



Duet Representation Learning with Entity Multi-attribute Information in Knowledge Graphs

Yuanbo Xu¹, Yuanbo Zhang¹, Yongjian Yang¹, Hangtong Xu^{1(✉)}, and Lin Yue²

¹ MIC Lab, College of Computer Science and Technology, Jilin University, Changchun, China
{yuanbox, yyj}@jlu.edu.cn, {zhangyb19, xuht21}@mails.jlu.edu.cn

² The University of Newcastle, Callaghan, Australia
Lin.Yue@newcastle.edu.au

Abstract. Representation Learning (RL) of knowledge graphs aims to project both entities and relations into a continuous low-dimension space. Most methods concentrate on learning entities' representations with structure information indicating the relations between entities (Trans- methods), while the utilization of entity multi-attribute information is insufficient for some scenarios, such as cold start issues or zero-shot problems. How to utilize the complex and diverse multi-attribute information for RL is still a challenging problem for enhancing knowledge graph embedding research. In this paper, we propose a novel RL model *Duet Entity Representation Learning* (DERL) for knowledge graphs, which takes advantage of entity multi-attribute information. Specifically, we devise a novel encoder *Entity Attribute Encoder* (EAE), which encodes both entity attribute types and values to generate the entities' attribute-based representations. We further learn the entities' representations with both structure information and multi-attribute information in DERL. We evaluate our method on two tasks: the knowledge graph completion task and the zero-shot task. Experimental results on real-world datasets show that our method outperforms other baselines on two downstream tasks by building effective representations for entities from their multi-attribute information. The source code of this paper can be obtained from <https://anonymous.4open.science/r/DUET-adma2023/>.

Keywords: Multi-attribute · Representation Learning · Knowledge graphs

1 Introduction

Knowledge graphs (KGs) provide a massive amount of structure information for entities and relations, which have been successfully utilized in various fields such as knowledge inference [21] and question answering [23]. Typical KGs like Freebase [1], or YAGO3 [8] usually model the multi-relational information with many structure triples represented as (*head entity*, relation, *tail entity*), which is also abridged as (*h, r, t*).

Currently, most RL methods focus on structure information but ignore attribute information in KGs. For example, in Fig. 1, we show two entity multi-attribute information in a structure triple sampled from DWY100K [13]. Although some works have realized the importance of multi-attribute information such as DT-GCN [11], they haven't

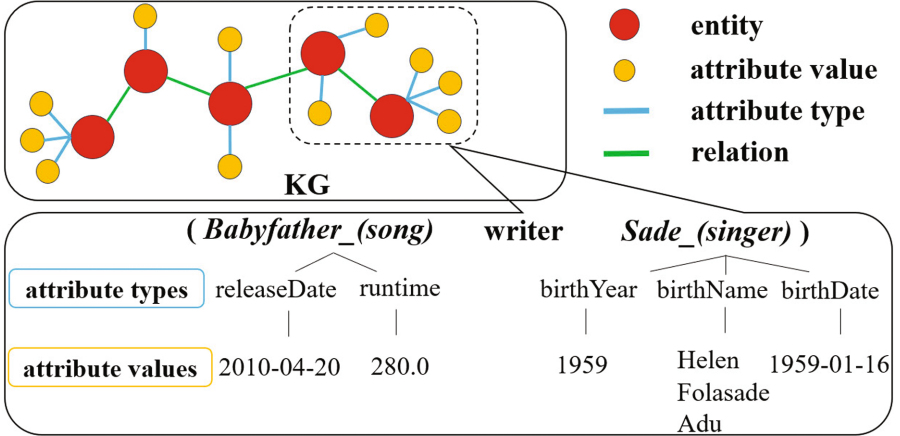


Fig. 1. Example of entity multi-attribute information in DWY100K.

fully used the rich semantic information. First, they didn't embed the attribute types and attribute values jointly and applied them to improve the entity representation semantic accuracy directly. Second, they didn't consider using both structure information and multi-attribute information to improve the overall RL effect. In addition, entity multi-attribute information is generally stored in KGs in the form of triples and the attribute triples can not tackle successive attribute values and might suffer from issues like one-to-multi or multi-to-one relations in KGs. Furthermore, entity multi-attribute information is usually diverse and complex: different entities may have multiple attribute types in KGs, and even different attribute values may have various data structures and value granularities. For example, in Fig. 1, *Babyfather_(song)* and *Sade_(singer)* have different attribute types; the three attribute types of *Sade_(singer)* that correspond to attribute values have different forms and structures. Entity multi-attribute information is too complex to use for learning embeddings directly. In the meanwhile, intuitively, different attribute types and values play different degrees of importance in the entities' representations. If entity multi-attribute information cannot be used reasonably, the entities' representations will lose a large amount of accurate semantic information thus reducing their semantic accuracy.

