ASKRL: An Aligned-Spatial Knowledge Representation Learning Framework for Open-World Knowledge Graph

Ziyu Shang[1], Peng Wang[1,2](✉), Yuzhang Liu[1], Jiajun Liu[1], and Wenjun Ke[1,2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China
{ziyus1999,yuzhangliu,jiajliu,kewenjun}@seu.edu.cn

[2] Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Nanjing, China
pwang@seu.edu.cn

Abstract. Knowledge representation learning (KRL) aims to project entities and relations in knowledge graphs (KGs) to densely distributed embedding space. As the knowledge base expands, we are often presented with zero-shot entities, often with textual descriptions. Although many closed-world KRL methods have been proposed, most of them focus on connections between entities in the existing KGs. Therefore, they cannot handle zero-shot entities well, resulting in the inability of bringing zero-shot entities to existing KGs. To address this issue, this paper proposes ASKRL, a straightforward yet efficient open-world knowledge representation learning framework. ASKRL learns representations of entities and relations in both structured and semantic spaces, and subsequently aligns the semantic space with the structured space. To begin with, ASKRL employs the off-the-shelf KRL models to derive entity and relation embeddings in the structured embedding space. Afterward, a Transformer-based encoder is applied to obtain contextualized representations of existing entities and relations in semantic space. To introduce structure knowledge of KG into the contextualized representations, ASKRL aligns semantic embedding space to structured embedding space from the perspective of common properties (i.e., angle and length). Additionally, it aligns the output distribution of the score function between the two spaces. To further learn representations of zero-shot entities effectively, a sophisticated three-stage optimization strategy is devised in the training phase. In the inference phase, representations of zero-shot entities can be directly derived from the Transformer-based encoder. ASKRL is plug-and-play, enabling off-the-shelf closed-world KRL models to handle the open-world KGs. Extensive experiments demonstrate that ASKRL significantly outperforms strong baselines in open-world datasets, and the results illuminate that ASKRL is simple and efficient in modeling zero-shot entities.

Keywords: Knowledge Graph · Knowledge Representation Learning · Knowledge Graph Completion

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14265, pp. 101–120, 2023.
https://doi.org/10.1007/978-3-031-47240-4_6

1 Introduction

In knowledge graphs (KGs) [6,9], entities and relations are organized in a graph structured form. KGs consist of a large number of factual triples (h, r, t), where h and t represent head and tail entities, respectively, and r represents the relationship between h and t. Some large-scale KGs such as DBpedia [1], Freebase [2], and YAGO [20] have been widely used in many applications including natural language understanding [26], question answering [14], and recommender systems [27]. Meanwhile, many knowledge representation learning (KRL) methods [3,11,13,21,24] have been proposed to embed entities and relations into densely low-dimensional spaces.

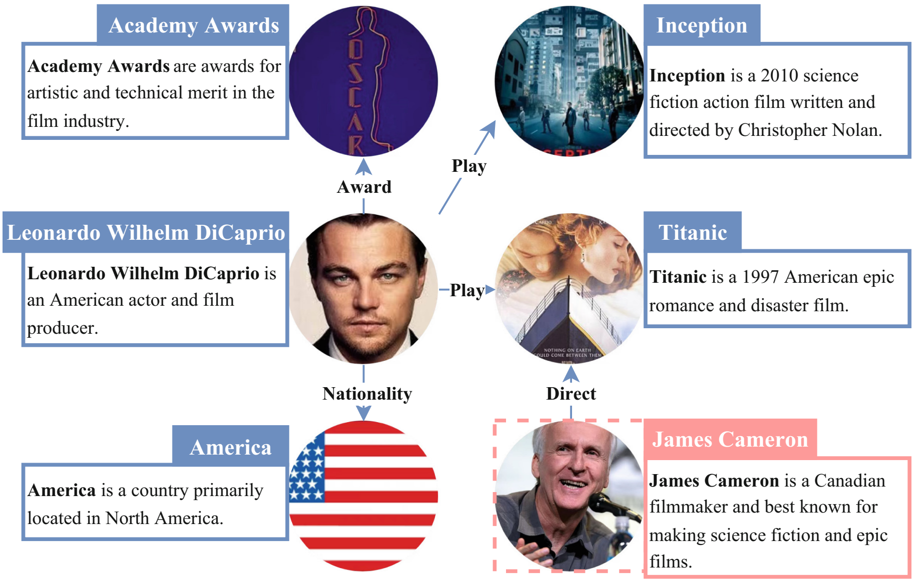

Fig. 1. Open-world knowledge graph examples with entity descriptions. The blue boxes represent in-KG entities and their descriptions, and the pink box represents the zero-shot entity and corresponding description to be added to the existing knowledge base. (Color figure online)

With the development of information extraction [10,30,36], many zero-shot (new) entities, which are out of the pre-defined entity set, have been mined. These zero-shot entities can empower the current KGs to provide more value for downstream applications. However, most of the existing KRL models follow closed-world assumption [16], which means that they can only handle the entities in the pre-defined entity set, and fail to process zero-shot entities. To handle zero-shot entities, the closed-world models must be retrained when zero-shot entities are added to existing KGs. In real-world scenarios, this paradigm is prone to inefficiency. Moreover, it is intractable to model the zero-shot entities with only the entity-self information such as entity name. To effectively process zero-shot entities, it usually needs more extra entity-related information. Fortunately, besides entities and relations, there are many textual descriptions of entities in most KGs. For a zero-shot entity, there is usually descriptive text as well [31]. As the example shown in Fig. 1, entity descriptions are informative, and the previously unseen entity *James Cameron* contains a textual description *James Cameron is a Canadian filmmaker and best known for making science fiction and epic films*. Valuable information is contained in this entity description such as *Canadian filmmaker*, *science fiction* and *epic films* to help to link with the correct entities in KGs, namely $(James\ Cameron, director, Titanic)$. However, such helpful entity descriptions providing further contextual information have not been exploited effectively in existing closed-world KRL models.

