Knowledge Representation Learning with Entities, Attributes and Relations

Yankai Lin[1], **Zhiyuan Liu**[1*], **Maosong Sun**[1,2]
[1] Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China
[2] Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China

Abstract

Distributed knowledge representation (KR) encodes both entities and relations in a low-dimensional semantic space, which has significantly promoted the performance of relation extraction and knowledge reasoning. In many knowledge graphs (KG), some relations indicate attributes of entities (attributes) and others indicate relations between entities (relations). Existing KR models regard all relations equally, and usually suffer from poor accuracies when modeling one-to-many and many-to-one relations, mostly composed of attribute. In this paper, we distinguish existing KG-relations into attributes and relations, and propose a new KR model with entities, attributes and relations (KR-EAR). The experiment results show that, by special modeling of attribute, KR-EAR can significantly outperform state-of-the-art KR models in prediction of entities, attributes and relations. The source code of this paper can be obtained from `https://github.com/thunlp/KR-EAR`.

1 Introduction

People build large-scale knowledge graphs (KG), such as Freebase, DBpedia and YAGO, to store complex structured information about the facts of the real world. The facts in KGs are usually organized in the form of triplets, e.g., (*Washington*, `CapitalOf`, *USA*). KGs have been widely adopted in various applications such as question answering and Web search.

Existing KGs have already included thousands of relation types, millions of entities and billions of triplets. Nevertheless these KGs remain far from complete as compared to the amount of real-world facts. In order to further expand KGs, many researches have been devoted to automated fact exploration.

Recently, neural-based representation learning (RL) methods are proposed to encode the semantics of both entities and relations in low-dimensional semantic space (i.e., embeddings), which can be further employed to discover novel facts. As a simple and effective neural-based RL model, TransE [Bordes *et al.*, 2013] learns low-dimensional vectors for both entities and relations, and regards the relation in a triplet as a translation between the embeddings of the two entities, that is, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when the triple (h, r, t) holds. TransE achieves amazing performance for knowledge graph completion and relation extraction from text [Bordes *et al.*, 2013].

However, TransE encounters issues when modeling one-to-many and many-to-one relations. From many KGs we observe that, some relations indicate attributes of entities (the tail entity is usually abstraction, such as `Gender` and `Profession`), and others indicate relations between entities (the head and tail entities are both real world objects). Hence, existing KG-relations can be divided into **attributes** and **relations**.

Table 1: The relationships between some typical attributes and relations and their corresponding mapping properties.

Relation Type	Relation	E_t	E_h
Attributes	nationality	1.05	1,551.90
	gender	1.00	637,333.33
	ethnicity	1.12	41.52
	religion	1.09	107.40
Relations	parents	1.58	1.67
	capital	1.29	1.42
	author	1.02	2.17
	founder	1.37	1.31

In Table 1, we list some typical attributes and relations with their information including name, E_h, E_t (For each relation or attribute, we compute two statistics, the expectation number of tail entities per head entity and the expectation number of head entities per tail entity, denoted as E_t and E_h respectively). As demonstrated in the table, attributes are the primary source of one-to-many and many-to-one relations. For example, in the attribute `Gender`, the attribute property *Male* is regarded as an entity, which is connected with millions of human entities. For these relations, TransE and its extensions like TransH [Wang *et al.*, 2014b] and TransR [Lin *et al.*, 2015b] cannot sufficiently build translation between entities and their attribute properties.

Evidently, attributes and relations exhibit rather distinct characteristics: (1) For relations, both head and tail entities are usually from a large entity set. Each entity only builds a specific relation with a limited number of entities. (2) For at-

[*] Corresponding author: Zhiyuan Liu (liuzy@tsinghua.edu.cn).