To address those problems, we first design a novel encoder EAE, which can encode the complex and diverse entity multi-attribute information to generate the entities' attribute-based representations. Moreover, we propose a novel RL model DERL for KGs, combining structure and multi-attribute information to improve KG embedding. In the DERL model, an entity's representation is responsible for jointly modeling the corresponding structure information and multi-attribute information.

For learning structure information, we follow a typical RL method TransE [2] and regard the relation in each structure triple as a translation from the head entity to the tail entity. For learning multi-attribute information, we use the EAE to learn the entities' attribute-based representations. In our EAE, we set up a training model that contains two embedding components. One component embeds the entity's different attribute

types, and the other component uses the bi-directional Long Short-Term Memory (Bi-LSTM) to characterize the attribute values. Attribute types and values apply an attention mechanism to learn their different importance for entities’ representations individually. Finally, we use the embeddings of these two parts to generate the entities’ attribute-based representations.

We evaluate our model on the knowledge graph completion task and the zero-shot task. Experimental results demonstrate that our model achieves state-of-the-art performances on both tasks. Our experimental results indicate that our model can use entity multi-attribute information to improve the overall KG embedding effect and verify the importance and necessity of attribute information for entity representation. We demonstrate the main contributions of this work as follows:

- We design a novel encoder *Entity Attribute Encoder* (EAE), which uses both the entity’s attribute types and values to generate the entity’s attribute-based representation. We adopt the attention mechanism for attribute types and values to distinguish the importance of different attribute information to the entity’s representation.
- We propose a novel RL model *Duet Entity Representation Learning* (DERL), which utilizes both entity structure information and multi-attribute information for enhancing RL’s effect.
- We evaluate the DERL model’s effectiveness on the knowledge graph completion and zero-shot tasks. Experimental results on real-world datasets illustrate that the DERL model consistently outperforms other baselines on these two tasks. To the best of our knowledge, this is the first work attempt to use entity multi-attribute information to solve the zero-shot problem.

2 Problem Formulation

We first introduce the symbols used in this paper. Given a structural triple $(h, r, t) \in T$, while $h, t \in E$ stand for entities, $r \in R$ stands for the relation. Respectively, h and t are the head entity and the tail entity. $a \in A$ stands for the attribute type and $v \in V$ stands for the attribute value. $c \in v$ stands for the attribute value character. T stands for the whole training set of structural triples. E is the set of entities, R is the set of relations, A is the set of attribute types, and V is the set of attribute values. We propose two kinds of representations for each entity to utilize structure information and multi-attribute information in DERL.

Definition 1. Structure-Based Representations: e_s represents the entity’s structure-based representation. e_{sh} and e_{st} are the structure-based representations based on the head entity and the tail entity. \mathbf{r} represents the relation’s representation. These representations could be learned through existing translation-based models.

Definition 2. Attribute-Based Representations: e_a represents the entity’s attribute-based representation. e_{ah} and e_{at} are the attribute-based representations based on the head entity and the tail entity. We will propose an encoder to construct this kind of representation in the following section.

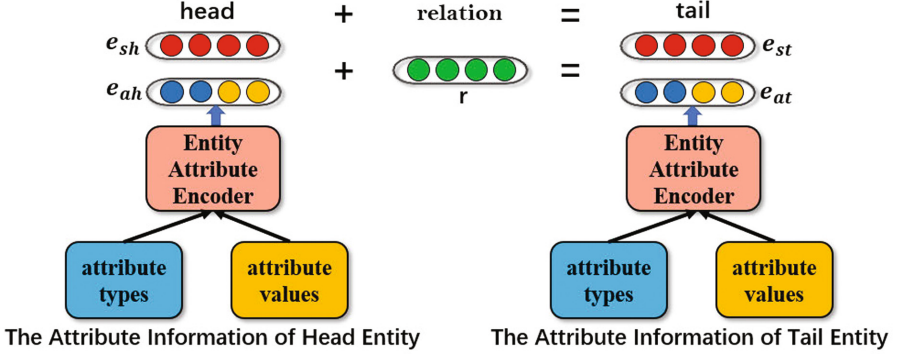


Fig. 2. The Overall Architecture of DERL Model

3 Methodology

3.1 Overall Architecture

We attempt to utilize entity structure information as well as multi-attribute information in the DERL model. Following the framework of translation-based methods, we define the overall energy function as follows:

$$S(h, r, t) = S_s + S_a, \quad (1)$$

where $S_s = \|e_{sh} + \mathbf{r} - e_{st}\|$. S_s is an energy function based on structure-based representations, which is the same as the translation-based methods. S_a is an energy function based on attribute-based representations and structure-based representations. To make the learning process of S_a compatible with S_s . We define S_a as:

