Pretrain-KGEs: Learning Knowledge Representation from Pretrained Models for Knowledge Graph Embeddings

Zhiyuan Zhang[1], **Xiaoqian Liu**[1, 2], **Yi Zhang**[1], **Qi Su**[1, 2], **Xu Sun**[1] and **Bin He**[3]
[1] MOE Key Laboratory of Computational Linguistic, School of EECS, Peking University
[2] School of Foreign Languages, Peking University
[3] Huawei Noah's Ark Lab
{zzy1210,liuxiaoqian,zhangyi16,sukia,xusun}@pku.edu.cn
hebin.nlp@huawei.com

Abstract

Learning knowledge graph embeddings (KGEs) is an efficient approach to knowledge graph completion. Conventional KGEs often suffer from limited knowledge representation, which causes less accuracy especially when training on sparse knowledge graphs. To remedy this, we present *Pretrain-KGEs*, a training framework for learning better knowledgeable entity and relation embeddings, leveraging the abundant linguistic knowledge from pretrained language models. Specifically, we propose a unified approach in which we first learn entity and relation representations via pretrained language models and use the representations to initialize entity and relation embeddings for training KGE models. Our proposed method is model agnostic in the sense that it can be applied to any variant of KGE models. Experimental results show that our method can consistently improve results and achieve state-of-the-art performance using different KGE models such as TransE and QuatE, across four benchmark KG datasets in link prediction and triplet classification tasks.

1 Introduction

Knowledge graphs (KGs) constitute an effective access to world knowledge for a wide variety of NLP tasks, such as question-answering, entity linking and information retrieval. A typical KG such as Freebase (Bollacker et al., 2008) and WordNet (Miller, 1995) consists of a set of triplets in the form of (h, r, t) with the head entity h and the tail entity t as nodes and relations r as edges in the graph. A triplet represents the relation between two entities, e.g., *(Steve Jobs, founded, Apple Inc.)*. Despite their effectiveness, KGs in real applications suffer from incompleteness and there have been several attempts for knowledge graph completion among which knowledge graph embedding is one of prominent approaches.

Knowledge graph embedding (KGE) models have been designed extensively in recent years (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015; Sun et al., 2019; Ebisu and Ichise, 2018; Nickel et al., 2011; Yang et al., 2015; Kazemi and Poole, 2018; Trouillon et al., 2016; Zhang et al., 2019). The general methodology of these models is to model entities and relations in vector spaces based on a score function for triplets (h, r, t). The score function measures the plausibility of each candidate triplet (h, r, t) compared to corrupted false triplets (h', r, t) or (h, r, t'). However, traditional KGE models often suffer from limited knowledge representation due to the simply symbolic representation of entities and relations. Some recent works take advantages of both fact triplets and textual description to enrich knowledge representation (Socher et al., 2013a; Xu et al., 2017; Xiao et al., 2017; Xie et al., 2016; An et al., 2018), but without exploitation of contextual information of the textual descriptions. Moreover, much of this research effort has been dedicated to developing novel architectures for knowledge representation without applications to KGE models.

Unlike many existing works which try to propose new architectures for KGEs or knowledge representation, we focus on model-agnostic pretraining technique for KGE models. We present a unified training framework named as Pretrain-KGEs which consists of three phases: fine-tuning phase, initializing phase and training phase (see Fig. 1). During the fine-tuning phase, we learn better knowledgeable entity and relation representations via pretrained language models using textual descriptions as input sequence. Different from previous works incorporating textual information into knowledge representation, we use pretrained langauge models such as BERT (Devlin et al., 2019) to better understand textual description by making full use of syntactic and semantic information in large-

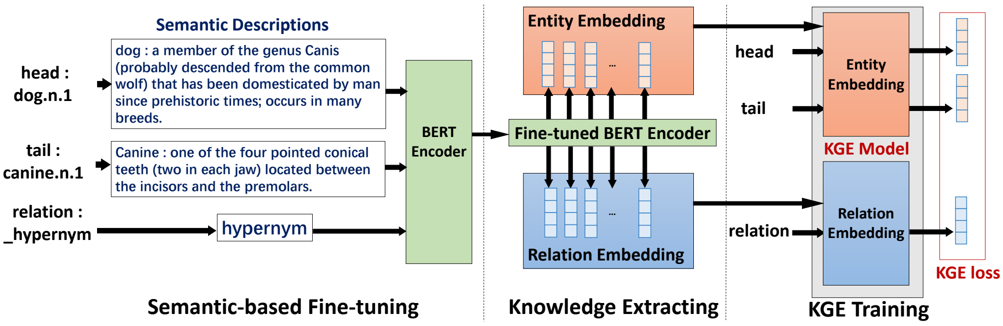

Semantic Descriptions
head : dog.n.1
dog : a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds.
tail : canine.n.1
Canine : one of the four pointed conical teeth (two in each jaw) located between the incisors and the premolars.
relation : _hypernym
hypernym
BERT Encoder
Semantic-based Fine-tuning
Entity Embedding
Fine-tuned BERT Encoder
Relation Embedding
Knowledge Extracting
head
tail
relation
Entity Embedding
KGE Model
Relation Embedding
KGE loss
KGE Training

Figure 1: An illustration of our proposed three-phase Pretrain-KGEs.

scale corpora on which BERT is pretrained. Thus, we enable to incorporate rich linguistic knowledge learned by BERT into entity and relation representations. Then during the initializing phase, we use knowledgeable entity and relation representations to initialize entity and relation embeddings so that the initialized KGEs inherit the rich knowledge. Finally, during the training phase, we train a KGE model the same way as a traditional KGE model to learn entity and relation embeddings.