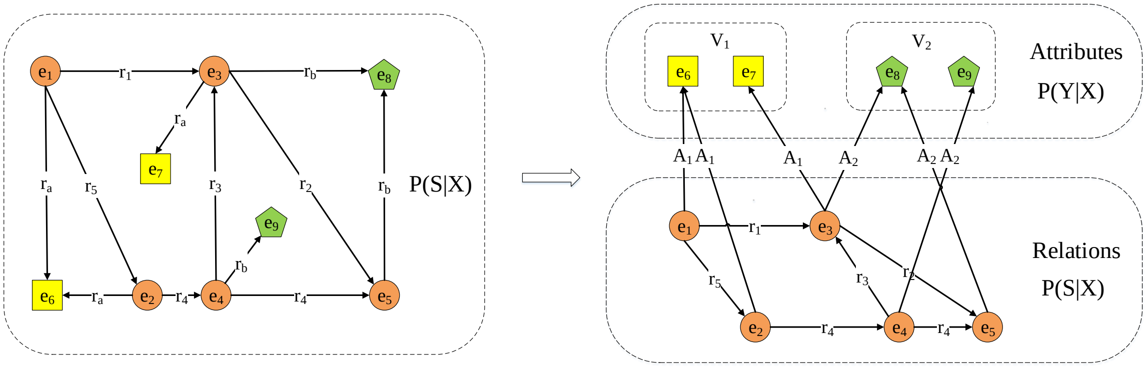

Figure 1: An illustration of KR-EAR and traditional KR representation models. A_1 and A_2 are two attributes. The value set of attribute A_1 (V_1) contains e_6, e_7 which are square (also colored with yellow), while A_2 (V_2) contains e_7, e_8 which are pentagon (also colored with green). In traditional KR representation model (left), attributes A_1 and A_2 are regarded as relations r_a and r_b. In contrast, KR-EAR uses traditional KR representation model to encode relational triples and regards attribute prediction as a classification problem.

tributes, the attribute properties are usually from a small set of entries. For example, the relation `Gender` typically has two properties $\{$*Female*, *Male*$\}$. Each property is always shared by many entities. Hence we should model attributes and relations in different ways.

In this paper, we present a new KR model with entities, attributes and relations (KR-EAR). In this framework, each entity has various attributes, and entities are connected with relations. Both entities and relations are represented with low-dimensional vectors (embeddings). The embeddings are learned by: (1) building translation between entities according to relations, as in TransE (also applies to TransH and TransR); and (2) inferring attribute properties based on entity embeddings.

We build a real-world dataset to evaluate the performance of KR-EAR on knowledge graph completion, including entity prediction, relation prediction and attribute prediction. The experiment results show that, by explicitly modeling of attributes, KR-EAR consistently and significantly outperforms state-of-the-art KR models in the three tasks.

2 Related Work

In recent years, many works have made great effort on modeling multi-relational data such as social networks and KGs. There are several approaches to model multi-relational data. Some works use latent representations of entities and relations to propagate information between triples and capture global dependencies in the data. In [Kemp *et al.*, 2006; Miller *et al.*, 2009; Sutskever *et al.*, 2009; Zhu, 2012], they use Bayesian clustering for link prediction. In [Singh and Gordon, 2008; Nickel *et al.*, 2011; 2012], they adopt the idea of matrix/tensor factorization. Moreover, neural based models[Bordes *et al.*, 2011; Chen *et al.*, 2013; Socher *et al.*, 2013; Bordes *et al.*, 2013; 2014] have attracted much attention. Among all of them, TransE [Bordes *et al.*, 2013] and its attention [Wang *et al.*, 2014b; Lin *et al.*, 2015b] achieve a good trade-off between prediction accuracy and computational efficiency by regarding relations as a translation between entities. We introduce TransE and its extensions TransH and TransR in detail.

TransE

For each triple (h, r, t), TransE [Bordes *et al.*, 2013] wants $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. This indicates that $\mathbf{t}$ should be the nearest entity from $(\mathbf{h} + \mathbf{r})$. Hence, TransE defines the following energy function

$$f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L1/L2} \qquad (1)$$

The function returns low score if (h, r, t) holds, vice versa.

TransH

TransH [Wang *et al.*, 2014b] enables an entity to have distinct embeddings when involved in different relations. For a relation r, TransH models the relation with a vector $\mathbf{r}$ and a hyperplane with $\mathbf{w}_r$ as the normal vector. Then the score function is defined as

$$f_r(h,t) = -\|\mathbf{h} - \mathbf{w}_r^{\mathrm{T}}\mathbf{h}\mathbf{w}_r + \mathbf{r} - (\mathbf{t} - \mathbf{w}_r^{\mathrm{T}}\mathbf{t}\mathbf{w}_r)\|_{L1/L2} \qquad (2)$$

TransR

TransR [Lin *et al.*, 2015b] models entities and relations in entity space and relation spaces, and performs translation in relation spaces. TransR sets a projection matrix $\mathbf{M}_r \in \mathbb{R}^{k \times d}$, which may projects entities from entity space to relation space. Via the mapping matrix, the energy function is correspondingly defined as

$$f_r(h,t) = \|\mathbf{h}\mathbf{M}_r + \mathbf{r} - \mathbf{t}\mathbf{M}_r\|_{L1/L2} \qquad (3)$$

We note that, these KR models regard all relations equally, and usually suffer from poor accuracies when modeling one-to-many and many-to-one relations mostly composed of attributes. In contrast, by splitting existing KG-relations into attributes and relations, we only use existing KR models to model relations between entities.