To model zero-shot entities, some open-world models exploiting entity descriptions have been proposed. DKRL [31] proposes using CNN to learn knowledge representations with both triples and descriptions. By this design, even though no embeddings have been learned for zero-shot entities, representations of zero-shot entities can be derived from their entity descriptions. Although DKRL can handle the zero-shot entities using entity descriptions, it neglects the noise of descriptions. To alleviate the impact of noisy entity descriptions, ConMask [19] proposes a relation-dependent content masking model to extract relevant content segments and then trains a CNN to model the extracted segments with entities in KGs. Despite the success of previous methods, they can only work for specific KRL models (e.g., TransE [3]) and cannot be migrated to other backbone models. To mitigate this problem, OWE [18] proposes an open-world extension for closed-world KRL models. OWE combines a regular closed-world KRL model learned from KGs with a simple word embedding model learned from the descriptions of entities. In OWE, the goal is to learn a mapping function from the textual description representations of entities to the structural representations learned by the closed-world KRL model. Therefore, closed-world KRL models are able to handle the open-world problem with this plug-and-play extension. However, in terms of the choice of the closed-world KRL model, it is not modular but is trained separately and independently from the contextualized representation model, so there may be a problem of error-propagating in OWE. Moreover, OWE attempts to map the word vector embedding space directly to the structural embedding space learned by the close-world KRL model via a linear layer. By mapping the semantic space directly to the structured space, we argue that a significant amount of semantic information is lost. In addition, the simple word embedding model might not be effective to model text descriptions. To deal with this deficiency, Caps-OWKG [29] applies capsule networks to encode entity descriptions better. It can be seen that these open-world models handle zero-shot entities by encoding entity descriptions. However, these models employ static word embedding trained on the domain?specific data, e.g., Wikipedia2Vec [33] is trained on Wikipedia. Therefore, if the zero-shot entities and their corresponding descriptions are derived from other domains, the existing models suffer from the generalization problem.

To address the above issues, we propose a novel Aligned-Spatial Knowledge Representation Learning framework ASKRL. This framework aligns the semantic space with the structural space from two distinct perspectives. The first perspective takes into consideration the common properties of different embedding spaces, i.e., the angle and length, while the second perspective focuses on the output distribution of distinct spaces on KRL. Specifically, ASKRL consists of three layers: a structured embedding layer, a description encoding layer, and an embedding space alignment layer. Firstly, it employs widely-used closed-world KRL models to learn the embeddings of entities and relations in the structured knowledge embedding layer. Then, it applies a Transformer-based model such as BERT [5] to encode the description of entities and relations, and then fine-tunes representations in the description encoding layer. Finally, in the embedding space alignment layer, we align common properties in different embedding spaces in a structure-to-structure manner, while aligning the output distribution of the KRL model on different embedding spaces in a distribution-to-distribution manner. Besides, to effectively learn representations of zero-shot entities, a three-stage optimization strategy is proposed. In the first stage, ASKRL focuses on optimizing the backbone KRL model in the structured embedding layer. In the second stage, ASKRL further optimizes the Transformer-based encoder in the description encoding layer and aligns semantic embedding space to structured embedding space in both structure-to-structure and distribution-to-distribution ways. In the third stage, in order to exploit the potential of the Transformer-based encoder, ASKRL only optimizes the description representation model to learn richer representations from textual descriptions. Because the Transformer-based encoder is usually trained on massive corpora by self-supervised tasks such as masked language model and next sentence prediction [5], they can learn rich prior knowledge within corpora. Thus the proposed ASKRL is more powerful than the previous several open-world models encoding the entity descriptions. Extensive experiments illuminate that ASKRL consistently achieves better performance than strong baseline models in widely-adopted open-world knowledge graph completion datasets. This evidence demonstrates that ASKRL is competent in modeling zero-shot entities.

In summary, our main contributions are twofold: (1) We propose a straightforward yet efficient open-world embedding framework, ASKRL, which proposes a novel approach to align two different embedding spaces to solve the zero-shot entity problem. (2) ASKRL is plug-and-play and can enable most KRL models under the closed-world assumption to be modular and efficient to produce embeddings of the zero-shot entities.

2 Related Work

2.1 Closed-World KRL Models

TransE [3] treats relations as translation operations from head entities to tail entities. It can model inverse and compositional relationship patterns, but it is too simple to handle complex relations such as 1-to-N, N-to-1, and N-to-N. To alleviate this problem, Yang et al. [34] proposed a bilinear diagonal model DistMult to capture the interaction between head entities, relations, and tail entities using the product of corresponding elements. DistMult can model symmetric relational patterns but fails to process the antisymmetric patterns. To solve antisymmetric patterns, Trouillon et al. [24] proposed ComplEx and introduced the concept of complex space to learn the representations of entities and relations in complex space. Liu et al. [13] proposed a triple-level self-attention model to handle the symmetric and antisymmetric patterns. Besides symmetric and antisymmetric patterns, Sun et al. [21] proposed RotatE to model more complex patterns including inversion, and composition. RotatE regards relations as rotations from source to target entities in complex space.

2.2 Open-World KRL Models

In real-world scenarios, entities and relations are constantly added, removed, or changed over time. The unseen added entities cannot be handled by closed-world models. Some open-world models have been proposed to model zero-shot entities. Xie et al. [31] first notice that there are descriptions for most zero-shot entities. To exploit entity descriptions, Xie et al. [31] proposed a model DKRL which employs CNN to encode entity descriptions and learns entity embedding from both entities and their descriptions. In this way, DKRL can model zero-shot entities using entity descriptions. To mitigate the effect of noise in descriptions, Shi et al. [19] proposed ConMask using a relation-dependent content masking mechanism to extract relevant segments and fuse them with entities in KGs by CNN. Recently, OWE [18] trains graph embeddings and text embeddings separately, and then learns the transformation function between the two embedding spaces. Because OWE ignores the unequal nature of different words in entity descriptions, WOWE [37] improves OWE by replacing the average aggregator with an attention mechanism to capture the weights of different words in entity descriptions. Furthermore, Caps-OWKG [29] combines text descriptions and KGs by using capsule networks to capture known and unknown triple features in open-world KGs.

2.3 Inductive KRL Models

The open-world setting emphasizes that the KRL model can deal with entities that are not seen during training, whereas the KRL model in the inductive setting focuses on the ability to leverage knowledge learned from the source KG to the target KG. Specifically, the sets of entities of the target and source KGs are disjoint, while the sets of relations are completely overlapping [22]. Therefore, inductive setting can be considered as a subset of open-world setting. Inductive KRL models can be divided into three main families: external resources-based models, logical rule-based models, and graph neural network-based models. The external resource-based models [4,28] mainly utilize accessible corpus about the target KG to assist in KRL. Logical rule-based methods [7,8] model logical rules with explicit frequent patterns, which are inherently inductive since logical rules are entity-independent. Neural LP [35] and DRUM [17] extract logical rules and confidence that are scored by differentiable rule learners in an end-to-end paradigm. Nevertheless, the neighbor structures surrounding the missing factual triples are ignored by these methods. Representative among the graph neural network-based models are GraIL [22] and SNIR [32], which extract the enclosing subgraphs around the target links to learn entity-independent features to deal with unseen entities.