$$S_a = S_{as} + S_{sa} + S_{aa}, \quad (2)$$

where $S_{as} = \|e_{ah} + \mathbf{r} - e_{st}\|$ and $S_{sa} = \|e_{sh} + \mathbf{r} - e_{at}\|$, in which one of the head entity or the tail entity is the structure-based representation, and the other is the attribute-based representation. $S_{aa} = \|e_{ah} + \mathbf{r} - e_{at}\|$, the head entity and the tail entity are both attribute-based representations. According to the overall energy function, the overall architecture of the DERL is demonstrated in Fig. 2. We learn the entities' structure-based representations and relations' representations from TransE. And we learn the entities' attribute-based representations from EAE. Under the overall energy function, we can get the attribute-based representations and the structure-based representations simultaneously. The overall energy function will project these two types of entities' representations into the same vector space with relation representations shared by all four energy functions, which will be promoted between two types of representations.

3.2 Entity Attribute Encoder

Entity multi-attribute information is difficult to use due to its complexity, heterogeneity, and different levels of importance. These problems directly lead to the difficulty

of learning multi-attribute information into entities’ representations. Therefore in our EAE, we consider the encoding of attribute types and attribute values respectively. The framework of the EAE is demonstrated in Fig. 3. The *Attribute Type Embedding (ATE)* learns the entity attribute type embeddings, and the *Attribute Value Character Embedding (AVCE)* learns the entity attribute value character embeddings. In the *Attribute Value Embedding (AVE)*, we use the Bi-LSTM to capture the attribute value characters’ information and generate the attribute value embeddings. We apply the attention mechanism to combine the attribute types and values for enhancing KG embedding. Finally, we combine the attribute type embeddings and the attribute value embeddings to generate the entities’ attribute-based representations.

Attribute Type Embedding (ATE). We first count the attribute types and randomly generate an embedding for each attribute type. Because each entity has a different number of attribute types and values, we adopt the zero-filling strategy to unify the numbers. To prevent the zero-filling strategy from affecting the model’s training, we separately generate the same embedding for all zeros to prevent problems such as vanishing gradients. Given the entity’s M attribute types: $A = (a_0, a_1, \dots, a_M)$, we obtain the following embeddings of the entity’s M attribute types:

$$\mathbf{A} = (\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_M). \quad (3)$$

Attribute Value Character Embedding (AVCE). We first count the characters that appear in the attribute values. Then we randomly generate an embedding for each attribute value character. Because the numbers of characters in each attribute value are different, we also utilize the zero-filling strategy and generate the same embedding for all zeros. Given the attribute value N characters: $v_i = (c_0, c_1, \dots, c_N)$, we get the following attribute value character embeddings:

$$\mathbf{v}_i = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_N). \quad (4)$$

Attribute Value Embedding (AVE). We observe that the different attribute values might appear differently in KGs. For example: “2012-12-12” and “180 cm” represent a person’s birthday and height respectively. In mono-lingual KGs, the attribute value can be considered as a sequence of characters with the same vocabulary. [15] proves that the LSTM can effectively capture the sequence information between characters. Therefore we choose the Bi-LSTM to learn the sequence information between characters from beginning to end. The following equations define the Bi-LSTM cell:

$$f_t = \sigma(\mathbf{W}_f[h_{t-1}, \mathbf{c}_t] + \mathbf{b}_f), \quad (5)$$

$$i_t = \sigma(\mathbf{W}_i[h_{t-1}, \mathbf{c}_t] + \mathbf{b}_i), \quad (6)$$

$$\tilde{H}_t = \tanh(\mathbf{W}_H[h_{t-1}, \mathbf{c}_t] + \mathbf{b}_H), \quad (7)$$