Attributes have already been considered in some previous works. [Nickel *et al.*, 2012] handles attributes by performing joint matrix factorization on an entity-attribute matrix and tensor factorization. [Suchanek *et al.*, 2011] filters unnecessary literals in when aligning KGs. We note that, these methods only regard the triples with literals as attributional triples. In fact, many one-to-many and many-to-one triples which consist of two entities can be an attributional triple, e.g., (*Steve Jobs*, `Gender`, *Male*). By separating the modeling of relational and attributional triples, KR-EAR can learn superior representations of entities, attributes and relations for knowledge graph completion as shown in our experiments.

3 Problem Formulation

We first introduce the notations used in this paper. Let $G = (E, S, Y)$ denotes a KG, where $E = \{e_1, e_2, \cdots, e_{|E|}\}$ is a set of $|E|$ entities, S is the relational triples and Y is the attributional triples.

Definition 1. Relational Triples: $S \subseteq E \times R \times E$ is a set of relational triples representing the relations between entities, where R is a set of $|R|$ relations.

Definition 2. Attributional Triples: $Y \subseteq E \times A \times V$ is a set of attributional triples representing the attributes of entities, where $A = \{A_1, A_2, ...A_{|A|}\}$ is a set of $|A|$ attributes, and each attribute $A_i \in A$ has a corresponding value set $V_i \in V$. For example, a person's gender may be *Male* or *Female*.

Given a knowledge graph G, our objective is to learn embeddings $\mathbf{X}$ of entities, relations and attributes, to predict relations between entities and to predict attributes of entities. The embeddings are set in $\mathbb{R}^k$ and we denote them with the same letters in boldface.

4 Knowledge Representation with Entities, Attributes and Relations

In this section, we introduce our KR-EAR model that learns knowledge representation with entities, attributes and relations.

4.1 Framework

Our goal is to design a unified model to represent not only relations between entities but also attributes of entities. We define an objective function by maximizing the joint probability of relational triples and attributional triples given the embeddings $\mathbf{X}$, i.e., $P(S, Y|\mathbf{X})$. It assumes that relational triples and attributional triples are conditionally independent, and formalize the objective function as:

$$\begin{aligned} &P(S, Y|\mathbf{X}) = P(S|\mathbf{X})P(Y|\mathbf{X}) \\ = &\prod_{(h,r,t)\in S} P((h,r,t)|\mathbf{X}) \prod_{(e,a,v)\in Y} P((e,a,v)|\mathbf{X}), \end{aligned} \tag{4}$$

where $P((h, r, t)|\mathbf{X})$ denotes the conditional probability of relational triples (h, r, t) and $P((e, a, v)|\mathbf{X})$ denotes the conditional probability of attributional triple (e, a, v). Hence, our model KR-EAR consists of two core components: (1) **Relational Triple Encoder** embeds the correlations between entities and relations; (2) **Attributional Triple Encoder** embeds the correlations between entities and attributes, acting as a classification model for attribution prediction.

Figure 1 shows an illustration of our proposed model KR-EAR as compared to existing KR models. In a traditional KR model (left), attributional triples are encoded in the same way as relational triples. In contrast, KR-EAR regards attribute prediction as a classification task, as well as still using traditional KR models to encode relational triples. We introduce the two components in detail as follows.

4.2 Relational Triple Encoder (RTE)

In RTE, we aim to embed entities and relations to capture the correlations between them. When learning from relational triples, we usually optimize the conditional probability $P(h|r, t, \mathbf{X})$, $P(t|h, r, \mathbf{X})$ and $P(r|h, t, \mathbf{X})$ instead of $P((h, r, t)|\mathbf{X})$.