Extensive experiments using six public KGE models across four benchmark KG datasets show that our proposed training framework can consistently improve results and achieve state-of-the-art performance in link prediction and triplet classification tasks. Our contributions are as follows:

- We propose a model-agnostic training framework for learning knowledge graph embeddings by first learning knowledge representation via pretrained language models.
- Results on several benchmark datasets show that our method can improve results and achieve state-of-the-art performance over variants of knowledge graph embedding models in link prediction and triplet classification tasks.
- Further analysis demonstrates the effects of knowledge incorporation in our method and shows that our Pretrain-KGEs outperforms baselines especially in the case of fewer training triplets, low-frequency and the out-of-knowledge-base (OOKB) entities.

2 Background and Related Work

2.1 Knowledge Graph Embedding

For each head entity h and tail entity t with their corresponding entity embeddings E_h, E_t, and each relation r with its relation embeddings R_r, we formulate KGE models as follows:

$$v_h, v_r, v_t = E_h, R_r, E_t \tag{1}$$

$$\text{score} = f(v_h, v_r, v_t) \tag{2}$$

where $v_h, v_r, v_t \in \mathbb{F}^d$ are the learnt vectors for each head entity, relation, and tail entity respectively, The model is then optimized to calculate a higher score for true triplets than corrupted false ones.

According to the score function, KGE models can be roughly divided into translational models and semantic matching models (Wang et al., 2017). Translational models popularized by TransE (Bordes et al., 2013) learn vector embeddings of the entities and the relations, and consider the relation between the head and tail entity as a translation between the two entity embeddings, i.e., in the form of $v_h + v_r \approx v_t$ when the candidate triplet (h, r, t) holds. Since TransE has problems when dealing with 1-to-N, N-to-1 and N-to-N relations, different translational models are proposed subsequently to define various relational patterns, such as TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015), RotatE (Sun et al., 2019), and TorusE (Ebisu and Ichise, 2018).

On the other hand, semantic matching models define a score function to match latent semantics of the head, tail entity and the relation. For instance, RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015), SimplE (Kazemi and Poole, 2018), and ComplEx (Trouillon et al., 2016) adopt a bilinear approach to model entities and relations for KGEs. Specifically, ComplEx learns complex-valued representations of entities and relations in complex space, while DistMult, SimplE, and RESCAL embed entities and relations in the traditional real number field. The recent state-of-the-art, QuatE (Zhang et al., 2019) represents entities as hypercomplex-valued embeddings and models relations as rotations in the quaternion space.

Both translational models and semantic matching models learn entity and relation embeddings in spite of different embedding spaces. However, these KGE models only use structural information observed in triplets without incorporating external knowledge resources into KGEs, such as textual description of entities and relations. Thus, the embeddings of entities and relations suffer from limited knowledge representation. We instead propose a unified approach to introduce rich linguistic knowledge into KGEs via pretrained language models.

2.2 Text mining for Knowledge Representation

In a knowledge graph dataset, names of each entity and relation are provided as textual description of entities and relations. Socher et al. (2013a) first utilize textual information to represent entities by averaging word embeddings of entity names. Following the word averaging method, Li et al. (2016) improve the coverage of commonsense resources in ConceptNet (Speer and Havasi, 2012) by mining candidate triplets from Wikipedia. They leverage a word averaging model to convert entity and relation names into name vectors. Other recent works also leverage textual description to enrich knowledge representation but ignore contextual information of the textual descriptions (Socher et al., 2013a; Xu et al., 2017; Xiao et al., 2017; Xie et al., 2016; An et al., 2018). Instead, our method exploits rich contextual information via pretrained models.

2.3 Deep Contextualized Word Embeddings

Recent approaches to modeling language representations offer significant improvements over embeddings, especially pretrained deep contextualized lanaguge representation models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and T5 (Raffel et al., 2019). These deep language models learn better contextualized word presentations, since they are pretrained on large-scale free text data, which make full use of syntactic and semantic information in the large corpora. In this work, we use BERT, a bidirectional Transformer encoder to learn entity and relation representation given textual description. Therefore, by incorporating the plentiful linguistic knowledge learned by pretrained language models, our proposed method can learn better knowledgeable entity and relation representations for subsequent KGE learning.

3 Method

In this section, we will introduce our unified training framework Pretrain-KGEs and provide details of learning knowledgeable entity and relation representations via BERT.

3.1 Training Framework

An overview of Pretrain-KGEs is shown in Fig. 1. The framework consists of three phases: fine-tuning phase, initializing phase, and training phase. Our major contribution is the fine-tuning phase with the initializing phase, which incorporates rich knowledge into KGEs via pretained language models, i.e., BERT that enables to exploit contextual information of textual description for entities and relations. By initializing embeddings with knowledgeable entity and relation representations, our training framework improves KGE models to learn better entity and relation embeddings.

Fine-tuning Phase Given textual description of entities and relations such as entity names and relation names, we first encode the textual descriptions into vectors via pretrained language models to represent entities and relations respectively. We then project the entity and relation representations into two separate vector spaces to get the entity encoder $\mathrm{Enc}_e(\cdot)$ for each entity e and the relation encoder $\mathrm{Enc}_r(\cdot)$ for each relation r. Formally, $\mathrm{Enc}_e(\cdot)$ and $\mathrm{Enc}_r(\cdot)$ output entity and relation representations as:

$$v_h, v_r, v_t = \mathrm{Enc}_e(h), \mathrm{Enc}_r(r), \mathrm{Enc}_e(t) \qquad (3)$$

where v_h, v_r, and v_t represents encoding vectors of the head entity, the relation, and the tail entity in a triplet (h, r, t) respectively. For details of $\mathrm{Enc}_e(\cdot)$ and $\mathrm{Enc}_r(\cdot)$, see section 3.2.

Given the entity and relation representations, we then calculate the score of a triplet to measure its plausibility in Eq. 2. For instance, if TransE is adopted, the score function is $\|v_h + v_r - v_t\|$. After fine-tuning, the knowledge representation is used in the following initializing phase.

Initializing Phase Given the knowledgeable entity and relation representation, we initialize entity embeddings E and relation embeddings R for a KGE model instead of random initialization.