3 Methodology

3.1 Preliminary

Definition 1. *(Knowledge Graph). A knowledge graph (KG) $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ is defined by the set of entities $\mathcal{E}$, relations $\mathcal{R}$, and triples $\mathcal{T}$. A triple is usually denoted as (h, r, t) $\in \mathcal{T}$, where $h \in \mathcal{E}$, $t \in \mathcal{E}$ and $r \in \mathcal{R}$ denote the head entity, and the tail entity, and the relation between them, respectively.*

Definition 2. *(Closed-World Assumption and Open-World Assumption). For a world $W(\mathcal{E}_W, \mathcal{R}_W)$, closed-world assumption believes that a KG $\mathcal{G}_C$ under the closed world is the closure of the world, which can be formulated as follows:*

$$\mathcal{G}_C = \bigcup_{\mathcal{G}' \in W} \mathcal{G}' \tag{1}$$

where the union is defined on the triple set. And for a KG $\mathcal{G}''$, if $\mathcal{G}'' \neq \mathcal{G}_C$, then $\mathcal{G}''$ is open-world KG under open-world assumption.

Keeping in line with the previous open-world KG setup [18], only new entities are required to appear and the type of relations does not change, i.e., for an open-world KG $\mathcal{G}''$, $\mathcal{E}_{\mathcal{G}''} \nsubseteq \mathcal{E}_{\mathcal{G}_C}$ and $\mathcal{R}_{\mathcal{G}''} \subseteq \mathcal{R}_{\mathcal{G}_C}$.

Definition 3. *(Open-Wrold Knowledge Graph Completion). Given an incomplete open-world Knowledge Graph $\mathcal{G}_O = \{\mathcal{E}_O, \mathcal{R}_O, \mathcal{T}_O\}$, where $\mathcal{E}_O \subset \mathcal{E}_W$ and $\mathcal{R}_O = \mathcal{R}_W$, open-world knowledge graph completion completes $\mathcal{G}_O$ by predicting a set of missing triple $\mathcal{T}' = \{(h, r, t) | (h, r, t) \notin \mathcal{T}_O, h \in \mathcal{E}_W, r \in \mathcal{R}_O, t \in \mathcal{E}_W\}$.*

In this paper, given a triple $(h,\ r,\ t)$, h, r, and t represent the head entity, relation, and tail entity, respectively, and $\mathbf{h}_{stru}, \mathbf{r}_{stru}, \mathbf{t}_{stru} \in \mathbb{R}^{d_s}$ denote their corresponding structured embedding vectors. For textual description, we denote the head entity description as $X_h = [x_1^h, x_2^h, ..., x_{|h|}^h]$, where $|h|$ is the description length of all words in the head entity, the tail entity description as $X_t =$

$[x_1^t, x_2^t, ..., x_{|t|}^t]$, where the $|t|$ stands for the length of all words in the tail description, and the relation description as $X_r = [x_1^r, x_2^r, ..., x_{|r|}^r]$, where the $|r|$ stands for the length of all words in the relation description. The $\mathbf{h}_{desc}, \mathbf{r}_{desc}, \mathbf{t}_{desc} \in \mathbb{R}^{d_t}$ denote the description representations of the head entity, relation, and tail entity, respectively.

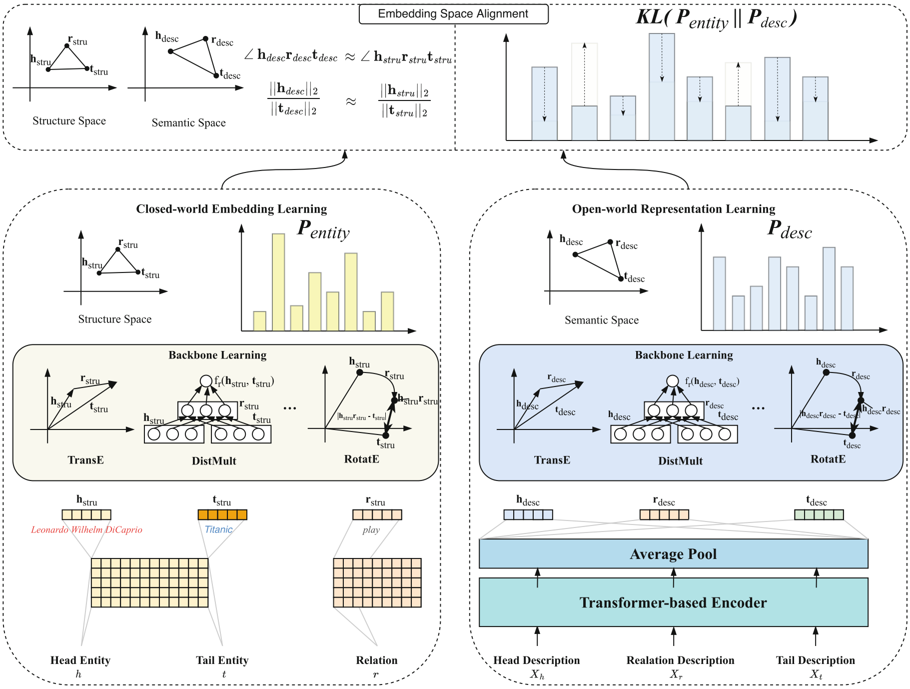

Fig. 2. The generic framework of ASKRL. The lower-left block is the component of closed-world embedding learning. The structured embedding is learned in this phase. The lower-right block is the component of open-world representation learning. The description representation is learned in this phase. The upper block is the component of embedding space alignment.

3.2 Framework Overview

This paper proposes a framework following the open-world assumption, namely ASKRL. There are three layers in ASKRL: the structured embedding layer, the description encoding layer, and the embedding space alignment layer. Firstly, the structured embedding layer is applied to learn embeddings of entities and relations in structured space. Then, the description encoding layer is used to produce embeddings of entities and relations in semantic space by encoding corresponding textual descriptions. Finally, in the embedding space alignment layer, we align the semantic space to the structured space in two steps. Firstly, we align the common properties of the semantic space and the structured space, such as angle and length. Concretely, ASKRL aligns the angle formed between the head, tail embeddings, and the relation embedding in the semantic space with the structured space, as well as aligning the ratio of the lengths of the head and tail embeddings with the structured space. Secondly, based on the score function defined in the structured embedding layer, ASKRL aligns the output distribution of the semantic space with the output distribution of the structured space. The overview framework of ASKRL is depicted in Fig. 2.

3.3 Structured Embedding Layer

The purpose of the structured embedding layer is to project entities and relations in KGs into densely embedding space. The procedure of this layer is similar to most closed-world KRL models. In other words, ASKRL can apply most closed-world KRL models to enable them for the open-world KGs. Specifically, we adopt the widely-used KRL models in this layer, including TransE [3], DistMult [34], ComplEx [21], and RotatE [21] as the backbone model.