In this paper, without less of generality, we adopt TransE and TransR to encode relational triples. In this framework, take the conditional probability $P(h|r, t, \mathbf{X})$ for example, it is formalized as follows:

$$P(h|r, t, \mathbf{X}) = \frac{\exp(g(\mathbf{h}, \mathbf{r}, \mathbf{t}))}{\sum_{\hat{h}\in E}\exp(g(\hat{\mathbf{h}}, \mathbf{r}, \mathbf{t}))} \tag{5}$$

where $g()$ is the energy function which indicates the correlation of relation r and entity pair (h, t). Here, we can follow TransE to define the energy function $g(h, r, t)$ as:

$$g(\mathbf{h}, \mathbf{r}, \mathbf{t}) = -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{L1/L2} + b_1, \tag{6}$$

or follow TransR to define it as:

$$g(\mathbf{h}, \mathbf{r}, \mathbf{t}) = -||\mathbf{h}\mathbf{M_r} + \mathbf{r} - \mathbf{t}\mathbf{M_r}||_{L1/L2} + b_1, \tag{7}$$

where b_1 is a bias constant, $\mathbf{M_r}$ is the projection matrix described in Eq. (3). We name the two versions as KR-EAR(TransE) and KR-EAR(TransR) separately. Note that, the effectiveness of formalizing TransE/TransR in a probability fashion has been verified in [Wang *et al.*, 2014a].

In fact, other KR models such as TransD [Ji *et al.*, 2015], TranSparse [Ji *et al.*, 2016], KG2E [He *et al.*, 2015], PTransE [Lin *et al.*, 2015a] can also be easily adopted as relational triple encoder. Due to the space limit, in this paper we only explore the effectiveness of TransE and TransR.

4.3 Attributional Triple Encoder (ATE)

It is intuitive that, the correlations between entities and their attributes should be captured with a classification model. Hence, in ATE we consider the conditional probability $P(v|a, r, \mathbf{X})$ for each triple (e, a, v), and formalize the conditional probability as follows:

$$P(v|e, a, \mathbf{X}) = \frac{\exp(h(\mathbf{e}, \mathbf{a}, \mathbf{v}))}{\sum_{\hat{v}\in V_a}\exp(h(\mathbf{e}, \mathbf{a}, \hat{\mathbf{v}}))} \tag{8}$$

where $h()$ is the scoring function for each attribute value of a given entity.

The function $h()$ is defined as follows. We first transform entity embeddings into the attribute space via a single-layer neural network, and then calculate the semantic similarity between the transformed embedding and the embedding of the corresponding attribute value:

$$h(\mathbf{e}, \mathbf{a}, \mathbf{v}) = -||f(\mathbf{e}\mathbf{W_a} + \mathbf{b_a}) - \mathbf{V_{av}}||_{L1/L2} + b_2, \tag{9}$$

where $f()$ is a nonlinear function such as $\tanh$, $\mathbf{V_{av}}$ is the embedding of attribute value v and b_2 is a bias constant.

Attribute Correlations

It is well-acknowledged that, various attributes of an entity usually have strong correlations. For example, a person having British nationality is more probable to speak English. The division of attributes and relations in the KR-EAR model enables the model to consider the correlations between attributes for inference.

Formally, for an entity e and its attributional triple (e, a, v), suppose $Y(e) = \{(e, \hat{a}, \hat{v})|(e, \hat{a}, \hat{v}) \in Y\}$ are other attributes of e except (e, a, v) and are already known. The conditional probability of (e, a, v) is formalized as follows:

$$P((e, a, v)|\mathbf{X}) \propto P(v|e, a, \mathbf{X})P((e, a, v)|Y(e)) \quad (10)$$

Here $P((e, a, v)|Y(e))$ indicates the probability of the attributional triple (e, a, v) when given other attributes of entity e, which is defined as a softmax function:

$$P((e, a, v)|Y(e)) = \frac{\exp(z(\mathbf{e}, \mathbf{a}, \mathbf{v}, Y(e))}{\sum_{\hat{v} \in V_a} \exp(z(\mathbf{e}, \mathbf{a}, \hat{\mathbf{v}}, Y(e)))} \quad (11)$$

where $z()$ is a scoring function measuring the inference correlations between attributes. It is calculated according to the correlation between (e, a, v) and each attributional triple involved in $Y(e)$ in a compositional fashion:

$$z(\mathbf{e}, \mathbf{a}, \mathbf{v}, Y(e)) \propto \sum_{(e, \hat{a}, \hat{v}) \in Y(e)} P((a, v)|(\hat{a}, \hat{v}))(\mathbf{A_a} \cdot \mathbf{A_{\hat{a}}}) \quad (12)$$

