# Pretrain-KGE: Learning Knowledge Representation from Pretrained Language Models

**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

Conventional knowledge graph embedding (KGE) often suffers from limited knowledge representation, leading to performance degradation especially on the low-resource problem. To remedy this, we propose to enrich knowledge representation via pretrained language models by leveraging world knowledge from pretrained models. Specifically, we present a universal training framework named *Pretrain-KGE* consisting of three phases: semantic-based fine-tuning phase, knowledge extracting phase and KGE training phase. Extensive experiments show that our proposed Pretrain-KGE can improve results over KGE models, especially on solving the low-resource problem.

## 1 Introduction

Knowledge graphs (KGs) constitute an effective access to world knowledge for a wide variety of NLP tasks, such as entity linking (Luo et al., 2017), information retrieval (Xiong et al., 2017), question answering (Hao et al., 2017) and recommendation system (Zhang et al., 2016). 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 relation $r$ as edges in the graph. A triplet represents the relation between two entities, e.g., *(Steve Jobs, founded, Apple Inc.)*. To learn effective representation of entities and relations in the graph, knowledge graph embedding (KGE) models are one of prominent approaches (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015; Sun et al., 2019; Nickel et al., 2011; Yang et al., 2015; Kazemi and Poole, 2018; Trouillon et al., 2016; Zhang et al., 2019).

However, traditional KGE models often suffer from limited knowledge representation due to the sparse and noisy dataset annotations. It leads to performance degradation, especially on the low-resource problem. To address this issue, we propose to enrich knowledge representation via pretrained language models (i.e., BERT (Devlin et al., 2019)) given a semantic description of entities and relations. We propose to incorporate world knowledge from BERT to the entity and the relation representation. Although simply fine-tuning BERT can enrich the knowledge representation, it suffers from learning inadequate structure information observed in training triplets, which we have demonstrated when we analyze the rationality of the KGE-training phase.

We propose a model-agnostic training framework for learning knowledge graph embedding consisting of three phases: semantic-based fine-tuning phase, knowledge extracting phase and KGE training phase (see Fig. 1). During the semantic-based fine-tuning phase, we learn knowledge representation via BERT given the semantic description of entities and relations as the input sequence. In this way, we incorporate world knowledge from BERT into the knowledge representation. Then during the knowledge extracting phase, we extract the entity and the relation representations encoded by BERT and inject them into embeddings of a KGE model. Finally, during the KGE training phase, we train the KGE model to learn adequate structure information of dataset, while reserving partial knowledge from BERT to learn better knowledge graph embedding.

Extensive experiments show that our proposed Pretrain-KGE can improve performance over KGE models on four benchmark KG datasets. Further analysis and visualization of the knowledge learning process demonstrate that our method can enrich knowledge representation via pretrained language models through the training framework.



*Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 259–266

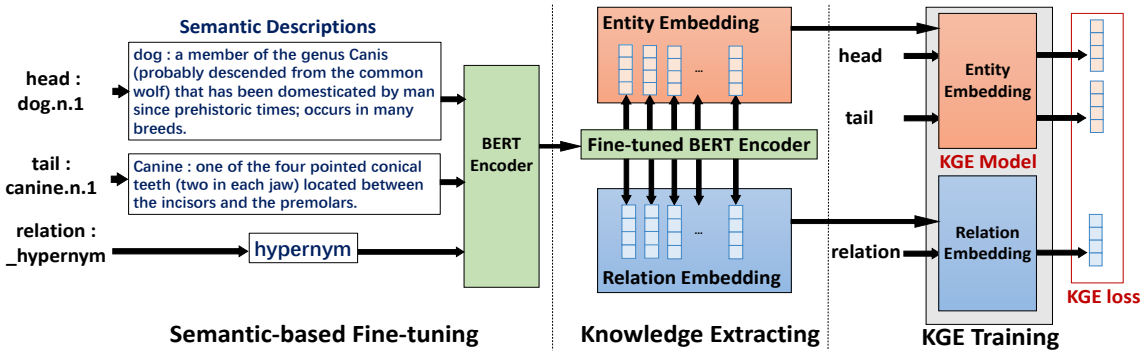



Figure 1: An illustration of our proposed three-phase Pretrain-KGE. "KGE loss" is the score function of an arbitrary KGE model, thus our method can be applied to any variant of KGE models. "BERT Encoder" represents the entity/relation encoder given semantic description of entities and relations.