Specifically, $E = [E_1; E_2; \cdots; E_k] \in \mathbb{F}^{k \times d}$ and $R = [R_1; R_2; \cdots; R_l] \in \mathbb{F}^{l \times d}$ in which ";" denotes concatenating column vectors into a matrix. k and l denote the total number of entities and

Model	FB15K			FB15K-237			WN18			WN18RR		
	H@10↑	MRR↑	MR↓	H@10↑	MRR↑	MR↓	H@10↑	MRR↑	MR↓	H@10↑	MRR↑	MR↓
TransE	**0.866**	**0.731**	40.3	0.528	0.330	171.6	0.920	**0.773**	265	0.528	0.223	3372
Pretrain-TransE	**0.866**	**0.731**	**36.6**	**0.529**	**0.332**	**162.0**	**0.928**	0.757	**85**	**0.557**	**0.235**	**1747**♠
DistMult	**0.887**	**0.768**	37.5	**0.484**	**0.307**	175.1	**0.931**	**0.686**	282	**0.534**	**0.440**	4886
Pretrain-DistMult	0.883	0.764	**37.0**	0.482	0.306	**171.3**	0.923	0.660	**142**	0.527	0.432	**3550**
ComplEx	**0.887**	**0.771**	47.1	0.511	0.322	166.1	0.925	**0.893**	323	**0.555**	**0.469**	5421
Pretrain-ComplEx	0.879	0.763	**45.2**	**0.513**	**0.323**	**156.9**	**0.949**	0.859	**194**	0.553	0.459	**4468**
RotatE	**0.881**	**0.790**♠	41.7	0.531	0.336	177.0	0.960	**0.949**	269	0.574	**0.474**	3363
Pretrain-RotatE	**0.881**	0.784	**38.4**	**0.534**	**0.337**	**168.3**	**0.962**	0.927	**125**	**0.580**	0.447	**2138**
QuatE	0.898	**0.778**	17.4	0.550	0.349	86.2	0.960	**0.951**♠	180	0.581	0.487	2290
Pretrain-QuatE	**0.899**♠	0.764	**17.2**♠	**0.554**♠	**0.350**♠	**84.4**♠	**0.964**♠	0.944	**72**♠	**0.586**♠	**0.488**♠	**2085**

Table 1: Link prediction results on four KG datasets. The experiments here use entity names and relation names as the semantic description. ↓ means that a lower metric is better. ↑ means that a higher metric is better. ♠ denotes state-of-the-art performance.

relations respectively. $\mathbb{F}$ satisfies $\mathbb{R} \subseteq \mathbb{F}$ and d denotes the embedding dimension. Then $E_i \in \mathbb{F}^d$ represents the embedding of entity with index i and $R_j \in \mathbb{F}^d$ represents the embedding of relation with index j.

During the initializing phase, we use the representation vector of entity with index i encoded by the entity encoder $\text{Enc}_e(\cdot)$ as the initialized embedding E_i for training KGE models to learn entity embeddings. Likewise, the representation vector of relation with index j encoded by the relation encoder $\text{Enc}_r(\cdot)$ is considered as the initialized embedding R_j for training KGE models to learn relation embeddings.

Training Phase After initializing entity and relation embeddings with knowledgeable entity and relation representations, we train a KGE model in the same way as a traditional KGE model. We calculate the score of each training triplet in Eq. 1 and Eq. 2 with the same score function in the fine-tuning phase. Finally, we optimize the entity embedding E and the relation embedding R using the same loss function of the corresponding KGE model. For example, if TransE and the max-margin loss function with negative sampling are adopted, the loss in the training phase is calculated as:

$$\mathcal{L} = \left[\gamma + f(v_h, v_r, v_t) - f(v_{h'}, v_{r'}, v_{t'})\right]_+ \quad (4)$$

where (h, r, t) and (h', r', t') represent a candidate and a corrupted false triplet respectively, γ denotes the margin, $[\ \cdot\ ]_+ = \max(\cdot, 0)$, and $f(\cdot)$ denotes score function of TransE (Bordes et al., 2013).

3.2 Learning knowledgeable Entity and Relation Representations via BERT

To learn better knowledge representation of entities and relations given textual description, we first encode the textual description through Bert (Devlin et al., 2019), a bidirectional Transformer encoder which is pretrained on large-scale corpora and thus learns rich contextual information of texts by making full use of syntactic and semantic information in the large corpora.

We define $T(e)$ and $T(r)$ as the textual description of entities and relations respectively. The textual description can be words, phrases, or sentences providing information about entities and relations such as names of entities and relations or definitions of word senses. For example, the definition of entity $e =$ *Nyala.n.1* in WordNet is *city in Sudan*. Then $T($*Nyala.n.1*$) =$ *Nyala : city in Sudan*.

Given the textual descriptions of entities and relations $T(e)$ and $T(r)$, $\text{Bert}(\cdot)$ converts $T(e)$ and $T(r)$ into entity representation and relation representation respectively in a vector space $\mathbb{R}^n$ (n denotes the vector size). We then project the entity and relation representations into two separate vector spaces $\mathbb{F}^d$ through linear transformations. Formally, we get the entity encoder $\text{Enc}_e(\cdot)$ for each entity e and the relation encoder $\text{Enc}_r(\cdot)$ for each relation r as:

$$\text{Enc}_e(e) = \sigma(W_e \text{Bert}(T(e)) + b_e) \quad (5)$$

$$\text{Enc}_r(r) = \sigma(W_r \text{Bert}(T(r)) + b_r) \quad (6)$$

where $W_e, W_r \in \mathbb{F}^{d \times n}$, $b_e, b_r \in \mathbb{F}^d$, and $\sigma : \mathbb{F}^d \rightarrow \mathbb{F}^d$ denotes a nonlinear function[1].