$$H_t = f_t \odot H_{t-1} + i_t \odot \tilde{H}_t, \quad (8)$$

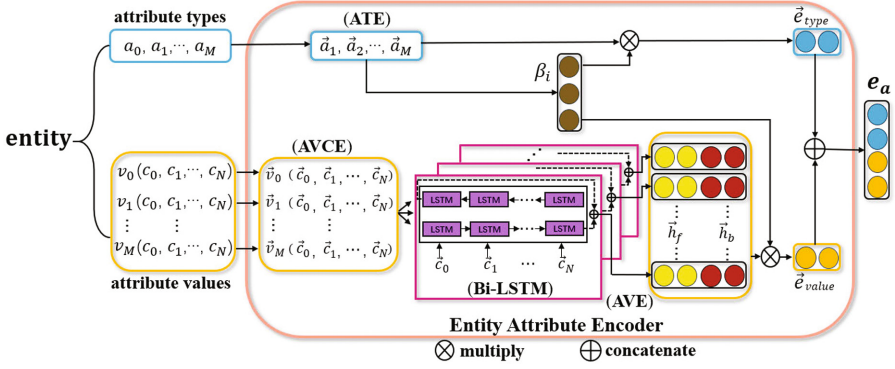


Fig. 3. The Framework of Entity Attribute Encoder

$$o_t = \sigma(\mathbf{W}_o[h_{t-1}, c_t] + \mathbf{b}_o), \quad (9)$$

$$h_t = o_t \odot \tanh(H_t), \quad (10)$$

where \odot denotes a vector multiplication, f_t , i_t , o_t are the forget gate, input gate, and out gate of the Bi-LSTM cells. \mathbf{W}_f , \mathbf{W}_i , \mathbf{W}_H , \mathbf{W}_o are weight matrices. σ is the sigmoid function. \mathbf{b}_f , \mathbf{b}_i , \mathbf{b}_H , \mathbf{b}_o are biases. Bi-LSTM is divided into the forward LSTM (F-LSTM) and the backward LSTM (B-LSTM). The F-LSTM reads the input character embeddings. For example, the F-LSTM reads the attribute value character embeddings $\mathbf{v}_i = (c_0, c_1, \dots, c_N)$ from left to right. The B-LSTM reads the attribute value character embeddings reversely. The outputs of the F-LSTM and B-LSTM are:

$$\mathbf{h}_f = \mathbf{F-LSTM}(c_N, \mathbf{h}_{f-1}), \quad (11)$$

$$\mathbf{h}_b = \mathbf{B-LSTM}(c_0, \mathbf{h}_{b+1}). \quad (12)$$

The initial states of the Bi-LSTM are set to zero vectors. After reading the embedding of all characters contained in an attribute value, we concatenate the final hidden states of the two-direction LSTM outputs to generate the attribute value embedding:

$$\mathbf{v}_i = [\mathbf{h}_f; \mathbf{h}_b]. \quad (13)$$

Given the entity's M attribute values: $V = (v_0, v_1, \dots, v_M)$, we get the following the attribute value embeddings:

$$\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_M). \quad (14)$$

Attention for Attribute Types and Attribute Values. An entity's attribute-based representation assembles all the entity attribute information, but not all attribute information is equally important to an entity's representation. To learn the importance of different attribute types and attribute values for an entity's representation, we adopt the

attention mechanism to solve this problem [22]. Given the entity attribute type embeddings: $\mathbf{A} = (\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_M)$, we calculate their attention weights:

$$\beta_i = \text{softmax}(\mathbf{A}^T \mathbf{W}_t \mathbf{a}_i), \quad (15)$$

where \mathbf{W}_t is the weight matrix of \mathbf{a}_i . Here we utilize the attribute type embeddings to get the attention weights. The attention weights of attribute value should be consistent with that of its attribute type:

$$\mathbf{e}_{type} = \sum_{i=0}^M \beta_i \mathbf{a}_i, \quad (16)$$

$$\mathbf{e}_{value} = \sum_{i=0}^M \beta_i \mathbf{v}_i, \quad (17)$$

we concatenate \mathbf{e}_{type} and \mathbf{e}_{value} to get the entity's attribute-based representation:

$$\mathbf{e}_a = [\mathbf{e}_{type}; \mathbf{e}_{value}]. \quad (18)$$

3.3 Objective Formalization

We utilize a margin-based score function as our training objective, which is defined as follows:

$$L = \sum_{(h,r,t) \in T} \sum_{(h',r',t') \in T'} \max(\gamma + S(h, r, t) - S(h', r', t'), 0), \quad (19)$$

where margin γ means the artificially defined minimum distance between positive and negative examples. $S(h, r, t)$ is the overall energy function, in which both head and tail entities have two kinds of representations: structure-based representations and attribute-based representations. The above energy functions are defined as the $L1$ -norm. It is verified by experiments that the DERL's effects based on $L1$ -norm are better than the DERL's effects based on $L2$ -norm. T' is the negative sample set of T , which we define as follows:

$$T' = (h', r, t) | h' \in E \cup (h, r', t) | r' \in R \cup (h, r, t') | t' \in E, \quad (20)$$

which means one of the entities or relations in a triple can be randomly replaced by another one. Since we have two entities' representations, if a triple already exists T , it will not treat it as a negative sample because the entity can be either a structure-based representation or an attribute-based representation.

3.4 Optimization and Implementation Details

DERL model can be defined as a parameter set $\theta = (\mathbf{E}, \mathbf{R}, \mathbf{A}, \mathbf{C}, \mathbf{W}, \mathbf{B})$. \mathbf{E} stands for the embedding set of entities and \mathbf{R} stands for the embedding set of relations. They can be randomly initialized or trained by previous translation-based methods such as TransH

[18] and TransR [7]. \mathbf{A} stands for the embedding set of attribute types and \mathbf{C} stands for the embedding set of attribute value characters and they are initialized randomly. \mathbf{W} and \mathbf{B} represent the weight set and bias set of Bi-LSTM and attention mechanism in EAE, which can be initialized randomly. We utilize the mini-batch stochastic gradient descent (SGD) to optimize our model, where chain rules are applied to update the variables and parameters. We use GPU to accelerate training.