## 2 Related Work

KGE models can be roughly divided into translational models and semantic matching models according to the score function (Wang et al., 2017). Translational models consider the relation between the head and tail entity as a translation between the two entity embeddings, such as TransE (Bordes et al., 2013), 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); while semantic matching models define a score function to match latent semantics of the head, tail entity and the relation, such as, RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015), SimplE (Kazemi and Poole, 2018), ComplEx (Trouillon et al., 2016) and QuatE (Zhang et al., 2019). QuatE (Zhang et al., 2019) is the recent state-of-the-art KGE model, which represents entities as hypercomplex-valued embeddings and models relations as rotations in the quaternion space.

In a knowledge graph dataset, the names of each entity and relation are provided as the semantic description of entities and relations. Recent works also leverage semantic description to enrich knowledge representation but ignore contextual information of the semantic description (Socher et al., 2013a; Li et al., 2016; Speer and Havasi, 2012; Xu et al., 2017; Xiao et al., 2017; Xie et al., 2016; An et al., 2018). Instead, our method exploits world information via pretrained models.

Recent approaches to modeling language representations offer significant improvements over embeddings, such as pretrained deep contextualized language models (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2019). KG-Bert (Yao et al., 2019) first utilizes BERT (Devlin et al., 2019) for knowledge graph completion, which treats triplets in knowledge graphs as textual sequences. However, KG-Bert does not extract knowledge representations from Bert and thus cannot provide entity or relation embeddings. In this work, we leverage world knowledge from BERT to learn better knowledge representation of entities and relations given semantic description.

## 3 Method

### 3.1 Training Framework

An overview of Pretrain-KGE is shown in Fig. 1. The framework consists of three phases: semantic-based fine-tuning phase, knowledge extracting phase, and KGE training phase.

**Semantic-based fine-tuning phase** We first encode the semantic description by BERT (Devlin et al., 2019). Define $S(e)$ and $S(r)$ as the semantic description of entity $e$ and relation $r$ respectively. BERT$(\cdot)$ converts $S(e)$ and $S(r)$ into the representation of entity and relation. We then project the entity and the relation representations into two separate vector spaces $\mathbb{F}^d$ through linear transformations, where $\mathbb{F}^d$ denotes a vector space on the number set $\mathbb{F}$. 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 $r$, then output the entity and the relation representations as:

$$\text{Enc}_e(e) = \sigma(W_e\text{BERT}(S(e)) + b_e) \quad (1)$$

$$\text{Enc}_r(r) = \sigma(W_r\text{BERT}(S(r)) + b_r) \quad (2)$$

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

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. $W_e, W_r \in$

$\mathbb{F}^{d\times n}, b_e, b_r \in \mathbb{F}^d$, and $\sigma$ denotes a nonlinear activation function.

The entity and the relation representations are used to train the BERT encoder based on a KGE loss. After fine-tuning, the entity encoder and the relation encoder are used in the following knowledge extracting phase.

**Knowledge extracting phase** In this phase, we extract knowledge representation encoded by BERT encoder and inject it into embedding of a KGE model as initialization: the entity embedding $E = [E_1; E_2; \cdots; E_k] \in \mathbb{F}^{k\times d}$; and the relation embedding $R = [R_1; R_2; \cdots; R_l] \in \mathbb{F}^{l\times d}$, where ";" means concatenating column vectors into a matrix, $k$ and $l$ denote the total number of entities and relations, respectively. Formally, we extract the knowledge representation encoded by BERT and inject it into a KGE model by setting $E_i$ to $\text{Enc}_e(e_i)$ and $R_j$ to $\text{Enc}_r(r_j)$.

**KGE training phase** After the knowledge extracting phase, we train a KGE model in the same way as a traditional KGE model. For example, if the max-margin loss function with negative sampling are adopted, the loss 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, $\gamma$ denotes the margin, $\left[\ \cdot\ \right]_+ = \max(\cdot, 0)$, and $f(\cdot)$ denotes the score function. The KGE training phase is indispensable because simply fine-tuning a pretrained language model cannot learn adequate structure information observed in training triplets. We demonstrate the rationality of the three-phase training framework in Section 5.2.

