial results.

## 5.2 Main Results

In this section, we compare the results of SFBR and other state-of-the-art models on three benchmark datasets.

Table 2 shows the comparison between two SFBRs and geometric models. Compared with TransE, TransE-SFBR has significant improvements: on WN18RR, Hit@10 increases by 3.8%; on FB15k-237, Hit@10 increases by 7%. Compared with RotatE, RotatE-SFBR also makes significant progress: on WN18RR, Hit@10 increases by 2.2%; on the FB15k-237, Hit@10 increases by 2%.

The matrix multiplication performed by the MLP-based SFBR requires a lot of GPU memory. Limited by GPU resources, we only experiment on WN18RR, and the embedding dim of entities in TransE-SFBR (MLP) is only 100, which is 1/5 of the original. Therefore, the results of MLP-based SFBR cannot be contrasted with the other two SFBRs. Through comparative experiments on the two datasets, we find that the performance of SFBR based on **Linear-2** is slightly better than that of SFBR based on **Diag**. Nevertheless, the extra parameters and the excessive resource occupancy of the **Linear-2** are twice the **Diag**. In terms of resource utilization, we select **Diag**-based SFBR. The paper chooses **Diag**-based SFBR in the subsequent experiments by default.

Table 3 shows the comparison between SFBR and the models based on tensor decomposition. SFBR improves the performance of the model on almost all datasets. On WN18RR, RESCAL-SFBR obtains the best result (the best Hit@10 is achieved by ComplEx-SFBR). On FB237, ComplEx-SFBR obtains the best result, and MRR is increased by 0.13. On YAGO3-10, although the performance of CP-SFBR and RESCAL-SFBR have been improved, they do not exceed ComplEx-DURA.

Overall, compared with the basic model, the performance on link prediction tasks has been improved by SFBR. Experiments on the standard dataset show that SFBR can improve the performance of base models.

## 5.3 Visualization and Analysis

In this part, we analyze the performance of SFBR from three aspects. First, we visualize the embedding through T-SNE; then, we randomly select a pair of samples to analyze the function of SFBR and show the additional resources occupied by SFBR.

**Visualization.** We use T-SNE to visualize tail entity embeddings. Suppose the link prediction task is  $(h, r, ?)$ , where  $h$  and  $r$  are head entities and relations, respectively. We randomly select ten queries in FB15k-237, which have more than 50 answers. Then, we use T-SNE to visualize the embeddings generated by RotatE and RotatE-SFBR. For each question, T-SNE converts the answers into 2-dimensional points and displays them on the graph with the same color. As shown in Figure 4 and 5, it is a visualization of the distribution of answers to 10 questions. SFBR makes the answers to the same question more similar, indicating that SFBR effectively extracts the needed semantics of each entity and suppresses the attributes of other dimensions, which verifies the claim in Section 4.1.

**Case study.** Two pairs of triples are randomly selected from the test set for analysis. Each pair of triples has the same query:  $(h, r, ?)$ . For each query, a correct answer and an incorrect answer are randomly selected. The first pair of triples,

	WN18RR			FB15K-237		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
TransE*	.223	-	.510	.298	-	.475
RotatE*	.476	.428	.571	.338	.241	.533
AutoERTR	-	-	-	.344	.250	.538
HAKE	<b>.497</b>	<b>.452</b>	.582	.346	.250	.542
TransE-SFBR(MLP)	.184	.006	.388	-	-	-
TransE-SFBR(Linear-2)	.263	.110	.495	.354	.258	.545
TransE-SFBR(Diag)	.242	.028	.548	.338	.240	.538
RotatE-SFBR(Linear-2)	.490	.447	.576	<b>.355</b>	<b>.258</b>	<b>.553</b>
RotatE-SFBR(Diag)	.489	.437	<b>.593</b>	.351	.254	.549

Table 2: Evaluation results of geometric models on FB15K-237 and WN18RR.