Here $(\mathbf{A_a} \cdot \mathbf{A_{\hat{a}}})$ is the dot product of $\mathbf{A_a}$ and $\mathbf{A_{\hat{a}}}$, indicating the relatedness between the attributes A_a and $A_{\hat{a}}$. $P((a, v)|(\hat{a}, \hat{v}))$ is the conditional probability of attribute value (a, v) given $(\hat{a}, \hat{v})$, which is obtained over each entity in training data, indicating the correlation between the attribute values (a, v) and $(\hat{a}, \hat{v})$.

4.4 Optimization and Implementation Details

Here we introduce the learning and optimization details for the KR-EAR model. We define the optimization function as the log-likelihood of the objective function in Eq. (4):

$$O(\mathbf{X}) = \log(P(S, Y|\mathbf{X})) + \gamma C(\mathbf{X}) \quad (13)$$

where γ is a hyper-parameter weighting the regularization factor $C(\mathbf{X})$, which is defined as follows:

$$C(\mathbf{X}) = \sum_{e \in E} \left[||\mathbf{e}|| - 1\right]_+ + \sum_{r \in R} \left[||\mathbf{r}|| - 1\right]_+ + \sum_{e \in E} \sum_{i=1}^{|A|} \left[||\mathbf{e}\mathbf{W_i} + \mathbf{b_i}|| - 1\right]_+ + \sum_{i=1}^{|A|} \left[||\mathbf{V_i}|| - 1\right]_+ \quad (14)$$

where $[x]_+ = \max(0, x)$ returns the greater one between 0 and x. The regularization factor will normalize the embeddings during learning.

To solve the optimization problem, we adopt stochastic gradient descent (SGD) to minimize the optimization function. For learning, we iteratively select random mini-batch from the training set until converge.

Here we also introduce an implementation detail that will significantly influence the efficiency of KR-EAR learning.

Approximation of Softmax Function

In the KR-EAR model, it is impractical to directly compute the softmax functions $P(h|r, t, \mathbf{X})$, $P(t|h, r, \mathbf{X})$, $P(r|h, t, \mathbf{X})$, $P(v|e, a, \mathbf{X})$ and $P((e, a, v)|Y(e))$. The reason is that, the cost of computing normalizers for these softmax functions is proportional to $|E|$ and $|V|$, which is considerably huge for large-scale KGs like Freebase. Hence, we adopt negative sampling [Mikolov *et al.*, 2013] as computationally efficient approximation of full softmax function. Take $P(h|r, t, \mathbf{X})$ in Eq. (5) for example. It is approximated via negative sampling as follows:

$$P(h|r, t, \mathbf{X}) = \prod_{(h, r, t) \in S} \Big[\sigma(g(\mathbf{h}, \mathbf{r}, \mathbf{t})) \prod_{i=1}^{c_1} \mathbb{E}_{(h_i, r, t) \sim P(S^-)} \sigma(g(\mathbf{h}_i, \mathbf{r}, \mathbf{t}))\Big] \quad (15)$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is sigmoid function, S^- is the invalid triple set defined as $\mathbf{S}^- = \{(h', r, t)\}$, and $P(S^-)$ is a function randomly sampling instances from S^-. Similarly, we use negative sampling to approximate $P(t|h, r, \mathbf{X})$, $P(r|h, t, \mathbf{X})$, $P(v|e, a, \mathbf{X})$ and $P((e, a, v)|Y(e))$.

4.5 Complexity Analysis

As shown in Table 2, we compare the number of parameters and computational complexity of various baselines with our model. In this table, we denote N_e as the number of entities, N_r as the number of relations, N_a as the number of attributes, K as the vector dimension, S_1 as the number of relational triples for learning and S_2 as the number of attributional triples for learning.