# 4 Experiments

## 4.1 Implementation of Baseline Models

To evaluate the universality of training framework Pretrain-KGE, 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;
- 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;

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 embeds 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.

Our implementations of TransE, DistMult, ComplEx, RotatE are based on the framework provided by Sun et al. (2019)[1]. Our implementation of QuatE is based on the framework provided by Zhang et al. (2019)[2]. The score functions of baselines are listed in Table 1.

| 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) | $\text{Re}(\langle v_h, v_r, \bar{v}_t\rangle)$ | $\mathbb{C}$ |
| RotatE (Sun et al., 2019) | $\|v_h \odot v_r - v_t\|$ | $\mathbb{C}$ |
| QuatE (Zhang et al., 2019) | $\|v_h \otimes \hat{v}_r \odot v_t\|$ | $\mathbb{H}$ |

Table 1: Score functions and corresponding $\mathbb{F}$. $v_h, v_r, v_t$ 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 $L_1$ norm. $\langle\cdot\rangle$ denotes generalized dot product. $\text{Re}(\cdot)$ and $\bar{\cdot}$ denote the real part and the conjugate for complex vectors respectively. $\otimes$ denotes circular correlation, $\odot$ denotes Hadamard product. $\hat{\cdot}$ denotes the normalized operator.

## 4.2 Datasets and Evaluation Metrics

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). Detailed statistics of datasets are in the appendix. WN18 and WN18RR are two subsets of WordNet (Miller, 1995); FB15K and FB15K-237 are two subsets of FreeBase (Bollacker et al., 2008). We use entity names and relation names provided by the four datasets as input semantic descriptions for BERT, and we also utilize synsets definitions provided by WordNet as additional semantic descriptions of entities.

[1] https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding

[2] https://github.com/cheungdaven/QuatE

| 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 2: 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.

| 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.

In our experiments, we perform the link prediction task (filtered setting) mainly with the triplet classification task. The link prediction task aims to predict either the head entity given the relation and the tail entity 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. We get ranks of test triplets 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).

### 4.3 Main Results

We present the main results of our Pretrain-KGE method in Table 2 and Table 3. As shown in Table 2, our universal training framework can be applied to multiple variants of KGE models despite different embedding spaces, and achieves improvements over TransE, DistMult, ComplEx, RotatE and QuatE on most evaluation metrics, especially on MR but still being competitive on MRR. The results in Table 3 demonstrate that our method can facilitate the performance of QuatE on most evaluation metrics for link prediction and triplet classification. The results verify the effectiveness of our proposed training framework and show that our universal training framework can be applied to multiple variants of KGE models and achieves improvements on most evaluation metrics, which shows the universality of our Pretrain-KGE.

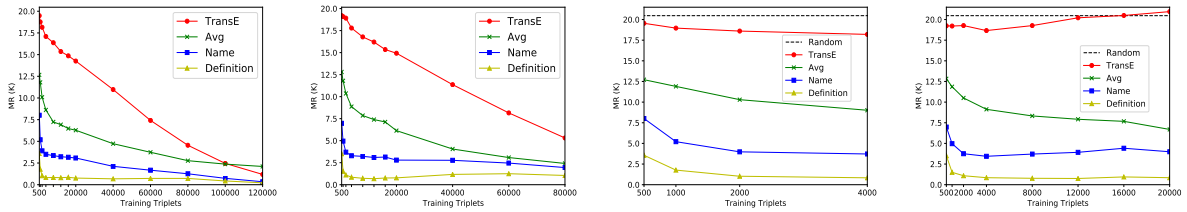
## 5 Analysis

In this section, we evaluate our Pretrain-KGE on the low-resource problem and further verify the rationality of our training framework.