4 Experiments

4.1 Datasets and Experiment Settings

Datasets. In our experiments, we use the DWY100K [13] to evaluate our models’ knowledge graph completion effect. For the zero-shot task, we build a new dataset FB24K-New based on FB24K [6] to simulate a zero-shot scenario. We select 12,789 entities as In-KG entities in FB24K and select 5,179 entities in FB24K that are related to In-KG entities as Out-of-KG entities. We extract the structure triples which contain In-KG entities and Out-of-KG entities and add them to the test set. Our test set is split into 4 types: (I - I), (O - I), (I - O), and (O - O). I represent an In-KG entity, and O represents an Out-of-KG entity. The DWY100K, FB24K, and FB24K-New details are listed in Table 1 and Table 2.

Table 1. Statistics of DWY100K

Datasets	#Ent	#Rel	#Attr	#Attr tr	#Rel tr
DBP-WD-Dbpedia	100,000	330	351	381,166	463,294
DBP-WD-Wikidata	100,000	220	729	789,815	448,774
DBP-YG-Dbpedia	100,000	302	334	451,646	428,952
DBP-YG-Wikidata	100,000	31	23	118,373	502,563

Experiment Settings. In the DERL model, the margin γ set among $\{1.0, 2.0, 3.0\}$. The learning rate λ set among $\{0.0005, 0.0003, 0.001\}$. We set different learning rates for different representation type combinations. The optimal configurations of the DERL are: $\lambda = 0.001$, $\gamma = 1.0$. We set the size of character embedding and attribute type embedding to 32. We set the attention weight size to 64 and the size of the hidden layer of Bi-LSTMs to 16. The dimensions of the attribute-based representation and structure-based representation are set to 64. The dimension of the relation’s representation is set to 64. We set two evaluation settings named “Raw” and “Filter”: “Filter” drops the repeated triples in the training stage (when we alternate the entities and relations, the reconstructed triple has a chance to be an existing triple), while “Raw” does not.

Table 2. Statistics of FB24K and FB24K-New

Dataset	#Ent	#Rel	#Attr	#Attr tr	#Rel tr
FB24K	23,634	673	314	207,610	216,409
Dataset	#Ent	#I - I	#O - I	#I - O	#O - O
FB24K-New	17,968	100,249	12,699	400	135

4.2 Knowledge Graph Completion Task

Due to the incompleteness and complexity of KGs, many KGs are missing triples, and a large number of potential relations between entities in the KGs are not discovered. Knowledge graph completion aims to learn appropriate entities’ and relations’ representations to discover the latent, correct triples. In addition, the knowledge graph completion task has been widely used to evaluate the quality of knowledge representations [24].

Evaluation Protocol. We will report four prediction results based on our models. The DERL(Structure) only utilizes structure-based representations for all entities when predicting the missing ones. While DERL(Attribute) only utilizes attribute-based representations. The DERL(Union) is a simple joint method considering the weighted concatenation of both entities’ representations. The DERL(Ablation) only uses attribute information for training. We use three measures as our evaluation indicators: Mean Rank, Hits@10 and Hits@1 [19,20]. In our experiment, we select TransE [2], ComplEx [16], Simple [5], RotatE [14], QuatRE [9], ParamE [3], TransRHS [24], DT-GCN [11], and HittER [4] as baselines, which will be discussed in the Related Work.

Experimental Results. Table 3 and Table 4 present the entity and relation prediction results respectively. Our analysis draws the following conclusions: (1) most DERL models outperform all baselines on both Mean Rank, Hits@10, and Hit@1. It indicates that the entities’ representations with multi-attribute information perform better in knowledge graph completion, which not only proves that EAE can effectively encode attribute information but also shows that DERL model can learn an accurate entity’s representation. (2) DERL(Structure) shows good performance, although it is inferior to some experimental results. After the mutual promotion of two kinds of information, compared with some models (such as TransE, ComplEx, DT-GCN) performance effects have been improved. The results indicate that two entities’ representations can learn and share the same vector space. This proves that two kinds of information can be jointly trained to improve the RL’s overall effect. (3) The DERL models’ results outperform baselines on Mean Rank. The Mean Rank can well reflect the overall quality of knowledge representation and determine the prediction results. In this paper, we use entity multi-attribute information as semantics information to improve the entity representation semantic precision. Therefore, the DERL models’ results are much better than the baselines’ results on Mean Rank. The case studies indicate that we may not know the