The structured embedding of each entity can be learned as follows:

$$\mathbf{E}_{stru}, \mathbf{R}_{stru} = \phi(\mathbf{E}_{init}, \mathbf{R}_{init}) \tag{2}$$

where $\mathbf{E}_{init}$ and $\mathbf{R}_{init}$ denote the initial entity and relation embedding, respectively. They are usually initialized by a uniform initializer or Gaussian initializer, and ϕ denotes the transformation function of closed-world KRL models. In this way, for triple (h, r, t), we can obtain structured embedding $\mathbf{h}_{stru}$, $\mathbf{r}_{stru}$, $\mathbf{t}_{stru}$.

3.4 Description Encoding Layer

The description encoding layer aims to learn rich textual semantic information from entity and relation descriptions. In this paper, we apply a Transformer-based encoder [5] to encode the description. More specifically, we first use the Transformer-based encoder to encode the textual description, and then we input each word vector in the description representation obtained by the encoder to the average pooling layer, which can be formulated as follows:

$$\begin{aligned}\mathbf{E}_{desc} &= \text{AvgPool}(\text{Transformer-Enc}([x^{[\text{CLS}]}, X_{h/t}, x^{[\text{SEP}]}]))\\ \mathbf{R}_{desc} &= \text{AvgPool}(\text{Transformer-Enc}([x^{[\text{CLS}]}, X_{r}, x^{[\text{SEP}]}]))\end{aligned} \tag{3}$$

In this way, for triple (h, r, t), we can get the head entity description representation $\mathbf{h}_{desc}$, the relation description representation $\mathbf{r}_{desc}$, the tail entity description representation $\mathbf{t}_{desc}$.

3.5 Embedding Space Alignment Layer

For KRL, the Transformer-based models usually need to be trained longer than conventional KRL models (e.g., TransE) due to the larger number of model parameters [25]. Moreover, there is a lack of KGs structural information in the contextualized representations of entities and relations obtained by encoding the corresponding descriptions using a textual encoder. Intuitively, the embedding space learned by the conventional KRL model and learned by the text-based encoder is different, the former being the structure space and the latter the semantic space. To introduce graph structure knowledge into the contextualized representations of entities, previous methods attempt to learn the mapping function from the textual description representation space to the structure space by minimizing the L2 norm of the corresponding embedding of the same entity in the transformation space. Nevertheless, it is worth noting that the dimensions of the semantic space and the structure space are often different, and therefore, mapping the semantic space onto the structure space could lead to the loss of significant semantic information. To this end, we propose a new approach to aligning structure embedding space and semantic embedding space in a soft alignment manner, while can accelerate the training of text-based learning. Specifically, we argue that the common properties in different embedding spaces are angle and length [15]. Given triple $(h,\ r,\ t)$ in different embedding spaces, we have two objectives: firstly, to minimize the difference in the angles between the head entity, tail entity, and relation in different embedding spaces, and secondly, to minimize the differences in the ratio of the length between the head and tail entity in different embedding spaces. Therefore, in heterogeneous spaces, both the angle and length ratio can measure structural similarity, rather than the absolute replicability. Therefore, the objective of embedding space common property alignment can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{property} =& l_{\mathcal{H}}(f_A(\mathbf{h}_{stru}, \mathbf{r}_{stru}, \mathbf{t}_{stru}), f_A(\mathbf{h}_{desc}, \mathbf{r}_{desc}, \mathbf{t}_{desc})) \\ &+ l_{\mathcal{H}}(f_{DR}(\mathbf{h}_{stru}, \mathbf{t}_{stru}), f_{DR}(\mathbf{h}_{desc}, \mathbf{t}_{desc}))\end{aligned} \tag{4}$$

where $f_A(\mathbf{h}_\cdot, \mathbf{r}_\cdot, \mathbf{t}_\cdot) =< \frac{\mathbf{h}_\cdot - \mathbf{r}_\cdot}{||\mathbf{h}_\cdot - \mathbf{r}_\cdot||_2}, \frac{\mathbf{r}_\cdot - \mathbf{t}_\cdot}{||\mathbf{r}_\cdot - \mathbf{t}_\cdot||_2} >$ and $f_{DR}(\mathbf{h}_\cdot, \mathbf{t}_\cdot) = \frac{||\mathbf{h}_\cdot||_2}{||\mathbf{t}_\cdot||_2}$, $l_{\mathcal{H}}$ is Huber loss which is defined in Eq. 5.

$$l_{\mathcal{H}}(a,b) = \begin{cases} \frac{1}{2}(a-b)^2, & |a-b| \leq 1 \\ |a-b| - \frac{1}{2}, & |a-b| > 1 \end{cases} \tag{5}$$

In addition to the alignment of embedding space common properties, the output distribution of the score function in the semantic space and the output distribution of the score function in the structure space also need to be aligned. Taking the prediction of missing triples $(h,\ r,\ ?)$ as an example, we input the Transformer-based contextualized representation and structured embedding corresponding to the candidate entities into the score function $f_{\mathbf{r}}(\cdot,\cdot)$ defined in the structured embedding layer to obtain logits of the candidate entities $\mathbf{P}_{entity}$ and $\mathbf{P}_{desc}$, and then align $\mathbf{P}_{desc}$ to $\mathbf{P}_{entity}$ as follows:

$$\mathcal{L}_{output} = \mathcal{L}_{KL}(\mathbf{P}_{stru} \parallel \mathbf{P}_{desc}) = \sum_{i=1}^{m} \mathbf{P}^i_{stru} log \frac{\mathbf{P}^i_{stru}}{\mathbf{P}^i_{desc}} \tag{6}$$

where m is the candidate entity size, and $\mathbf{P}^i_{stru}$ and $\mathbf{P}^i_{desc}$ can be defined as follows:

$$\begin{aligned}\mathbf{P}^i_{stru} &= \frac{\exp f_{\mathbf{r}}(\mathbf{h}_{stru}, \mathbf{E}^i_{stru})}{\sum_{j=1}^{m} \exp f_{\mathbf{r}}(\mathbf{h}_{stru}, \mathbf{E}^j_{stru})} \\ \mathbf{P}^i_{desc} &= \frac{\exp f_{\mathbf{r}}(\mathbf{h}_{desc}, \mathbf{E}^i_{desc})}{\sum_{j=1}^{m} \exp f_{\mathbf{r}}(\mathbf{h}_{desc}, \mathbf{E}^j_{desc})}\end{aligned} \tag{7}$$