### 5.1 Performance on the Low-resource Problem

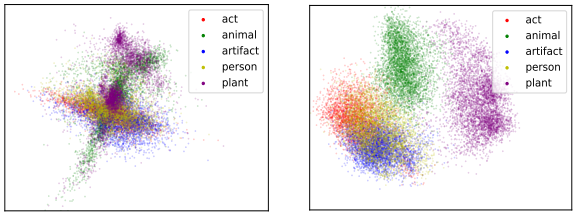
We evaluate our training framework in the case of fewer training triplets on WordNet, and test its performance on OOKB entities as shown in Fig. 2. To test the performance of our Pretrain-KGE given fewer training triplets, we conduct experiments on WN18 and WN18RR by feeding varying numbers of training triplets as shown in Fig. 2a and 2b. We also evaluate our Pretrain-KGE on WordNet for the OOKB entity problem as shown in Fig. 2c and 2d. We use traditional TransE and the word averaging model following Li et al. (2016) as baselines. Experimental details are in the appendix.

Results show that our training framework achieves the best performance in the case of fewer training triplets and OOKB entities. Baseline-TransE performs the worst when training triplets are few and cannot address the OOKB entity problem because it does not utilize any semantic de-




(a) MR results on WN18. (b) MR results on WN18RR. (c) OOKB MR on WN18. (d) OOKB MR on WN18RR.

Figure 2: Performance on the low-resource. "Random" and "Avg" denote a random and word averaging baseline.




(a) Baseline-TransE. (b) Pretrain-TransE.

Figure 3: Visualization of knowledge learning process. Different colors mark different supersenses in WordNet. Each point represents an entity. Red (*act*), yellow (*person*) and blue (*artifact*) refer to word senses relevant to *human beings*.

| **Model** | **FB15K** | | **FB15K-237** | |
|---|---|---|---|---|
| | MRR↑ | MR↓ | MRR↑ | MR↓ |
| Pretrain-TransE | **0.731** | **36.6** | **0.332** | **162.0** |
| w/o KGE training phase | 0.099 | 462.8 | 0.073 | 594.8 |
| **Model** | **WN18** | | **WN18RR** | |
| | MRR↑ | MR↓ | MRR↑ | MR↓ |
| Pretrain-TransE | **0.757** | **85** | **0.235** | 1747 |
| w/o KGE training phase | 0.086 | 1020 | 0.096 | **1444** |

Table 4: MRR results of the full Pretrain-KGE method and the ablation version ("w/o KGE training phase"). The experiments here use entity names and relation names as the semantic description.

scription. The word averaging model contributes to better performance of TransE on fewer training triplets, yet it does not learn knowledge representation as well as BERT because the latter can better understand the semantic description of entities and relations by exploiting world knowledge in the description. In contrast, our Pretrain-TransE can further enrich knowledge representation by encoding semantic description of entities and relations via BERT, and uses the learned representation to initialize the embedding for TransE. In this way, we can incorporate world knowledge from BERT into the entity and the relation embedding so that TransE can perform better given fewer training triplets and also alleviate the problem of OOKB entities.

### 5.2 Rationality of the Framework

We visualize the knowledge learning process of Baseline-TransE and our Pretrain-TransE in Fig. 3. 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*. In Fig. 3a, we can observe that Baseline-TransE learns the structure information in training triplets and does not distinguish *plant* and *animal* from the other three supersenses. In contrast, Fig. 3b shows that our Pretrain-TransE can distinguish entities belonging to different supersenses. Especially, entities relevant to the same concept *human beings* are more condensed and entities belonging to significantly different supersenses are more clearly separated. The main reason is that we introduce knowledge from BERT to enrich the knowledge representation of entities and relations.

We also demonstrate the rationality of the KGE-training phase. Table 4 shows that The full Pretrain-KGE method outperforms the ablation version which excludes the KGE training phase.

## 6 Conclusion

We propose Pretrain-KGE, an efficient pretraining technique for learning knowledge graph embedding. Pretrain-KGE is a universal training framework that can be applied to any KGE model. It learns knowledge representation via pretrained language models and incorporates world knowledge from the pretrained model into the entity and the relation embedding. Extensive experimental results demonstrate consistent improvements over KGE models across multiple benchmark datasets. The knowledge incorporation introduced in Pretrain-KGE alleviates the low-resource problem and we justify our three-phase training framework through an analysis of the knowledge learning process.

## 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.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 221–231.

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.

Angen Luo, Sheng Gao, and Yajing Xu. 2017. Deep semantic match model for entity linking using knowledge graph and text. In *2017 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2017, Shandong, China, October 19-21, 2017*, pages 110–114.

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.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1271–1279.

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*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 353–362.