The entity and relation representation encoded by $\text{Enc}_e(\cdot)$ and $\text{Enc}_r(\cdot)$ are then used to initialize entity and relation embeddings for a KGE model.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our proposed training framework on four benchmark KG datasets: WN18 (Bor-

[1] In our implementation, we adopt a generalized $\tanh(\cdot)$ function defined on $\mathbb{F}^d$. See Appendix.A.

Model	WN18+Name	+Definition	WN18RR+Name	+Definition
TransE	85	**63**(+)	1747	**1228**(+)
DistMult	142	**136**(+)	3550	**3515**(+)
ComplEx	194	**168**(+)	4468	**4448**(+)
RotatE	125	**110**(+)	2138	**1917**(+)
pRotatE	305	**196**(+)	3814	**3016**(+)
QuatE	72	**62**(+)	**2085**	2106(-)

Table 2: MR results of link prediction on WordNet. "Name" means using entity names and relation names as textual description. "Definition" means using names of entities and relations as well as definitions of word senses as textual description.

des et al., 2013), WN18RR (Dettmers et al., 2018), FB15K (Bordes et al., 2013) and FB15K-237 (Toutanova and Chen, 2015).[2] WN18 and WN18RR are two subsets of WordNet; FB15K and FB15K-237 are two subsets of FreeBase. WordNet is a large KG where entities are synsets corresponding to word senses and relations represents lexical relations between entities. Freebase is a large KG of general world facts. We use entity names and relation names provided by the four datasets as input textual descriptions for BERT, and we also utilize synsets definitions provided by WordNet as additional textual descriptions of entities.

In our experiments, we perform link prediction task (filtered setting) mainly with triplet classification task. The link prediction task aims to predict either the head entity h given the relation r and the tail entity t or the tail entity given the head entity and the relation, while triplet classification aims to judge whether a candidate triplet is correct or not.

For the link prediction task, we generate corrupted false triplets (h', r, t) and (h, r, t') using negative sampling. For n test triplets, we get their ranks $\mathbf{r} = (r_1, r_2, \cdots, r_n)$ and calculate standard evaluation metrics: Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits at N (H@N).

For triplet classification, we follow the evaluation protocol in Socher et al. (2013b) and adopt the accuracy metric (Acc) to evaluate our training method.

4.2 Baselines

To evaluate the universality of our training framework Pretrain-KGEs, we select multiple public KGE models as baselines including **translational models**:

- TransE (Bordes et al., 2013), the translational-based model which models the relation as translations between entities;

Dataset	Link prediction					Class.
FB15K	H@10 ↑	H@3 ↑	H@1 ↑	MRR ↑	MR ↓	Acc ↑
QuatE	0.898	**0.832**♠	**0.704**♠	**0.778**♠	17.4	0.927
+Name	**0.899**♠	**0.832**♠	0.677	0.764	**17.2**♠	**0.928**♠
FB15K-237	H@10 ↑	H@3 ↑	H@1 ↑	MRR ↑	MR ↓	Acc ↑
QuatE	0.550	0.383	0.249	0.349	86.2	0.816
+Name	**0.554**♠	**0.384**♠	**0.250**♠	**0.350**♠	**84.8**♠	**0.817**♠
WN18	H@10↑	H@3↑	H@1↑	MRR↑	MR↓	Acc↑
QuatE	0.960	0.954	**0.946**♠	**0.951**♠	180	0.977
+Name	**0.964**♠	**0.954**♠	0.931	0.944	72	**0.981**♠
+Definition	0.963	**0.954**♠	0.930	0.943	**62**♠	0.980
WN18RR	H@10↑	H@3↑	H@1↑	MRR↑	MR↓	Acc↑
QuatE	0.581	0.507	**0.438**♠	0.487	2290	0.866
+Name	**0.586**♠	**0.509**♠	0.437	**0.488**♠	**2085**♠	0.874
+Definition	**0.586**♠	**0.509**♠	0.433	0.487	2106	**0.876**♠

Table 3: Link prediction and triplet classification ("Class.") results over QuatE. ↓ means a lower metric is better. ↑ means a higher metric is better. ♠ denotes state-of-the-art performance of KGE models. "+Name" means Pretrain-KGE uses entity and relation names as semantic description. "+Definition" means Pretrain-KGE also adopts definitions of word senses as additional semantic description.

- RotatE (Sun et al., 2019), the extension of translational-based models which introduces complex-valued embeddings to model the relations as rotations in complex vector space;
- pRotatE (Sun et al., 2019), a variant of RotatE where the modulus of complex entity embeddings are constrained and only phase information is involved;

and **semantic matching models**:

- DistMult (Yang et al., 2015), a semantic matching model where each relation is represented with a diagonal matrix;
- ComplEx (Trouillon et al., 2016), the extension of semantic matching model which embedds entities and relations in complex space.
- QuatE (Zhang et al., 2019), the recent state-of-the-art KGE model which learns entity and relation embeddings in the quaternion space.

4.3 Main Results

We present results for the Pretrain-KGEs algorithm in Table 1, Table 2 and Table 3. Table 1 shows the link prediction results on four benchmark KG datasets using six public KGE models. Table 2 compares the results on WordNet of using entity names and relation names to the results of adding definitions of word senses as additional textual description for entities. Table 3 demonstrates the state-of-the-art performance of our proposed

[2] Detailed statistics of datasets are in Appendix.A.

method in both link prediction and triplet classification tasks[3]. From the results, we can observe that:

(1) Our unified training framework can be applied to multiple variants of KGE models in spite of different embedding spaces, and achieves improvements over TransE, DistMult, ComplEx, RotatE, pRotatE and QuatE on most evaluation metrics, especially on MR but still being competitive on MRR (see detailed analysis of MR and MRR in section 5.2.1). Yet, it verifies the universality of our training framework. The reason is that our method incorporates rich linguistic knowledge into entity and relation representation via pretrained language models to learn better knowledgeable representation for the embedding initialization in KGE models. For the effects of knowledge incorporation, see detailed analysis in section 5.2.