3.6 Training Optimization

Structured Embedding Optimization. We follow the settings of previous closed-world KRL models [21] to optimize the structured embedding layer. We first gather positive triples and build their corresponding negative samples to compute the rank-based hinge loss function. Considering each positive triple (h, r, t), the impact of their negative triples is different. We apply the self-adversarial negative sampling method [21] to measure the impact as follows:

$$p((h_j^{'}, r, t_j^{'})|(h_i, r_i, t_i)) = \frac{exp(\alpha f_r(\mathbf{h}^{'}_{stru_j}, \mathbf{t}^{'}_{stru_j}))}{\sum_i exp(\alpha f_r(\mathbf{h}^{'}_{stru_i}, \mathbf{t}^{'}_{stru_i}))} \tag{8}$$

where α denotes the temperature coefficient of sampling, $f_r(\mathbf{h}^{'}_{stru_i}, \mathbf{t}^{'}_{stru_i})$ denotes the score of the i-th negative triple in the negative sample candidate set, and $f_r(\mathbf{h}^{'}_{stru_j}, \mathbf{t}^{'}_{stru_j})$ denotes the score of the j-th negative sample. The hinge loss is calculated as follows:

$$\begin{aligned}\mathcal{L}_{stru} &= -log\ \sigma(\gamma - f_r(\mathbf{h}_{stru}, \mathbf{t}_{stru})) \\ &- \sum_{i=1}^{n} p(h_i^{'}, r, t_i^{'}) log\ \sigma(f_r(\mathbf{h}^{'}_{stru_i}, \mathbf{t}^{'}_{stru_i}) - \gamma)\end{aligned} \tag{9}$$

where γ denotes the margin of the hinge loss.

Transformer-Based Encoder Optimization. Similar to the structured embedding optimization, we also adopt the self-adversarial negative sampling to optimize the Transformer-based encoder, as follows:

$$p((h_j^{'}, r, t_j^{'})|(h_i, r_i, t_i)) = \frac{exp(\alpha f_r(\mathbf{h}^{'}_{desc_j}, \mathbf{t}^{'}_{desc_j}))}{\sum_i exp(\alpha f_r(\mathbf{h}^{'}_{desc_i}, \mathbf{t}^{'}_{desc_i}))} \tag{10}$$

The loss function is calculated as follows:

$$\begin{aligned}\mathcal{L}_{desc} &= -log\ \sigma(\gamma - f_r(\mathbf{h}_{desc}, \mathbf{t}_{desc})) \\ &- \sum_{i=1}^{n} p(h_i^{'}, r, t_i^{'}) log\ \sigma(f_r(\mathbf{h}^{'}_{desc_i}, \mathbf{t}^{'}_{desc_i}) - \gamma)\end{aligned} \tag{11}$$

where γ is the margin of the hinge loss.

3.7 Three-Stage Optimization Strategy

To model the zero-shot entities effectively, this paper proposes a three-stage optimization strategy. In the first stage, ASKRL is trained following the closed-world setting, i.e., only the structured embedding layer is trained in this stage. Specifically, it is optimized only using the structured embedding optimization and the loss is calculated by $\mathcal{L}_{stru}$. The first stage ends when the validation set's MRR score calculated by structured embedding essentially remains unchanged. In the second stage, the Transformer-based encoder is participated in encoding the descriptions of entities and relations via the Transformer-based encoder optimization $\mathcal{L}_{desc}$. It is usually more time-consuming for Transformer-based to fine-tune descriptions than structured embedding layer learning. To alleviate this problem and introduce graph structure knowledge into contextualized representations, we apply the embedding space alignment layer to align the semantic embedding space to the structure embedding space, and the loss in the second stage is calculated by $\mathcal{L}_{property} + \mathcal{L}_{output} + \mathcal{L}_{desc}$. The second stage ends when the validation set's MRR calculation using the text semantic embedding is substantially unaffected. In the third stage, to continuously activate the power of the Transformer-based encoder to encode the description, ASKRL is optimized only by $\mathcal{L}_{desc}$. The third stage finishes until the validation set's MRR calculated with the semantic embedding stays essentially unchanged. In a nutshell, the three-stage training objectives of the ASKRL can be formulated as follows:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{stru} & ,\text{FirstStage} \\ \mathcal{L}_{property} + \mathcal{L}_{output} + \mathcal{L}_{desc} & ,\text{SecondStage} \\ \mathcal{L}_{desc} & ,\text{ThirdStage} \end{cases} \tag{12}$$

4 Experiment

4.1 Datasets and Evaluation Metrics

To evaluate the performance of ASKRL, we conduct experiments on widely-used open-world knowledge graph completion datasets, including FB20k [31], DBPedia50k [19], and FB15k-237-OWE [18]. Particularly, FB20k is built on top of the FB15k [3] dataset by adding triples with new entities, which are selected to have long textual descriptions. DBPedia50k contains approximately 50k entities and is constructed from DBPedia [1]. FB15k-237-OWE is built upon the FB15k-237 [23] dataset, where redundant inverse relations have been removed and new entities with short descriptions are added. The statistics of these datasets are summarized in Table 1. The code of ASKRL and the datasets can be accessed via https://github.com/seukgcode/ASKRL.

We evaluate baselines and our proposed ASKRL in the open-world link prediction task. For a fair comparison, we evaluate the performance of all models on tail prediction following to [18]. We report the MRR (Mean Reciprocal Rank), and Hits@N (the proportion of correct entities ranked in the top N) metrics as most baselines do. Notice that metrics in the main experiment are reported in

Table 1. The statistics of datasets. $|E|$ stands for the entity size, $|R|$ denotes the relation size, $|L_{desc}|$ is the average length of all words in entity descriptions, and $|E^{open}|$ is the set of new entities which are not in KGs. Head Pred. is Head Prediction, and Tail Pred. denotes Tail Prediction.

Dataset	$\|R\|$	$\|E\|$	$\|E^{open}\|$	$\|L_{desc}\|$	Number of triples				
					Train	Head Pred.		Tail Pred.	
						Valid	Test	Valid	Test
FB20k	$1,341$	$149,04$	$5,019$	147	$472,860$	$1,800$	$18,753$	$1,000$	$11,586$
DBPedia50k	351	$24,624$	$3,636$	454	$32,388$	55	$2,139$	164	$4,320$
FB15k-237-OWE	235	$12,324$	$2,081$	5	$242,489$	$1,539$	$13,857$	$9,424$	$22,393$

the target filter setting [19] for a fair comparison with baselines. In the target filter setting, when evaluating a test triple (h, r, t), a candidate tail t' is only included in the ranked result list if the triple $(?, r, t')$ exists in the training data, otherwise, it is removed.

4.2 Baselines

We compare the proposed model ASKRL with the following widely-adopted open-world state-of-the-art models: DKRL [31] uses a two-layer CNN to encode entity descriptions. ConMask [19] employs the relation-dependent content masking mechanism to extract relevant content description segments and applies CNNs to encode the entity descriptions. OWE [18] maps the text-based entity description representation to the pre-trained graph embedding space. WOWE [37] applies the attention mechanism instead of the average aggregator to model entity descriptions. Caps-OWKG [29] uses the capsule network to capture known and unknown triples features in open-world KGs.