Table 3. Entity Prediction Results in Knowledge Graph Completion Task

Model	DBP-WD-Dbpedia				DBP-WD-Wikidata				DBP-YG-Dbpedia				DBP-YG-Wikidata			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	343	242	30.1	41.9	341	254	30.1	41.4	354	258	28.9	40.3	445	379	14.4	27.8
ComplEx	341	240	30.3	41.7	338	259	30.3	42.3	349	261	29.3	39.6	443	381	13.9	27.6
Simple	332	251	29.8	40.5	356	287	28.4	40.3	402	301	26.4	39.1	487	392	11.9	25.6
RotatE	314	<u>229</u>	33.4	45.2	<u>322</u>	<u>244</u>	<u>31.8</u>	46.9	<u>341</u>	<u>249</u>	<u>30.9</u>	45.2	432	377	14.6	30.1
QuatRE	322	261	27.3	38.1	354	287	26.7	38.5	371	270	24.3	36.5	447	379	<u>14.8</u>	<u>30.2</u>
ParamE	437	311	22.3	31.1	367	298	22.7	32.4	381	331	20.3	30.5	533	401	10.7	21.2
TransRHS	<u>310</u>	237	<u>33.5</u>	<u>45.9</u>	327	245	31.2	<u>47.2</u>	345	251	29.9	44.2	431	379	14.1	29.7
DT-GCN	351	245	32.1	33.2	519	388	31.7	33.3	354	287	30.3	45.3	455	367	14.0	22.6
HittER	401	302	27.8	44.7	444	363	30.7	45.1	368	311	26.4	<u>46.9</u>	<u>384</u>	<u>357</u>	13.8	28.6
DERL(Ablation)	310	241	33.9	49.8	320	251	33.2	44.1	345	251	29.9	44.2	422	370	14.5	29.5
DERL (Structure)	330	245	31.9	41.4	341	251	30.5	43.1	342	249	29.5	43.7	443	382	14.3	27.1
DERL (Attribute)	311	236	34.9	50.8	317	239	34.3	47.6	321	247	34.1	46.8	429	371	14.9	30.5
DERL (Union)	307	221	35.1	51.7	318	245	34.1	50.5	312	240	34.4	47.1	381	352	14.8	30.1
Improv.	1.0%	3.5%	4.8%	12.6%	1.6%	2.1%	7.0%	7.8%	8.5%	3.6%	8.5%	0.4%	0.8%	1.4%	0.7%	1.0%

entities’ details only by using the structure information, but we may know the entity better by learning rich potential information from entity multi-attribute information.

Table 4. Relation Prediction Results in Knowledge Graph Completion Task

Model	DBP-WD-Dbpedia				DBP-WD-Wikidata				DBP-YG-Dbpedia				DBP-YG-Wikidata			
	Mean Rank		Hits@1(%)		Mean Rank		Hits@1(%)		Mean Rank		Hits@1(%)		Mean Rank		Hits@1(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	4.01	3.53	37.3	41.1	3.97	3.58	32.2	43.5	4.22	3.52	30.4	41.7	6.13	5.51	<u>18.8</u>	29.2
ComplEx	3.98	3.55	36.8	46.8	3.92	3.54	36.4	45.8	3.91	3.56	35.6	42.7	6.19	5.42	18.4	26.4
Simple	4.45	3.98	31.8	35.8	4.91	4.53	33.4	34.6	4.78	4.56	32.3	43.6	7.19	6.58	10.4	23.1
RotatE	<u>3.54</u>	<u>3.21</u>	41.4	50.8	<u>3.51</u>	<u>3.09</u>	<u>44.9</u>	54.1	<u>3.76</u>	<u>3.32</u>	<u>43.9</u>	50.1	6.02	5.52	18.6	33.4
QuatRE	4.21	3.93	38.6	47.2	3.87	3.51	41.4	50.5	4.02	3.88	36.6	44.3	7.14	6.11	16.9	30.9
ParamE	4.54	3.99	36.4	43.2	3.57	3.14	44.1	54.8	4.54	3.91	34.3	41.2	8.24	6.31	13.9	25.9
TransRHS	3.72	3.51	42.8	<u>51.4</u>	4.55	4.05	35.1	46.7	3.79	3.36	39.2	<u>51.3</u>	6.17	5.55	16.7	31.6
DT-GCN	5.28	5.05	21.3	29.4	6.91	5.53	23.4	34.1	6.98	6.56	20.1	23.4	7.48	6.52	11.7	17.8
HittER	4.07	3.43	<u>43.5</u>	50.3	5.48	4.56	39.6	<u>54.9</u>	5.48	5.01	37.7	49.8	<u>5.58</u>	<u>5.33</u>	14.5	<u>33.6</u>
DERL(Ablation)	3.54	3.31	41.8	50.4	3.61	3.19	44.1	50.7	3.51	2.99	47.2	54.6	5.82	5.46	18.2	30.9
DERL (Structure)	3.91	3.55	37.4	48.2	3.96	3.51	40.5	45.4	3.97	3.64	38.1	44.1	6.21	5.57	17.8	25.5
DERL (Attribute)	3.61	3.22	46.5	53.3	3.41	3.02	46.1	56.6	3.51	3.07	44.1	54.8	5.94	5.69	18.8	32.9
DERL (Union)	3.53	3.09	47.7	51.9	3.57	2.98	45.7	55.3	3.34	2.89	47.8	56.1	5.51	5.17	18.9	34.3
Improv.	0.3%	3.7%	9.7%	4.0%	2.8%	3.6%	2.7%	3.1%	11.2%	13.0%	8.9%	9.4%	1.3%	3.0%	0.5%	2.9%