(2) Our training framework can also facilitate in improving the recent state-of-the-art even further over QuatE on most evaluation metrics in link prediction and triplet classification tasks. It verifies the effectiveness of our proposed training framework.

5 Analysis

In this section, we provide further analysis of Pretrain-KGEs' performance in the case of fewer training triplets, low-frequency entities and the out-of-knowledge-base (OOKB) entities which are particularly hard to handle due to lack of knowledge representation. We also evaluate the effects of knowledge incorporation into entity and relation embeddings by demonstrating the sensitivity of MR and MRR metrics and visualizing the process of knowledge incorporation.

5.1 Performance on Fewer Training Triplets, Low-frequency and OOKB Entities

We also evaluate our training framework in the case of fewer training triplets on WordNet and test its performance on entities of varying frequency in test triplets on FB15K as well as the performance on the OOKB entities in test triplets on WordNet as shown in Fig. 2a- 2e.

To test the performance of our training framework given fewer training triplets, we conduct experiments on WN18 and WN18RR by feeding varying number of training triplets to a KGE model. We use traditional TransE as one of the baselines. Baseline-TransE does not utilize any textual description and randomly initializes entity and relation embeddings before the training phase. Thus, it suffers from learn knowledgeable KGEs when training triplets become fewer. In contrast, our Pretrain-TransE first learns knowledgeable entity and relation representations by encoding textual description through BERT, and uses the learned representations to initialize KGEs for TransE. In this way, we enable to incorporate rich linguistic knowledge from BERT into initizalized entity and relation embeddings so that TransE can perform better given fewer training triplets.

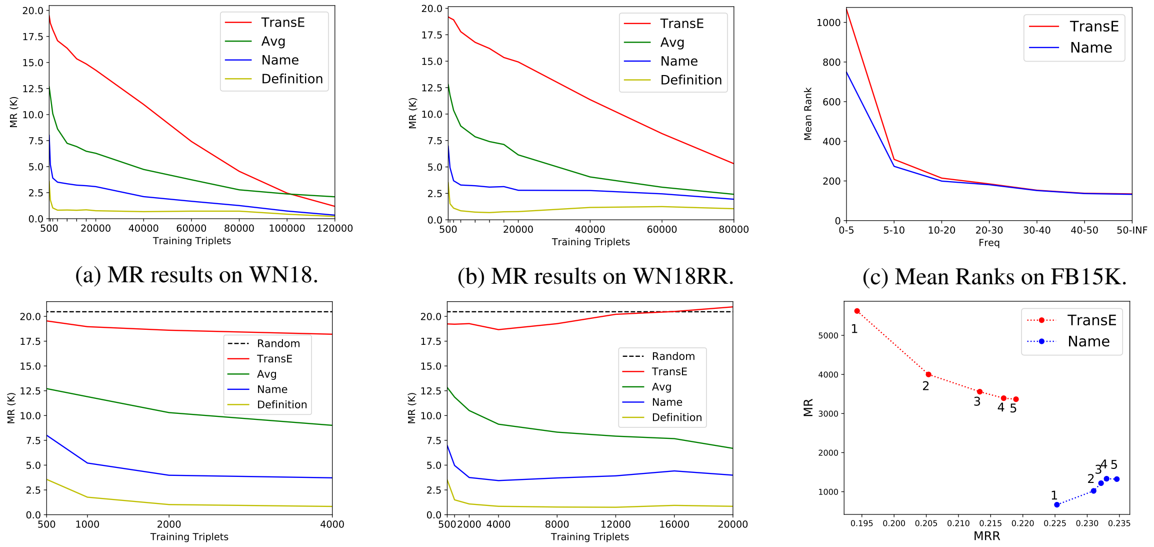
On the other hand, to verify the effectiveness of BERT during the fine-tuning phase, we also set the word averaging model following Li et al. (2016) to be the entity encoder $\mathrm{Enc}_e(\cdot)$ in Eq. 3 for comparison[4]. From the results, we can observe that although the word averaging model contributes to better performance of TransE on fewer training triplets compared to Baseline-TransE, it does not learn knowledgeable entity and relation representations as well as BERT because BERT can better understand textual descriptions of entities and relations by exploiting rich contextual information of the textual descriptions. Moreover, by utilizing definitions of word senses as additional textual description of entities, the results show that our training method achieves the best performance in the case of fewer training triplets.

Besides, we also evaluate our training framework for its performance on entities of varying frequency in training triplets on FB15K. From the results in Fig. 2c, we can observe that our training framework outperforms Baseline-TransE especially on infrequent entities. The reason is that traditional TransE method cannot learn good representation of infrequent entities due to inadquate dataset information and lack of textual description of entities.

When training triplets becomes fewer, there can be increasing OOKB entities in test triplets not observed at training time. Traditional training method of KGE models cannot address the OOKB entity problem since it randomly gives scores of test triplets containing OOKB entities due to random initialization of entity embeddings before training. In contrast, our training method initializes entity embeddings with knowledgeable entity representation. Thus, we also evaluate our training method

[3] Implementation and hyperparameter details are in Appendix.A.

[4] Implementation details of the word averaging baseline are in Appendix A.



(a) MR results on WN18.

(b) MR results on WN18RR.

(c) Mean Ranks on FB15K.

(d) MR results on OOKB of WN18.

(e) MR results on OOKB of WN18RR.

(f) Valid MR and MRR on WN18RR.