	WN18RR			FB15K-237			YAGO3-10		
	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10
CP	.438	.414	.485	.333	.247	.508	.567	.494	.698
RESCAL	.455	.419	.493	.353	.264	.528	.566	.490	.701
ComplEx	.460	.428	.522	.346	.256	.525	.573	.500	.703
CP-DURA	.478	.441	.552	.367	.272	.555	.579	.506	.709
RESCAL-DURA	.498	.455	.577	.368	.276	.550	.579	.505	.712
ComplEx-DURA	.491	.449	.571	.371	.276	.560	<b>.584</b>	.511	<b>.713</b>
CP-SFBR	.485	.447	.561	.370	.274	.563	.582	.510	.711
RESCAL-SFBR	<b>.500</b>	<b>.458</b>	.581	.369	.276	.555	.581	.509	.712
ComplEx-SFBR	.498	.454	<b>.584</b>	<b>.374</b>	<b>.277</b>	<b>.567</b>	.584	<b>.512</b>	.712

Table 3: Evaluation results of tensor decomposition models on WN18RR, FB15K-237 and YAGO3-10.

Model	WN18RR		FB15K-237		YAGO3-10	
	original	SFBR	original	SFBR	original	SFBR
TransE	20.48M	20.49M	14.78M	15.25M	-	-
RotatE	40.95M	40.96M	29.32M	29.79M	123.20M	123.24M
CP	163.82M	163.90M	59.11M	61.01M	246.45M	246.60M
RESCAL	11.92M	11.93M	131.70M	132.19M	246.52M	246.82M
ComplEx	163.86M	164.04M	60.06M	60.85M	82.47M	82.55M

Table 4: Comparison of parameter size between SFBR and basic models on different datasets.

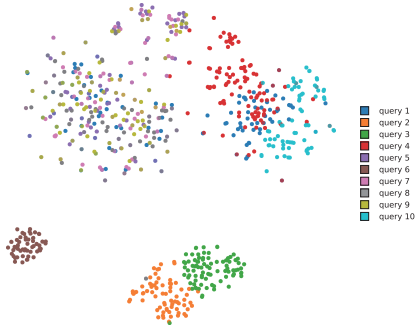


Figure 4: Visualization of tail entities in RotatE using T-SNE. A point represents a tail entity. Points in the same color represent tail entities that have the same context( $h_r, r_j$ ).

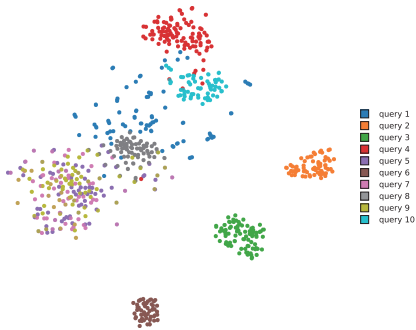


Figure 5: Visualization of tail entities in RotatE-SFBE using T-SNE.

which cannot be predicted in TransE, can be distinguished by TransE-SFBER; for the other pair, both models can effectively predict them. Draw the distance of two triples:  $\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_1$ , where  $\mathbf{h}, \mathbf{r}, \mathbf{t}$  are embeddings of entities and relations. Figure 6 and Figure 7 show the distance; the blue one is the triple deviation of the correct answer, the red one is the deviation of error triple. The top of the figure shows the distance of TransE, and the bottom is the distance of TransE-SFBE. From Figure 6, we can find that for the tail entity that cannot be predicted in TransE, TransE-SFBR suppresses the influence of irrelevant dimensions, and the tail entity can be predicted. For the tail entities, which TransE can predict in Figure 7, SFBR further suppresses the noise of other dimensions, and the distance between the correct and wrong tails is further enlarged, which enhances the model.

**Resource occupation.** As shown in Table 4, the parameters of SFBR and other basic models on the three datasets are compared. The comparison finds that the parameter of SFBR only increases by 0.01 ~ 0.5M on the model based on geometric

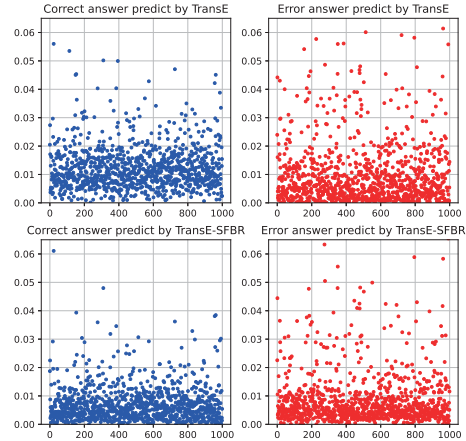


Figure 6: Distance of a pair of triples, which cannot be predicted in TransE(upper) and can be reasoned in TransE-SFBR(lower).