4.3 Implementation Details

All experiments are obtained on the single NVIDIA RTX 3090Ti GPU. The hyper-parameters are tuned by grid search and the range of hyper-parameters is set as follows: embedding size $d \in \{300, 400, 500, 600, 1000\}$, the initial learning rate of backbone models: $lr_1 \in \{1e-3, 2e-3, 3e-3, 4e-3, 5e-3\}$, the initial learning rate of Transformer-based encoder: $lr_2 \in \{1e-4, 5e-4, 1e-5, 5e-5\}$, batch size $b \in \{16, 32, 256, 512, 1024\}$, temperature coefficient $\alpha \in \{0.1, 0.5, 1.0\}$, and margin $\gamma \in \{3, 6, 9, 12, 18, 24\}$. There are two types of negative samples in the training process: (1) Other entities within the same batch as negative samples. (2) The current entity as a difficult negative sample, e.g., taking the predicted tail entity (h, r, t) as an example, we consider (h, r, h) as a difficult negative sample. We use the popular library HuggingFace Transformers[1] to load the Transformer-based encoder and fine-tune it. For the default setting, we apply the RotatE as

[1] https://github.com/huggingface/transformers.

the KRL model and BERT-base-uncased [5][2] as the default Transformer-based encoder. In our experiments, we utilize the textual descriptions of entities that are available in the dataset, while the relation name is considered as the textual description of the corresponding relation. For those entities in the datasets that do not have textual descriptions, we directly use text mentions of their name as the corresponding descriptions.

Table 2. Tail prediction results of baselines models on FB20k, DBPedia50k, and FB15k-237-OWE datasets (with target filter). † stands for results obtained from [18] , and ‡ denotes results retrieved from original papers. The numbers in bold indicate the best performances, whereas the second-best performances are underlined

Model	FB20k				DBPedia50k				FB15k-237-OWE			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Target Filt. Base. †	27.2	17.5	32.1	41.2	11.0	4.5	9.7	23.0	12.7	6.4	14.2	23.3
DKRL †	–	–	–	67.4	23.0	–	–	40.0	–	–	–	–
ConMask †	53.3	42.3	57.3	71.7	58.4	47.1	64.5	81.0	29.9	21.5	39.9	45.8
OWE ‡	53.1	44.8	57.1	69.1	60.3	51.9	65.2	76.0	40.1	31.6	43.9	56.0
Caps-OWKG ‡	–	–	–	–	59.6	48.8	64.8	75.8	35.2	25.5	39.0	50.8
WOWE ‡	54.1	45.2	58.3	70.0	61.2	52.7	66.5	76.9	40.4	31.9	44.1	56.4
ASKRL-LSTM	56.2	47.3	61.4	75.2	61.3	52.3	66.5	77.9	42.4	33.5	45.9	58.2
ASKRL-BERT$_{base}$	**61.5**	**52.9**	**69.8**	**81.4**	**68.7**	**60.3**	**73.6**	**82.9**	**44.3**	**34.5**	**47.6**	**61.8**

4.4 Main Results

We evaluate models with different settings to provide a multi-perspective analysis. In Table 2, we compute metrics in the target filter setting. We can see that our proposed ASKRL-BERT$_{\text{base}}$ achieves new state-of-the-art performance in Table 2. ASKRL-BERT$_{\text{base}}$ outperforms all baselines on the FB20k dataset. More specifically, ASKRL-BERT$_{\text{base}}$ achieves approximately 7.4% gain to 61.5 on MRR against WOWE. For the DBPedia50k dataset, ASKRL-BERT$_{\text{base}}$ achieves approximately 7.5% gain to 68.7 on MRR against WOWE. Besides, we can also find that ASKRL-BERT$_{\text{base}}$ achieves a 3.9% gain to 44.3 on MRR than WOWE on the FB15k-237-OWE dataset. Furthermore, to verify whether the performance of ASKRL is brought by the pre-trained language model, we replace BERT$_{\text{base}}$ with an LSTM encoder and use 300-dimensional Wiki-pedia2Vec word embedding. The results are reported in Table 2. We can see that ASKRL-LSTM beats most of the baselines except Hits@3 and Hits@10 on the DBPedia50k dataset, which shows the effectiveness of the proposed approach of aligning two different representation spaces by aligning both common properties and the output distribution of the score function. However, compared to on the FB20k and DBPedia50k datasets, using BERT$_{\text{base}}$ on the FB15k-237-OWE is not a significant improvement compared to using LSTM. The reason for this can be seen in

[2] https://github.com/google-research/bert.

Table 1, where the average length of all words in entity descriptions for FB15k-237-OWE is 5. Most of the descriptions are even just entity names, which do not contain more additional information that can be used.

Apart from the target filter setting, we also report the metrics in the normal setting (without target filter) on the FB15k-237-OWE dataset in Table 3. We can find that ASKRL-BERT$_{\text{base}}$ still consistently achieves better results than the baseline model OWE in the normal setting. Notably, ASKRL-BERT$_{\text{base}}$ achieves 4.6% gain than OWE on the MRR, which suggests that ASKRL-BERT$_{\text{base}}$ can process zero-shot entities effectively. Besides, the MRR of ASKRL-LSTM is improved by 2.9% compared to OWE, which indicates that the performance of ASKRL is not entirely attributable to the pre-trained Transformer-based encoder.

Table 3. Tail prediction results on the FB15k-237-OWE dataset without target filter. † marks results retrieved from the original paper.

Model	MRR	H@1	H@3	H@10
OWE †	35.2	27.8	38.6	49.1
ASKRL-LSTM	**38.1**	**30.5**	**41.2**	**52.8**
ASKRL-BERT$_{\text{base}}$	**39.8**	**31.0**	**43.9**	**57.6**

Table 4. Comparison of ASKRL based on different KRL models on the FB15k-237-OWE (with target filter).

Model	MRR	H@1	H@3	H@10
ASKRL-TransE	38.3	30.6	42.3	53.9
ASKRL-DistMult	39.8	31.2	43.3	55.6
ASKRL-ComplEx	40.3	32.2	43.9	55.6
ASKRL-RotatE	**44.3**	**34.5**	**47.6**	**61.8**

4.5 Results of Ablation Study

Effect of KRL Models. One of our critical insights is to design a plug-and-play strategy that can be seamlessly adapted to diverse KRL methods, such as TransE and RotatE. Specifically, ASKRL encodes the raw entity semantics with a Transformer-based encoder and augments their comparability in the perspective of spatial aspects (angles and length ratios). Such a design can be easily integrated as an auxiliary loss to fine-tune the BERT efficiently. Despite the different geometric suppositions made by different structured-based KRL models, their corresponding output distribution remains generally applicable across the semantic-spatial transformation. Thus, the alignment space mechanism is also applicable. To verify the effectiveness of ASKRL for different KRL models, we conduct experiments on different backbone KRL models. The results are shown in Table 4. It can be seen that ASKRL can be applied to off-the-shelf KRL models with different geometrical assumptions and that the results are slightly different in terms of different backbone KRL models. In general, the performance ranking of KRL models is as follows: RotatE > ComplEx > DisMult > TransE. Our results follow the above rank. In our results, RotatE-based ASKRL achieves the best results.