Table 2: The number of parameters and computational complexity of models

Method	# of Parameters	Complexity
TransE	$(N_e + N_r + N_a)K$	$(S_1 + S_2)K$
TrasnH	$(N_e + 2N_r + 2N_a)K$	$(S_1 + S_2)K$
TransR	$N_eK + (N_r + N_a)K^2$	$(S_1 + S_2)K^2$
KR-EAR(TransE)	$(N_e + N_r)K + N_aK^2$	$S_1K + S_2K^2$
KR-EAR(TransR)	$N_eK + (N_r + N_a)K^2$	$(S_1 + S_2)K^2$

5 Experiments

5.1 Datasets and Experiment Setting

We evaluate our model on a typical large-scale KG Freebase. Freebase [Bollacker *et al.*, 2008] is a large-scale and growing collaborative KG consisting of data composed mainly by its community members, which provides general facts of the real world. For example, the triple (*Barack Obama*, `Spouse`, *Michelle Obama*) describes there is a relation `Spouse` between *Barack Obama* and *Michelle Obama*.

Dataset Construction

We construct the dataset for evaluation as follows:

(1) Filtering of low-frequency entities and relations. We select those entities and relations which have appeared in at least 30 triples.

(2) Filtering of reversed relations. In Freebase, each relation also corresponds to a reversed relation with swapped head and tail entities. For each relation with its reversed relation, we only need to keep one in our dataset. More specifically, for those many-to-one relations and their one-to-many reversed relations, we will keep the former and remove the latter, since the latter ones are usually at odds with human intuitions. For example, for the relation `people.person.nationality`, we will remove its reversed relation `!people.person.nationality`.

(3) Division between attributes and relations. In original Freebase, there is no explicit division between relational triples and attributional triples. We manually divide the original Freebase relations into two types: attributes and relations. Besides, we also do experiments in the data which we split relations and attributes according to their mapping properties (The classification accuracy is 86.2%) and get similar conclusion. Due to the limit of space, we don't report the results in this paper.

Finally, we build a dataset named as FB24k, and we randomly separate datas into training and testing sets. The statistics of the dataset is shown in Table 3.

Table 3: Statistics of datasets.

Dataset	FB24k
#Entities	23,634
#Relations	673
#Attributes	314
#Total Tripes	423,560
#Train (Relational Triples)	205,643
#Test (Relational Triples)	10,766
#Train (Attributional Triples)	196,850
#Test (Attributional Triples)	10,301

Using dataset FB24k, we evaluate the performance of our model and other baselines on the task of KG completion, which has been widely used for evaluation in previous works [Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b]. In this paper, we divide KG completion into three sub-tasks: (1) Entity Prediction; (2) Relation Prediction; and (3) Attribute Prediction.

5.2 Baselines and Parameter Settings

In the three sub-tasks of KG completion, we compare our model with several state-of-art KR models including TransE [Bordes *et al.*, 2013], TransH [Wang *et al.*, 2014b] and TransR [Lin *et al.*, 2015b], implemented with the source codes released by [Lin *et al.*, 2015b].

We tune our models using five-fold validation on the training set. We use a grid search to determine the optimal parameters and manually specify the parameter spaces learning rate λ for SGD among $\{0.1, 0.01, 0.001\}$, γ for soft constraints among $\{0.1, 0.01, 0.001\}$, the vector dimension k among $\{20, 50, 80, 100\}$ and all bias constant b_1, b_2, c_1, c_2 among -10 to 10. The best configurations are $\lambda = 0.001$, $\gamma = 0.1$, $k = 100$, $b_1 = 7$, $b_2 = -2$, $c_1 = 10$, $c_2 = 1$ and taking L_1 as dissimilarity metric. For training, we set the iteration number over all the training triples as 1000. The running time of per iteration is 14s for TransE and 297s for TransR in single thread.

5.3 Entity Prediction

Entity prediction aims to infer the possible head/tail entities in testing triples when one of them is missing. For each testing triple (h, r, t), we replace its head/tail entity with each entity in the dataset, and calculate the ranking score $\sigma(g(\mathbf{h}, \mathbf{r}, \mathbf{t}))$. Afterwards, we rank all candidate entities in dataset according to their scores in ascending order.

Following the setting in [Bordes *et al.*, 2013] , we report two measures as our evaluation metrics: the average rank of all correct entities (Mean Rank) and the proportion of correct entities ranked in top 10 (Hits@10). Note that, for a particular triple (h, r, t), its corrupted triples may also exist in the KG and should also be regarded as valid. The evaluation metrics may be unfair for those methods that rank other valid triples higher than (h, r, t). Hence, we filter out all other valid triples before ranking. Following the setting in [Bordes *et al.*, 2013], we name the filtered version as "Filter" and the unfiltered one as "Raw".