4.3 Knowledge Graph Completion in Zero-Shot Task

How to embed the new entities in the KGs and apply them is the main purpose of the zero-shot task. However, it is difficult to embed the Out-of-KG entities directly, and efficiently finding the latent relations between Out-of-KG and In-KG entities is difficult.

In this paper, we use multi-attribute information to learn the Out-of-KG entities’ representations, which solves the problems that the Out-of-KG entities can’t embed directly and the knowledge graph completion in zero-shot tasks.

Evaluation Protocol. We select DKRL [19], ConMask [12], and OWE [10] as our baselines which will be discussed in the Related Work. We utilize Hits@10, and Hits@1 [19] for entity and relation prediction. We only present the results on the “Filter” setting. We present four results in the experiment, and the (O - I), (I - O), and (O - O) have been explained above; the Total is the combined result of these three test sets.

Experimental Results. Fig. 4 shows the experimental results of (O - I), (I - O), (O - O), and Total. We can observe that: (1) In most cases, DERL significantly outperforms other models on all four types of test sets. DERL achieves about 16.2% improvement in entity prediction and 5.7% improvement in relation prediction. It demonstrates that DERL can effectively utilize the Out-of-KG entity multi-attribute information into the entity’s representation to handle the zero-shot problem. (2) The entity description information and multi-attribute information belong to the text information of the entity, but the DERL model performs better in entity prediction, relation prediction, and Mean Rank. It not only shows the effectiveness of the DERL model in embedding text information and capturing entity semantic information but also explains the advantages of using entity attribute information to solve the zero-shot problem. (3) From Fig. 4, we can see that some DERL’s results are not ideal, which may be because two entities belong to two entity spaces. Therefore, the connections between In-KG and Out-of-KG entities are still in need of enhancement.

5 Related Work

5.1 Knowledge Graph Embedding

In recent years, knowledge graph embedding methods have achieved great success and promotion. TransE [2] follows the rule ($\mathbf{h} + \mathbf{r} \approx \mathbf{t}$) to embed the entities and relations. Simple [5] not only uses the Polyadia-Score but also utilizes the inverse of the relation. ParamE [3] extends current embedding methods by combining the nonlinear-fitting ability of neural networks and translational properties. ComplEx [16] first introduces the Complex-Spaces to capture symmetric and antisymmetric relations. RotatE [14] treats the relation as a rotation from the head entity to the tail entity. QuatRE [9] defines the Quaternion-Space with Hamilton-Product to enhance correlations between head and tail entities. TransRHS [24] utilizes the relative positions between vectors and spheres to enhance the generalization between relations. HittER [4] proposes a Transformer-based RL model to enhance the effects of entities and relations. DT-GCN [11] makes full use of the advantages of multiple-types entity’s attribute values to explore the expressiveness of the entity’s representation.