Table 5. Comparison results of different Transformer-based encoders on FB15k-237-OWE dataset (with target filter).

Encoder	MRR	H@1	H@3	H@10
ASKRL-BERT$_{\text{tiny}}$	34.0	27.1	37.6	48.9
ASKRL-BERT$_{\text{small}}$	42.5	32.1	45.7	59.5
ASKRL-BERT$_{\text{base}}$	44.3	34.5	47.6	61.8
ASKRL-BERT$_{\text{large}}$	**44.9**	**35.0**	**47.9**	**62.3**
ASKRL-RoBERTa$_{\text{base}}$	**45.3**	**36.2**	**48.5**	**63.8**
ASKRL-RoBERTa$_{\text{large}}$	44.7	35.8	47.6	62.7

Effect of Transformer-Base Encoder Models. Furthermore, we conduct experiments on the effect of different Transformer-base encoders on ASKRL. We use two families of Transformer-base encoders, i.e., BERT [5] and RoBERTa [12], with ASKRL. In general, RoBERTa is superior to BERT in terms of the performance of pre-trained language models on downstream tasks [5,12]. The results, shown in Table 5, fit with our common sense. Specifically, ASKRL-RoBERTa$_{\text{base}}$ achieves the best MRR in all encoder settings, with a 0.4% improvement compared to the second-best result. However, we find that ASKRL-RoBERTa$_{\text{large}}$ does not perform better than ASKRL-RoBERTa$_{\text{base}}$. The main reason could be that there are too many parameters in RoBERTa$_{\text{base}}$, making it difficult to train to converge to a globally optimal parameter.

Effect of Embedding Space Alignment Layer. To explore the effect of aligning the semantic space to the structure space by aligning the common properties and the output distribution of score function proposed in this paper, we removed the corresponding parts from the default settings of ASKRL. The experimental results are shown in Table 6. After removing the alignment of spatial properties (refer to -$\mathcal{L}_{property}$), the performance of ASKRL drops significantly on all datasets. Concretely, the MRR of ASKRL decreases by 3.1%, 3.2%, and 1.4%

Table 6. Experimental results on the ablation of embedding space alignment layer, where Time refers to the time (hours) required to train the model until the MRR of validation set is essentially unchanged.

Model	FB20k			DBPedia50k			FB15k-237-OWE		
	MRR	H@10	Time	MRR	H@10	Time	MRR	H@10	Time
ASKRL-BERT$_{base}$	61.5	81.4	20.58 h	68.7	82.9	8.97 h	44.3	61.8	15.30 h
-$\mathcal{L}_{property}$	58.4	79.6	20.36 h	65.5	80.9	8.33 h	42.9	59.3	14.62 h
-$\mathcal{L}_{output}$	59.1	80.2	20.49 h	66.8	81.5	8.67 h	43.8	60.6	15.21 h
BERT$_{\text{base}}$	50.0	63.8	28.33 h	58.6	70.3	12.90 h	35.8	50.0	19.63 h
BERT$_{\text{large}}$	51.3	64.2	35.95 h	59.9	71.6	16.47 h	36.3	51.2	25.71 h

on FB20K, DBPedia50K, and FB15k-237-OWE datasets, respectively, compared to the default setting. Meanwhile, after removing the alignment of output distribution (refer to -$\mathcal{L}_{output}$), the performance degradation of ASKRL is not too significant, which indicates that it is necessary to introduce the common structural information of the triples into the semantic space.

Furthermore, we remove the entire embedding space alignment layer, and then only optimize $\mathcal{L}_{desc}$ (refer to BERT$_{\text{base}}$ and BERT$_{\text{large}}$). The following conclusions can be drawn: (1) Optimizing solely $\mathcal{L}_{desc}$ results in a significant decline in performance for ASKRL, suggesting that relying solely on features from the semantic space is inadequate for modeling zero-shot entities. (2) There is a substantial difference in training time between BERT$_{base}$ and ASKRL-BERT$_{base}$ in FB20K, DBPedia50k, and FB15k-237-OWE datasets, with the former taking as much as 37.7%, 43.8%, and 28.3% longer than the latter, respectively. This suggests that our proposed embedding space alignment layer can provide a more explicit optimization direction and accelerate the convergence of the Transformer-based encoder.

Effect of the Three-Stage Optimization Strategy. This paper introduces a three-stage optimization strategy to train the model effectively. Each stage includes unique and luminous optimization targets, i,e., structured, semantic, and continual fine-tuning relative to the 1st, 2nd, and 3rd stages, respectively. Meanwhile, such a pipeline training strategy makes ASKRL converge successfully and efficiently. Moreover, the effectiveness of the three-stage optimization strategy has also been validated in the experiments. In the first optimization stage, the structured KRL model is trained to learn representations of entities and relations. After training, the backbone model will learn an experienced prior distribution. In the second optimization stage, the experienced prior distribution is applied to the embedding space alignment, in which the distribution of the Transformer-based encoder is aligned to the experienced prior distribution of the trained structured KRL model. Moreover, the average MRR experiences a decrease of 10.0% without the first and second Stages. This design not only accelerates the convergence of the Transformer-based encoder, but also introduces the knowledge of the KGs structure into the textual encoder. In our experiment, it usually needs to be trained five or more epochs to converge for the Transformer-based encoder without the embedding space alignment layer. In the third optimization stage, the Transformer-based encoder is continuously trained to achieve better performance. In our experiment, there is approximately 0.2% to 0.7% improvement in MRR with the third stage optimization than without it.

Table 7. Case Study: The actual tail prediction results on the FB15k-237-OWE dataset of ASKRL

Test Triples	Head Description	Top-k Predicted Tails
(**The Mask of Zorro**, /film/film/language, **English**)	1998 American swashbuckler film	**1.English** 2.Russian 3.French 4.Spanish 5.United States of America
(**Bury My Heart at Wounded Knee**, /film/film/language, **English**)	2007 US TV film	**1.English** 2.Library of Congress Classification 3.Birdie Kim 4.United States of America 5.French
(**Daytona Beach**, /base/biblioness /bibs_location/country, **United States of America**)	city in Florida, United States	1.Library of Congress Classification **2.United States of America** 3.actor 4.CE Campos 5.EA Vancouver
(**Thomas Jefferson**, /influence/influence_node /peers./influence /peer_relationship/peers , **John Adams**)	3rd President of the United States of America	1.Europe 2.marriage 3.New York University 4.New York City **132.John Adams**
(**Christopher McDonald**, /people/person /spouse_s./people /marriage/type_of_union, **marriage**)	American actor	1.actor 2.film producer 3.United States of America 4.Warner Bros. **249.marriage**

4.6 Case Study

To intuitively explain how ASKRL solves the zero-shot entities, we provide some prediction cases of ASKRL on the FB15k-237-OWE dataset, as shown in Table 7.