Results

We show the evaluation results on entity prediction in Table 4. From the table we observe that: (1) KR-EAR outperforms other baseline methods including TransE, TransH and TransR significantly and consistently in Mean Rank. This indicates that KR-EAR learns more reasonable embeddings for entities and relations; (2) KR-EAR(TransE) outperforms TransE and KR-EAR(TransR) outperforms TransR in Hit@10. This indicates KR-EAR can take advantage of traditional KR models.

5.4 Relation Prediction

Relation prediction aims to infer the possible relation between two given entities. For each testing triple (h, r, t), we replace its relation with each possible relation $\hat{r}$ in the KG and calculate the ranking score $\sigma(g(\mathbf{h}, \hat{\mathbf{r}}, \mathbf{t}))$. Afterwards, we rank all the candidate relations in the KG according to their scores in ascending order.

We report two measures as our evaluation metrics: Mean Rank and Hits@1.

Correlation between Relations & Attributes

It has shown that type-constraints can generally support multi-relational data modeling with latent variable models [Krompaß *et al.*, 2015]. We argue that the type information of entities is a special case of entity attributes. In KR-EAR, we can easily adopt the constraints between the attributes of head and tail entities for relation prediction, named as CRA.

Results

We show the evaluation results on relation prediction in Table 5. From the table we observe that: (1) KR-EAR again outperforms other baseline methods significantly and consistently in both Mean Rank and Hits@1, while TransE, TransH and TransR achieve close results in this sub-task. (2) For KR-EAR(TransE) and KR-EAR(TransR), CRA can further bring the improvements of 2.5% and 1.4% in Hits@1, and also get

Table 4: Evaluation results on entity prediction.

Entity	Head				Tail				Total			
Metric	Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)		Mean Rank		Hits@10 (%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	385	277	20.2	39.2	134	124	51.4	66.7	259	200	35.8	53.0
TransH	416	309	17.7	35.4	147	138	50.0	65.0	282	224	33.9	50.2
TransR	394	285	20.5	41.2	125	116	53.4	71.0	260	200	37.0	56.1
KR-EAR(TransE)	295	198	22.7	39.6	77	69	54.2	69.5	186	133	38.5	54.5
KR-EAR(TransR)	**268**	**170**	**23.4**	**43.0**	**75**	**66**	**55.7**	**71.5**	**172**	**118**	**39.5**	**57.3**

Table 5: Evaluation results on relation prediction.

Metric	Mean Rank		Hits@1 (%)	
	Raw	Filter	Raw	Filter
TransE	3.1	2.8	65.9	83.8
TransH	3.4	3.1	64.9	84.1
TrasnR	3.4	3.1	65.2	84.5
KR-EAR(TransE)	2.4	2.1	67.9	86.2
+ CRA	**1.8**	**1.6**	70.9	88.7
KR-EAR(TransR)	2.6	2.2	66.8	89.0
+ CRA	1.9	**1.6**	**71.5**	**90.4**

a lower Mean Rank. This demonstrates the effectiveness of considering entity attributes for relation prediction.

5.5 Attribute Prediction

Attribute prediction aims to predict the missing attributes for an entity. This task used to be a part of entity prediction in previous works [Bordes *et al.*, 2013; Wang *et al.*, 2014b; Lin *et al.*, 2015b]. For each testing triple (e, a, v), we replace the attribute value with each possible value $\hat{v}$ of the attribute and calculate a ranking score $\sigma(h(\mathbf{e}, \mathbf{a}, \hat{\mathbf{v}}))$. Afterwards, we rank all the candidates according to their scores in ascending order.

Note that, KR-EAR can also consider Attribute Correlations (AC) by ranking candidates according to $\sigma(h(\mathbf{e}, \mathbf{a}, \hat{\mathbf{v}}))\sigma(z(\mathbf{e}, \mathbf{a}, \hat{\mathbf{v}}, Y(e)))$.

We report two evaluation measures for attribute prediction: Mean Rank and Hits@1.

Table 6: Evaluation results on attributes prediction.