Figure 2: (a)&(b) show the MR results of different methods by varying training triplets number on WN18 and WN18RR. (c) compares the MR results of Baseline-TransE with our Pretrain-TransE by varying entity frequency in the training set of FB15K. (d)&(e) show the MR results of different methods on the out-of-knowledge-base (OOKB) entities on WN18 and WN18RR. (f) compares the changing results of MR and MRR on WN18RR between Baseline-TransE and Pretrain-TransE as iteration increases. In (a)-(e), "TransE" means TransE baseline with random initialization; "Avg" means a word averaging model using entity names and definitions provided in WordNet as textual description; "Name" refers to our proposed Pretrain-TransE method using entity names and relation names as textual description; "Definition" refers to our proposed Pretrain-TransE method using names of entities and relations as well as definitions in WordNet as textual description. In (d)-(e), "Random" means randomly giving scores of triplets. In (f), "1"-"5" denotes the number of iterations during the training phase are 10000-50000 updates.

in the case of OOKB entities. From the results in Fig. 2d- 2e, we can observe that our training framework can solve the OOKB entity problem on WordNet dataset and performs best when using BERT to encode textual description of entities and relations including their names and definitions of word senses.

5.2 Effects of Knowledge Incorporation

Our training framework has natural advantages over traditional training method of KGE models since we learn better knowledgeable entity and relation representation via BERT before training a KGE model. This section verifies the effectiveness of knowledge incorporation during the fine-tuning phase.

5.2.1 Analysis of the Sensitivity of MR and MRR

We show the performance of Baseline-TransE and Pretrain-TransE on WN18RR as iteration increases during the training phase in Fig. 2f. From the results, we can observe a new changing trend of MR and MRR for Pretrain-TransE. Compared to Baseline-TransE for which MR decreases (better performance) and MRR increases (better performance) at training time, Pretrain-TransE shows both increasing results of MR and MRR.

We analyze the changing trend of MR and MRR in Theorem 1. Formally, for n test triplets, we get corresponding ranks in link prediction task $\mathbf{r} = (r_1, r_2, \cdots, r_n)$, and $\mathrm{MR}(\mathbf{r}) = \sum_{i=1}^{n} r_i/n$; $\mathrm{MRR}(\mathbf{r}) = \sum_{i=1}^{n} r_i^{-1}/n$.

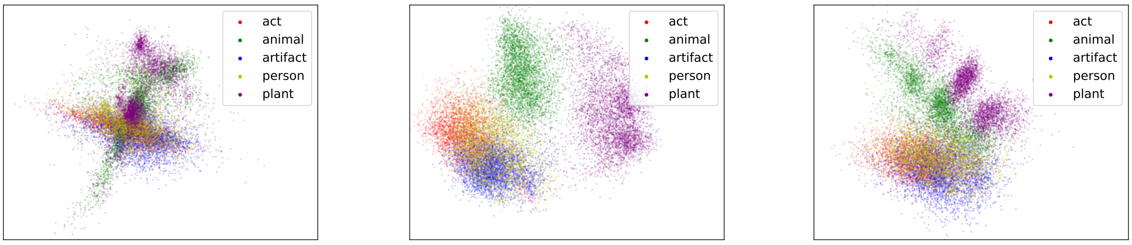
Theorem 1. [5]***Sensitivity of MR and MRR metrics*** *MR is more sensitive to tricky triplets than MRR. Formally, for $\boldsymbol{r} = (r_1, r_2, \cdots, r_n)$ and $r_i > r_j$ (triplet i is worse-learnt than triplet j):*

$$\frac{|MR_i'(\boldsymbol{r})|}{|MR_j'(\boldsymbol{r})|} > \frac{|MRR_i'(\boldsymbol{r})|}{|MRR_j'(\boldsymbol{r})|}$$

where $f_k'(\boldsymbol{r})$ denotes $\frac{\partial f}{\partial r_k}$ ($f \in \{MR, MRR\}$) and means the sensitivity of metric f to triplet k.

In Figure 2c, we can observe that there is better performance on high-frequency triplets than

[5] See detailed proof in Appendix B.



(a) Visualization of TransE.

(b) Visualization of Initialization.

(c) Visualization of Pretrain-TransE.

Figure 3: Visualization of knowledge incorporation. "Initialization" means our Pretrain-TransE model after the initializing phase and before the training phase. Different colors mark different supersenses in WordNet. Red (*noun.act*), yellow (*noun.person*) and blue (*noun.artifact*) denote senses relevant to *human beings*.

low-frequency ones which are more tricky to handle, since there is less information in datasets provided for low-frequency triplets. According to Theorem 1, we can thus suggest that MR is more sensitive to low-frequency triplets while MRR is more sensitive to high-frequency triplets. Reasons for the increasing MR of Pretrain-TransE in Fig. 2f are illustrated in the following.

5.2.2 Visualization of Knowledge Incorporation

We visualize the knowledge learning process of Baseline-TransE and our Pretrain-TransE in Fig. 3a-3c. We select top five common supersenses in WN18: *plant*, *animal*, *act*, *person* and *artifact*, among which the last three supersenses are all relevant to the concept of *human beings* and thus can be considered to constitute one common supersense.

In Fig. 3a, we can observe that Baseline-TransE learns entity and relation embeddings for triplets containing the five supersenses but does not distinguish embeddings between *plant*, *animal* and the other three supersenses. In contrast, Fig. 3b shows that our Pretrain-TransE can further distinguish embeddings between different supersenses, especially separating supersenses related to *human beings* from others. The main reason is that we can learn better knowledgeable entity and relation representation via BERT by incorporating rich linguistic knowledge into entity and relation embeddings during the initializing phase.

However, during the training phase, our Pretrain-TransE gradually learns different KGEs from those in the initializing phase. Fig. 3c shows that it is due to the oblivion of partial linguistic knowledge incorporated into entity and relation embeddings as the KGEs learn more information contained in datasets at training time. This process can account for the increasing MR results of Pretrain-TransE during the training phase in Fig. 2f. But the absolute values of MR and MRR for our Pretrain-TransE are overtly lower than those for TransE baseline, which demonstrates that our training framework enables to learn better knowledgeable entity and relation representation and there still remains incorporated knowledge in entity and relation embeddings during the training phase.

To conclude, during the training phase, TransE baseline learns original knowledge contained in datasets. Instead, our proposed method first learns rich linguistic knowledge from BERT, and continues to learn knowledge from datasets while losing partial knowledge learned from BERT. Finally, there still remains knowledge incorporated in entity and relation embeddings during the training phase.