Supposed that *The Mask of Zorro* is a zero-shot entity with description *1998 American swashbuckler film*. Because the *The Mask of Zorro* does not belong to the predefined entity set, the entity-based structured backbone model cannot encode it. At this time, ASKRL can encode its description *1998 American swashbuckler film* by the Transformer-based encoder to obtain its embedding. Table 7 shows that the top-1 tail prediction of ASKRL is *English* when the input relation is */film/film/language*, which is equal to the ground-truth tail entity. We can also see that the first three examples show good performance: the ground truth is in the top five predicted results. These actual prediction cases intuitively prove that the proposed model ASKRL is capable of modeling zero-shot entities effectively.

However, we notice that some complicated relations consist of multiple sub-relations in the FB15k-237-OWE dataset. For instance, there are two sub-relations: */people/person/spouse_s* and */people/marriage/type_of_union* in the relation */people/person/spouse_s./people/marriage/type_of_union*. It is still challenging for ASKRL to handle relations with multiple sub-relations, e.g., the fourth and fifth examples.

5 Conclusion

In this paper, we propose a novel model ASKRL to handle the knowledge representation learning of zero-shot entities. For given zero-shot entities, ASKRL uses the Transformer-based encoder to encode their descriptions as input. ASKRL can be a plug-and-play extension for off-the-shelf closed-world KRL models to enable them to handle zero-shot entities. We conduct extensive experiments on widely-used open-world KGC datasets to demonstrate the effectiveness of ASKRL.

References

1. Auer, S., Bizer, C., Kobilarov, G., et al.: Dbpedia: a nucleus for a web of open data. In: Proceedings of the 6th International Semantic Web Conference (2007)
2. Bollacker, K., Evans, C., Paritosh, P., et al.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 27th ACM SIGMOD International Conference (2008)
3. Bordes, A., Usunier, N., García-Durán, A., et al.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 27th Neural Information Processing Systems (2013)
4. Daza, D., Cochez, M., Groth, P.: Inductive entity representations from text via link prediction. In: Proceedings of the Web Conference 2021 (2021)
5. Devlin, J., Chang, M., Lee, K., et al.: 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 (2019)
6. Dong, X.L., Gabrilovich, E., Heitz, G., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD International Conference (2014)

7. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. VLDB J. **24**(6), 707–730 (2015)
8. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the Web Conference 2013 (2013)
9. Hogan, A., Blomqvist, E., Cochez, M., et al.: Knowledge graphs. ACM Comput. Surv. **54**(4), 1–37 (2021)
10. Li, X., Luo, X., Dong, C., et al.: TDEER: an efficient translating decoding schema for joint extraction of entities and relations. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021)
11. Li, Z., Liu, H., Zhang, Z., Liu, T., Xiong, N.N.: Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Trans. Neural Netw. Learn. Syst. **33**(8), 3961–3973 (2022)
12. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
13. Liu, Y., Wang, P., Li, Y., et al.: Aprile: attention with pseudo residual connection for knowledge graph embedding. In: Proceedings of the 28th International Conference on Computational Linguistics (2020)
14. Mohammed, S., Shi, P., Lin, J.: Strong baselines for simple question answering over knowledge graphs with and without neural networks. In: Proceedings of the 16th North American Chapter of the Association for Computational Linguistics (2018)
15. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
16. Reiter, R.: On closed world data bases. In: Webber, B.L., Nilsson, N.J. (eds.) Readings in Artificial Intelligence, pp. 119–140. Morgan Kaufmann (1981)
17. Sadeghian, A., Armandpour, M., Ding, P., Wang, D.Z.: Drum: end-to-end differentiable rule mining on knowledge graphs. In: Proceedings of the 33th Neural Information Processing Systems (2019)
18. Shah, H., Villmow, J., Ulges, A., et al.: An open-world extension to knowledge graph completion models. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (2019)
19. Shi, B., Weninger, T.: Open-world knowledge graph completion. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (2018)
20. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web (2007)
21. Sun, Z., Deng, Z., Nie, J., et al.: Rotate: knowledge graph embedding by relational rotation in complex space. In: Proceedings of the 7th International Conference on Learning Representations (2019)
22. Teru, K.K., Denis, E.G., Hamilton, W.L.: Inductive relation prediction by subgraph reasoning. In: Proceedings of the 37th International Conference on Machine Learning (2020)
23. Toutanova, K., Chen, D., Pantel, P., et al.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015)
24. Trouillon, T., Welbl, J., Riedel, S., et al.: Complex embeddings for simple link prediction. In: Proceedings of the 33nd International Conference on Machine Learning (2016)
25. Wang, B., Shen, T., Long, G., Zhou, T., Wang, Y., Chang, Y.: Structure-augmented text representation learning for efficient knowledge graph completion. In: Proceedings of the Web Conference 2021 (2021)

26. Wang, J., Wang, Z., Zhang, D., et al.: Combining knowledge with deep convolutional neural networks for short text classification. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (2017)
27. Wang, X., He, X., Cao, Y., et al.: KGAT: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference (2019)
28. Wang, X., et al.: Kepler: a unified model for knowledge embedding and pre-trained language representation. Trans. Assoc. Comput. Linguist. **9**, 176–194 (2021)
29. Wang, Y., Xiao, W.D., Tan, Z., Zhao, X.: Caps-OWKG: a capsule network model for open-world knowledge graph. Int. J. Mach. Learn. Cybern. **12**, 1627–1637 (2021)
30. Wei, Z., Su, J., Wang, Y., Tian, Y., et al.: A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
31. Xie, R., Liu, Z., Jia, J., et al.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (2016)
32. Xu, X., Zhang, P., He, Y., Chao, C., Yan, C.: Subgraph neighboring relations infomax for inductive link prediction on knowledge graphs. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (2022)
33. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (2016)
34. Yang, B., Yih, W., He, X., et al.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the 3rd International Conference on Learning Representations (2015)
35. Yang, F., Yang, Z., Cohen, W.W.: Differentiable learning of logical rules for knowledge base reasoning. In: Proceedings of the 31th Neural Information Processing Systems (2017)
36. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (2021)
37. Zhou, Y., Shi, S., Huang, H.: Weighted aggregator for the open-world knowledge graph completion. In: Proceedings of the 6th International Conference of Pioneering Computer Scientists (2020)