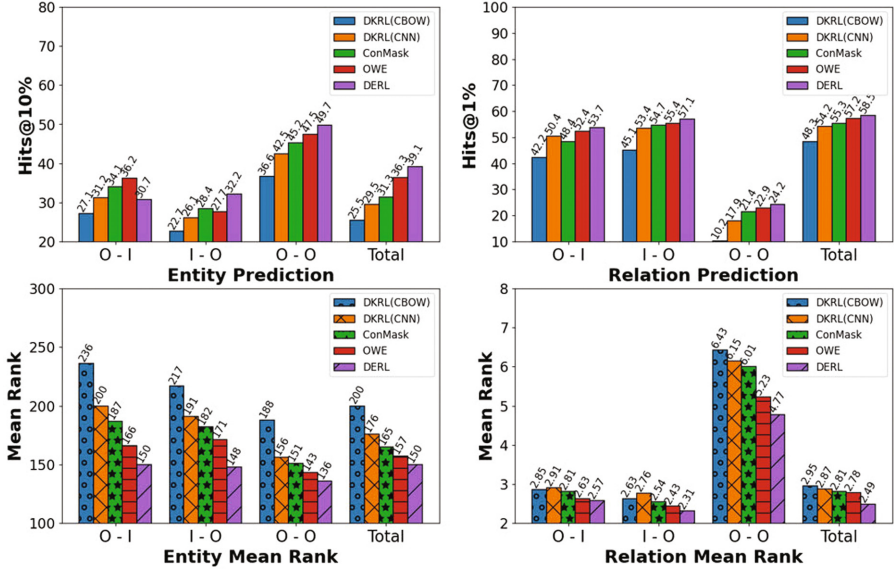


Fig. 4. Entity and Relation Prediction Results in Zero-Shot Task

5.2 Zero-Shot Problem

Zero-shot problem is a key issue in Knowledge Graph Completion because of the data sparsity (including entity and relations). Currently, few models are in a position to solve the zero-shot problem by using ancillary information. DKRL [19] proposes to use entity description information to generate entities' representations to solve the zero-shot problem. ConMask [12] comprehensively utilizes entities' names and textual information to deal with zero-shot situations. OWE [10] combines the entities' names and description information in the Transformation Space to improve open-world link prediction. To benefit the zero-shot problem in KGs, we utilize ancillary information directly to learn attribute-based representation and structure-based representation jointly, thus enriching the sparse information hidden in knowledge graphs.

6 Conclusion

In this paper, we propose a novel RL model (DERL) that utilizes both structure and multi-attribute information to improve the RL's effect in KGs. To effectively encode entity multi-attribute information, we also design an attribute information encoder EAE. Experimental results on real-world datasets demonstrate that the DERL model consistently outperforms other baselines on the knowledge graph completion task and zero-shot task. [17]

References

1. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of ICMD, pp. 1247–1250 (2008)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NIPS, pp. 2787–2795 (2013)
3. Che, F., Zhang, D., Tao, J., Niu, M., Zhao, B.: Parame: regarding neural network parameters as relation embeddings for knowledge graph completion. In: Proceedings of AAAI, pp. 2774–2781 (2020)
4. Chen, S., Liu, X., Gao, J., Jiao, J., Zhang, R., Ji, Y.: Hitter: Hierarchical transformers for knowledge graph embeddings. In: Proceedings of EMNLP, pp. 10395–10407 (2021)
5. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs, pp. 4289–4300 (2018)
6. Lin, Y., Liu, Z., Sun, M.: Knowledge representation learning with entities, attributes and relations. In: Proceedings of IJCAI, pp. 2866–2872 (2016)
7. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of AAAI, pp. 2181–2187 (2015)
8. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: a knowledge base from multilingual wikipedias. In: Proceedings of IDSR (2015)
9. Nguyen, D.Q., Vu, T., Nguyen, T.D., Phung, D.: Quatre: relation-aware quaternions for knowledge graph embeddings. CoRR abs/ [arXiv: 2009.12517](https://arxiv.org/abs/2009.12517) (2020)
10. Shah, H., Villmow, J., Ulges, A., Schwanecke, U., Shafait, F.: An open-world extension to knowledge graph completion models. In: Proceedings of the AAAI, pp. 3044–3051 (2019)
11. Shen, Y., Li, Z., Wang, X., Li, J., Zhang, X.: Datatype-aware knowledge graph representation learning in hyperbolic space. In: Proceedings of CIKM, pp. 1630–1639 (2021)
12. Shi, B., Wenginger, T.: Open-world knowledge graph completion. CoRR abs/ [arXiv: 1711.03438](https://arxiv.org/abs/1711.03438) (2017)
13. Sun, Z., Hu, W., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: Proceedings of IJCAI, pp. 4396–4402 (2018)
14. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: Proceedings of ICLR (2019)
15. Trisedya, B.D., Qi, J., Zhang, R.: Entity alignment between knowledge graphs using attribute embeddings. In: Proceedings of AAAI, pp. 297–304 (2019)
16. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML, pp. 2071–2080 (2016)
17. Wang, S., et al.: Knowledge graph representation via hierarchical hyperbolic neural graph embedding. In: 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021, pp. 540–549. IEEE (2021). <https://doi.org/10.1109/BigData52589.2021.9671651>
18. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI, pp. 1112–1119 (2014)
19. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of AAAI, pp. 2659–2665 (2016)
20. Xie, R., Liu, Z., Luan, H., Sun, M.: Image-embodied knowledge representation learning. In: Proceedings of IJCAI, pp. 3140–3146 (2017)
21. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Bengio, Y., LeCun, Y. (eds.) In Proceedings of ICLR (2015)

22. Yang, K., Liu, S., Zhao, J., Wang, Y., Xie, B.: COTSAE: co-training of structure and attribute embeddings for entity alignment. In: Proceedings of AAAI, pp. 3025–3032 (2020)
23. Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X.: Neural generative question answering. In: Proceedings of IJCAI, pp. 2972–2978 (2016)
24. Zhang, F., Wang, X., Li, Z., Li, J.: Transrhs: A representation learning method for knowledge graphs with relation hierarchical structure. In: Proceedings of IJCAI, pp. 2987–2993 (2020)