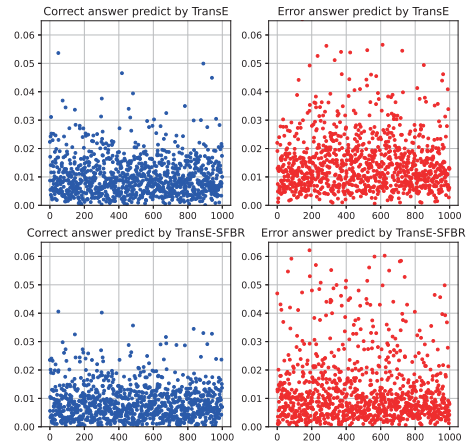


Figure 7: Distance of a pair of triples, which can be predicted both in TransE(upper) and TransE-SFBR(lower)

distance; the parameter of SFBR only increases by 0.01 ~ 1.9M on the model based on tensor decomposition. Especially in the geometric models, there is a small growth of parameters, and the performance can be significantly improved. In all cases, SFBR brings minimal growth in resource occupation to the basic model.

## 6 Conclusion

This paper designs a relation-based semantic filter—SFBR—for the geometric and tensor decomposition models based on knowledge graph completion. SFBR is based on the observation that judging the rationality of a particular triple is to compare specific attributes between the entities, ignoring other unrelated dimensions. Therefore, this paper provides a relation-based semantic filter to extract the attributes that need to be compared and suppress the irrelevant attributes of entities.



Experiments show that SFBR can effectively improve the performance of the traditional models, especially the geometric models. The visualization shows that SFBR can effectively extract the relevant dimensions and distinguish the comparisons among different attributes. Compared with the base models, SFBR only has a slight growth in resource occupation.

## Acknowledgements

This work was partially supported by the Anhui Provincial Natural Science Foundation (NO.1908085MF202) and Independent Scientific Research Program of National University of Defense Science and Technology (NO.ZK18-03-14).

## References

- F. Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and C. Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. *ArXiv*, abs/1901.09590.
- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, J. Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and S. Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.
- Guoliang Ji, Shizhu He, L. Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*.
- Timothée Lacroix, Nicolas Usunier, and G. Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *ICML*.
- Yankai Lin, Zhiyuan Liu, M. Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.
- Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *ICML*.
- F. Mahdisoltani, J. Biega, and Fabian M. Suchanek. 2015. Yago3: A knowledge base from multilingual wikipeidias. In *CIDR*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.
- Dai Quoc Nguyen, T. Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. *ArXiv*, abs/1712.02121.
- Dai Quoc Nguyen, Thanh Vu, T. Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. *ArXiv*, abs/1808.04122.
- Maximilian Nickel, Volker Tresp, and H. Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*.
- Guanglin Niu, Bo Li, Yongfei Zhang, S. Pu, and Jingyang Li. 2020. Autoeter: Automated entity type representation for knowledge graph embedding. *ArXiv*, abs/2009.12030.
- Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, 6(3):203–217.
- Zhiqing Sun, Zhihong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *ArXiv*, abs/1902.10197.
- Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference.
- Théo Trouillon, Johannes Welbl, S. Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.
- Shikhar Vashishth, Soumya Sanyal, V. Nitin, and P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. *ArXiv*, abs/1911.03082.
- Zhen Wang, J. Zhang, Jianlin Feng, and Z. Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.
- B. Yang, Wen tau Yih, X. He, Jianfeng Gao, and L. Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575.
- Zhanqiu Zhang, Jianyu Cai, Yong dong Zhang, and J. Wang. 2020a. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *AAAI*.

Zhanqiu Zhang, Jianyu Cai, and J. Wang. 2020b. Duality-induced regularizer for tensor factorization based knowledge graph completion. *ArXiv*, abs/2011.05816.