6 Conclusion

We present Pretrain-KGEs, a simple and efficient pretraining technique for knowledge graph embedding models. Pretrain-KGEs is a general technique that can be applied to any KGE model. It contributes to learn better knowledgeable entity and relation representations from pretrained language models, which are leveraged during the initializing and the training phases for a KGE model to learn entity and relation embeddings. Through extensive experiments, we demonstrate state-of-the-art performances using this effective pretraining technique on various benchmark datasets. Further, we verify the effectiveness of our method by demonstrating promising results in the case of fewer training triplets, infrequent and OOKB entities which are particularly hard to handle due to lack of knowledge representation. We finally analyze the effects of knowledge incorporation by demonstrating the sensitivity of MR and MRR metrics and visualizing the process of knowledge incorporation.

Acknowledgments

This work is partly supported by Beijing Academy of Artificial Intelligence (BAAI). Xu Sun is the corresponding author.

References

Bo An, Bo Chen, Xianpei Han, and Le Sun. 2018. Accurate text-enhanced knowledge graph representation learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 745–755.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1819–1826.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 687–696.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4289–4300.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2181–2187.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

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

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

R. Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3679–3686.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2071–2080.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1112–1119.

Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu. 2017. SSP: semantic space projection for knowledge graph embedding with text descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3104–3110.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2659–2665.

Jiacheng Xu, Xipeng Qiu, Kan Chen, and Xuanjing Huang. 2017. Knowledge graph representation with jointly structural and textual encoding. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1318–1324.

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 *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embedding. *CoRR*, abs/1904.10281.

A Detailed Implementation

A.1 Implementation

Our implementations of TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019), pRotatE (Sun et al., 2019) are based on the framework provided by Sun et al. (2019)[6].

Our implementation of QuatE is based on on the framework provided by Zhang et al. (2019)[7].

In fine-tuning phase, we adopt the following non-linear pointwise function $\sigma(\cdot)$: for $x = x_0 + \sum\limits_{i=1}^{K-1} x_i\mathbf{e}_i \in \mathbb{F}$ (where $\mathbb{F}$ can be real number filed $\mathbb{R}$, complex number filed $\mathbb{C}$ or quaternion number ring $\mathbb{H}$):

$$\sigma(x) = \tanh(x_0) + \sum_{i=1}^{K-1} \tanh(x_i)\mathbf{e}_i \tag{7}$$

where $x_i \in \mathbb{R}$ and $\mathbf{e}_i$ is the K-dimension hypercomplex-value unit. For instance, when $K = 1, \mathbb{F} = \mathbb{R}$; when $K = 2, \mathbb{F} = \mathbb{C}$, $\mathbf{e}_1 = \mathbf{i}$ (the imaginary unit); when $K = 4, \mathbb{F} = \mathbb{H}$, $\mathbf{e}_{1,2,3} = \mathbf{i}, \mathbf{j}, \mathbf{k}$ (the quaternion units).

The score functions of baselines are listed in Table 4.

Method	Score function	$\mathbb{F}$
TransE (Bordes et al., 2013)	$\|v_h + v_r - v_t\|$	$\mathbb{R}$
DistMult (Yang et al., 2015)	$\langle v_h, v_r, v_t \rangle$	$\mathbb{R}$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle v_h, v_r, \bar{v}_t \rangle)$	$\mathbb{C}$
RotatE (Sun et al., 2019)	$\|v_h \odot v_r - v_t\|$	$\mathbb{C}$
pRotatE (Sun et al., 2019)	$2C\|\sin\frac{\theta_h+\theta_r-\theta_t}{2}\|$	$\mathbb{C}$
QuatE (Zhang et al., 2019)	$\|v_h \otimes \hat{v}_r \odot v_t\|$	$\mathbb{H}$

Table 4: Score functions and corresponding $\mathbb{F}$ of previous work. v_h, v_r, v_t denote head, tail and relation embeddings respectively. $\mathbb{R}, \mathbb{C}, \mathbb{H}$ denote real number field, complex number field and quaternion number division ring respectively. $\|\cdot\|$ denotes L1 norm. $\langle\cdot\rangle$ denotes generalized dot product. $\mathrm{Re}(\cdot)$ denotes the real part of complex number. $\bar{\cdot}$ denotes the conjugate for complex vectors. $\otimes$ denotes circular correlation, $\odot$ denotes Hadamard product. C denotes a constraint on the pRotatE model: $\|v_h\|_2 = \|v_t\|_2 = C$. $\hat{\cdot}$ denotes the normalized operator. $\theta_h, \theta_r, \theta_t$ denote the angle of complex vectors v_h, v_r, v_t respectively.

A.2 Implementation of Word-averaging Baseline

We also implement the word-averaging baseline to utilize the entitiy names and entity definition in WordNet to represent the entity embedding better. Formally, for entitiy e and its textual description $T(e) = w_1 w_2 \cdots w_L$, where w_i denotes the i-th token in sentence $T(e)$ and $T(e)$ here together utilizing the entitiy names and entity definition in WordNet.

$$\mathrm{Avg}(e) = \frac{1}{L}\sum_{i=1}^{L} u_i \tag{8}$$

where u_i denotes the word embedding of token w_i, which is a trainable randomly initialized parameter and will be trained in the pretraining phase.

We also adopt our three-phase training method to train word-averaging baseline. Similarly, $E = [E_1; E_2; \cdots; E_k] \in \mathbb{F}^{k\times d}$ and $R = [R_1; R_2; \cdots; R_l] \in \mathbb{F}^{l\times d}$ denote entity and relation embeddings. In pretraining phase, for head entity h, tail entity t and relation r, the score function is calculated as:

$$v_h, v_r, v_t = \mathrm{Avg}(h), R_r, \mathrm{Avg}(t) \tag{9}$$

$$\mathrm{Score} = \|v_h + v_r - v_t\| \tag{10}$$

[6] This responsibility: https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding

[7] This responsibility: https://github.com/cheungdaven/QuatE

where R_r denotes the relation embedding of relation r. In initializing phase, similar to our proposed model, we initialize E_i with $\mathrm{Avg}(e_i)$. In training phase, we optimize E and R with the same training method to TransE baseline.