Metric	Mean Rank		Hits@1 (%)	
	Raw	Filter	Raw	Filter
TransE	10.7	5.6	36.5	55.9
TransH	10.7	5.6	38.5	57.9
TrasnR	9.0	3.9	42.7	65.6
KR-EAR(TransE)	8.3	3.2	47.2	69.0
+AC	**7.5**	**3.0**	49.4	70.4
KR-EAR(TransR)	8.3	3.2	47.6	69.8
+AC	**7.5**	**3.0**	**49.8**	**70.8**

Results

We show the evaluation results on attribute prediction in Table 6. From the table we observe that: (1) KR-EAR still outperforms other baselines significantly and consistently. This verifies the necessity of modeling attribute prediction as classification instead of translation in traditional KR models; (2) For both KR-EAR(TransE) and KR-EAR(TransR), taking attribute correlations into consideration can achieve 1.4% and 1.0% improvements in Hits@1. This indicates that attribute correlations are useful in attribute prediction.

Table 7: Illustration of Attribute Correlations

Attribute	Correlated Attributes
Profession	Marital Status, Nationality, Gender, Language, Ethnicity
Release Region of Film	Country of Film, Language of Film, Release Date of Film, Genre of Film
Time Zone of Location	Country of Location, Currency of Location
Musical Genres	Instruments Played, Recording contributions, Profession, Instrument(s) or Vocal Role
TV Genres	Country of TV, Languages of TV, Original Network of TV, Regular Acting Performances

Illustration of Attribute Correlations

In Table 7, we give some examples of attribute correlations obtained on the FB24k training set via KR-EAR. We can find that when given an attribute, the related attributes often reflect reasonable correlations in common-sense. This indicates that KR-EAR can effectively capture the correlations among attributes.

6 Conclusion and Future Work

In this paper, we distinguish existing KG-relations into attributes and relations, and propose a new KR model with entities, attributes and relations (KR-EAR). In addition, we also encode the correlations between entity attributes in KR-EAR. In experiments, we evaluate our model on three subtasks for predicting entities, relations and attributes. By explicitly modeling entity attributes, KR-EAR can significantly and consistently outperform state-of-the-art KR models on all three sub-tasks.

In the future, we will explore more in the following research directions: (1) Currently KR-EAR regards the inference of entities, relations and attributes independently. In the future, we can employ probabilistic graphical model to further capture the complicated correlations between them. (2) In this paper, we split relations and attributes manually which consumes large amount of time. In the future, we can employ how to split relations and attributes by held-out machine learning methods.

Acknowledgments

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61572273 and 61532010) and the Tsinghua University Initiative Scientific Research Program (20151080406).

References

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of KDD*, pages 1247–1250, 2008.

[Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*, pages 301–306, 2011.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795, 2013.

[Bordes *et al.*, 2014] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.

[Chen *et al.*, 2013] Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *Proceedings of ICLR*, 2013.

[He *et al.*, 2015] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of CIKM*, pages 623–632. ACM, 2015.

[Ji *et al.*, 2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, pages 687–696, 2015.

[Ji *et al.*, 2016] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. *Proceedings of AAAI*, 2016.

[Kemp *et al.*, 2006] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of AAAI*, volume 3, page 5, 2006.

[Krompaß *et al.*, 2015] Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *Proceedings of ISWC*, 2015.

[Lin *et al.*, 2015a] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Modeling relation paths for representation learning of knowledge bases. *Proceedings of EMNLP*, 2015.

[Lin *et al.*, 2015b] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, pages 2181–2187, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119, 2013.

[Miller *et al.*, 2009] Kurt Miller, Michael I Jordan, and Thomas L Griffiths. Nonparametric latent feature models for link prediction. In *Proceedings of NIPS*, pages 1276–1284, 2009.

[Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML*, pages 809–816, 2011.

[Nickel *et al.*, 2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of WWW*, pages 271–280, 2012.

[Singh and Gordon, 2008] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of KDD*, pages 650–658, 2008.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pages 926–934, 2013.

[Suchanek *et al.*, 2011] Fabian M Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *Proceedings of VLDB*, 5(3):157–168, 2011.

[Sutskever *et al.*, 2009] Ilya Sutskever, Joshua B Tenenbaum, and Ruslan Salakhutdinov. Modelling relational data using bayesian clustered tensor factorization. In *Proceedings of NIPS*, pages 1821–1828, 2009.

[Wang *et al.*, 2014a] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*, pages 1591–1601, 2014.

[Wang *et al.*, 2014b] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, pages 1112–1119, 2014.

[Zhu, 2012] Jun Zhu. Max-margin nonparametric latent feature models for link prediction. *In Proceedings of ICML*, pages 719–726, 2012.