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

# A Appendix

## A.1 Dataset Statistics

We evaluate our proposed training framework on four benchmark KG datasets: WN18, WN18RR, FB15K and FB15K-237. We list detailed statistics of datasets are in Table 5. Datasets can be downloaded at this repository[3].

| **Dataset** | Entities | Relations | Train Triplets | Valid. Triplets | Test Triplets |
|---|---|---|---|---|---|
| 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.

## A.2 Detailed Implementation

### A.2.1 Details in Semantic-based Fine-tuning Phase

In semantic-based fine-tuning phase, we adopt the following non-linear pointwise function $\sigma(\cdot)$: for

[3] https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding

| **FB15K** | Dim. | Dim.$\mathbb{R}$ | Neg.1 | Neg.2. | Batch.1. | Batch.2 | Lr.1 | Lr.2 | Updates.1 | Updates.2. | Opt.1 | Opt.2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| QuatE | 250 | 1000 | 10 | 20 | 4 | 50 batches | 1e-5 | 0.1 | 40k | 5000 epochs | adam | adagrad |
| **FB15K-237** | Dim. | Dim.$\mathbb{R}$ | Neg.1 | Neg.2. | Batch.1. | Batch.2 | Lr.1 | Lr.2 | Updates.1 | Updates.2. | Opt.1 | Opt.2 |
| 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 |
| QuatE | 100 | 400 | 10 | 10 | 6 | 10 batches | 1e-5 | 0.1 | 200k | 15000 epochs | adam | adagrad |
| **WN18** | Dim. | Dim.$\mathbb{R}$ | Neg.1 | Neg.2. | Batch.1. | Batch.2 | Lr.1 | Lr.2 | Updates.1 | Updates.2. | Opt.1 | Opt.2 |
| 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 |
| QuatE | 250 | 1000 | 10 | 20 | 1 | 10 batches | 1e-5 | 0.1 | 200k/300k | 1500 epochs | adam | adagrad |
| **WN18RR** | Dim. | Dim.$\mathbb{R}$ | Neg.1 | Neg.2. | Batch.1. | Batch.2 | Lr.1 | Lr.2 | Updates.1 | Updates.2. | Opt.1 | Opt.2 |
| 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 |
| 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 semantic-based fine-tuning phase and 2. denotes in KGE 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 the proposed model utilizing entity and relation names as semantic description and 300k in the proposed model utilizing entity and relation names as well as entity definition as semantic description. In column Updates.2, 5000 epochs means the number of training updates is 5000 epochs.

$x = x_0 + \sum_{i=1}^{K-1} x_i \mathbf{e}_i \in \mathbb{F}$ (where $\mathbb{F}$ can be real number filed $\mathbb{R}$, complex number filed $\mathbb{C}$ or quaternion number ring $\mathbb{H}$):

$$\sigma(x) = \tanh(x_0) + \sum_{i=1}^{K-1} \tanh(x_i)\mathbf{e}_i \quad (5)$$

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). For example:

$$\sigma\Big(\begin{bmatrix} a + b\mathbf{i} \\ c + d\mathbf{i} \end{bmatrix}\Big) = \begin{bmatrix} \tanh(a) + \tanh(b)\mathbf{i} \\ \tanh(c) + \tanh(d)\mathbf{i} \end{bmatrix} \quad (6)$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ denote the quaternion units.

#### A.2.2 Implementation of the Word-averaging Baseline

We implement the word-averaging baseline to utilize the entity names and entity definition in WordNet to represent the entity embedding better. Formally, for entity $e$ 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 entity names and entity definition in WordNet.

$$\text{Avg}(e) = \frac{1}{L} \sum_{i=1}^{L} u_i \quad (7)$$

where $u_i$ denotes the word embedding of token $w_i$, which is a trainable randomly initialized parameter and will be trained in the semantic-based fine-tuning 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 semantic-based fine-tuning phase, for head entity $h$, tail entity $t$ and relation $r$, the score function is calculated as:

$$v_h, v_r, v_t = \text{Avg}(h), R_r, \text{Avg}(t) \quad (8)$$

$$\text{Score} = \|v_h + v_r - v_t\| \quad (9)$$

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

### A.3 Experimental Settings

The hyper-parameters are listed in Table 6. Experiments are conducted on a GeForce GTX TITAN X GPU.