A.3 Dataset Statistics

We evaluate our proposed training framework on four benchmark KG datasets: WN18 (Bordes et al., 2013), WN18RR (Dettmers et al., 2018), FB15K (Bordes et al., 2013) and FB15K-237 (Toutanova and Chen, 2015). We list detailed statistics of datasets are in Table 5.

Dataset	Ent	Rel	Train	Val	Test
WN18	40943	18	141442	5000	5000
WN18RR	40943	11	86835	3034	3134
FB15K	14951	1345	483142	50000	59071
FB15K-237	14541	237	272115	17535	20466

Table 5: Statisics of datasets. Ent and Rel denote the total number of entities and relations.

A.4 Experimental Settings

The hyper-parameters of are listed in Table 6.

Dataset	**Model**	Dim.	Dim.$\mathbb{R}$	Neg.1	Neg.2.	Batch.1.	Batch.2	Lr.1	Lr.2	Updates.1	Updates.2.	Opt.1	Opt.2
FB15K	TransE	1000	1000	3	256	8	1024	5e-6	1e-4	150k	150k	adam	adam
	DistMult	2000	2000	3	256	8	1024	5e-6	1e-3	150k	150k	adam	adam
	ComplEx	1000	2000	3	256	8	1024	5e-6	1e-3	150k	150k	adam	adam
	RotatE	1000	2000	3	256	8	1024	5e-6	1e-4	150k	150k	adam	adam
	pRotatE	1000	2000	3	256	8	1024	5e-6	1e-4	150k	150k	adam	adam
	QuatE	250	1000	10	20	4	50 batches	1e-5	0.1	40k	5000 epochs	adam	adagrad
FB15K-237	TransE	1000	1000	3	256	8	1024	5e-6	5e-5	150k	150k	adam	adam
	DistMult	2000	2000	3	256	8	1024	5e-6	5e-5	150k	150k	adam	adam
	ComplEx	1000	2000	3	256	8	1024	5e-6	5e-5	150k	150k	adam	adam
	RotatE	1000	2000	3	256	8	1024	5e-6	1e-3	150k	150k	adam	adam
	pRotatE	1000	2000	3	256	8	1024	5e-6	1e-3	150k	150k	adam	adam
	QuatE	100	400	10	10	6	10 batches	1e-5	0.1	200k	15000 epochs	adam	adagrad
WN18	TransE	500	500	3	512	8	512	5e-6	1e-4	80k	80k	adam	adam
	DistMult	1000	1000	3	512	8	512	5e-6	1e-3	80k	80k	adam	adam
	ComplEx	500	1000	3	512	8	512	5e-6	1e-3	80k	80k	adam	adam
	RotatE	500	1000	3	512	8	512	5e-6	1e-4	80k	80k	adam	adam
	pRotatE	500	1000	3	512	8	512	5e-6	1e-4	80k	80k	adam	adam
	QuatE	250	1000	10	20	1	10 batches	1e-5	0.1	200k/300k	1500 epochs	adam	adagrad
WN18RR	TransE	500	500	3	512	8	512	5e-6	5e-5	80k	80k	adam	adam
	DistMult	1000	1000	3	512	8	512	5e-6	2e-3	80k	80k	adam	adam
	ComplEx	500	1000	3	512	8	512	5e-6	2e-3	80k	80k	adam	adam
	RotatE	500	1000	3	512	8	512	5e-6	5e-5	80k	80k	adam	adam
	pRotatE	500	1000	3	512	8	512	5e-6	5e-5	80k	80k	adam	adam
	QuatE	100	400	10	20	8	10 batches	1e-5	0.1	60k/10k	40000 epochs	adam	adagrad

Table 6: Experimental settings. Dim. denotes embedding dimension. Dim.$\mathbb{R}$ denotes embedding dimension when embeddings are flatten into the real number filed. Batch. denotes batch size. Norm. denotes p-norm in score function, Lr. denotes learning rate. Neg. denotes entity negative sampling rate. 1. denotes in pretraining phase and 2. denotes in training phase and during the training of traditional embedding-based models. In column Batch.2, 50 batches means the dataset are devided into 50 batches. In column Updates.1, 200k/300k means 200k updates in Pretrain (Name) and 300k in Pretrain (Definition) model. In column Updates.2, 5000 epochs means the number of training updates is 5000 epochs.

B Proof of Theorem 1

Proof. According to definitions

$$\mathrm{MR}(\mathbf{r}) = \frac{1}{n}\sum_{i=1}^{n} r_i, \quad \mathrm{MRR}(\mathbf{r}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i}, \tag{11}$$

derive with respect to r_k,

$$\mathrm{MR}_k'(\mathbf{r}) = \frac{1}{n}, \quad \mathrm{MRR}_k'(\mathbf{r}) = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_k^2}, \tag{12}$$

therefore,

$$\frac{|\mathrm{MR}'_i(\mathbf{r})|}{|\mathrm{MR}'_j(\mathbf{r})|} = 1, \quad \frac{|\mathrm{MRR}'_i(\mathbf{r})|}{|\mathrm{MRR}'_j(\mathbf{r})|} = (\frac{r_j}{r_i})^2 < 1 \; (r_i > r_j), \quad \frac{|\mathrm{MR}'_i(\mathbf{r})|}{|\mathrm{MR}'_j(\mathbf{r})|} > \frac{|\mathrm{MRR}'_i(\mathbf{r})|}{|\mathrm{MRR}'_j(\mathbf{r})|} \tag{13}$$

□