## A Analysis of generality for MLP-based SFBR

The MLP-based SFBR model after different regularization will be equivalent to various geometric models or tensor decomposition models. The paper randomly selects several models for analysis. First, we merge the biases of MLP:

$$\begin{aligned} d_r^f(\mathbf{h}, \mathbf{t}) &= \left\| \mathbf{h} \times \mathbf{W}_r^h + \mathbf{b}_r^h - \mathbf{t} \times \mathbf{W}_r^t - \mathbf{b}_r^t \right\|_p \\ &= \left\| \mathbf{h} \mathbf{W}_r^h + \mathbf{r} - \mathbf{t} \mathbf{W}_r^t \right\|_p \end{aligned} \quad (16)$$

where  $\mathbf{r} = \mathbf{b}_r^h - \mathbf{b}_r^t$ .

As shown in Eq.(17), when the two semantic filter matrixs satisfy  $\mathbf{W}_r^h, \mathbf{W}_r^t = \mathbf{I}$ , the semantic filter model is equivalent to TransE(Bordes et al., 2013).

$$\begin{aligned} d_r^f(\mathbf{h}, \mathbf{t}) &= \left\| \mathbf{h} \times \mathbf{I} + \mathbf{r} - \mathbf{t} \times \mathbf{I} \right\|_p \\ &= \left\| \mathbf{h} + \mathbf{r} - \mathbf{t} \right\|_p \end{aligned} \quad (17)$$

When the MLP does not have any bias part, the filter matrix for heads is a special **Linear-2** diagonalized matrix and the entity dimension is expanded to twice the original, the SFBR model is equivalent to RotatE(Sun et al., 2019).

$$\begin{aligned} d_r^f(\mathbf{h}, \mathbf{t}) &= \left\| \mathbf{h} \times \mathbf{W}_r^h + \mathbf{r} - \mathbf{t} \times \mathbf{W}_r^t \right\|_p \\ &= \left\| \mathbf{h} \mathbf{W}_r^h - \mathbf{t} \right\|_p \end{aligned} \quad (18)$$

$$\mathbf{W}_r^h = \begin{bmatrix} \text{diag}(\cos\theta) & \text{diag}(-\sin\theta) \\ \text{diag}(\sin\theta) & \text{diag}(\cos\theta) \end{bmatrix} \quad (19)$$

where  $\mathbf{b}_r^h + \mathbf{b}_r^t = \mathbf{r} = 0$ ,  $\mathbf{W}_r^t = \mathbf{I}$ ,  $\theta \in \mathbf{R}^{n/2}$ ,  $\mathbf{h} = [\mathbf{h}_{\text{rel}}, \mathbf{h}_{\text{img}}]$  and  $\mathbf{t} = [\mathbf{t}_{\text{rel}}, \mathbf{t}_{\text{img}}]$ .

When  $p = 2, \mathbf{r} = 0$  and  $\mathbf{W}_r^t = \mathbf{I}$ , the MLP-based SFBR can be simplified as Eq.(20). RESCAL(Nickel et al., 2011) selects the third term of the formula as the optimization goal. The performance will be better with the other two items used as additional regularization items. RESCAL-DURA(Zhang et al., 2020b) takes the first two items as regularization. Just Similar to RotatE,

ComplEx(Trouillon et al., 2016) achieves performance improvement by expanding the dimension and applying special matrix regularization for RESCAL. DURA(Zhang et al., 2020b) has the corresponding proof for ComplEx-DURA.

$$\begin{aligned} \left( d_r^f(\mathbf{h}, \mathbf{t}) \right)^2 &= \left\| \mathbf{h} \times \mathbf{W}_r^h + \mathbf{r} - \mathbf{t} \times \mathbf{W}_r^t \right\|_p^2 \\ &= \left\| \mathbf{h} \times \mathbf{W}_r^h - \mathbf{t} \right\|_2^2 \\ &= \left\| \mathbf{h} \mathbf{W}_r^h \right\|_2^2 + \|\mathbf{t}\|_2^2 + 2\mathbf{h} \mathbf{W}_r^h \mathbf{t}^T \end{aligned} \quad (20)$$

From the above cases, we can quickly discover that the MLP-based SFBR is a general model. Most geometric models and tensor decomposition models can be equivalent to MLP-based SFBR after particular regularization.