Type-augmented Relation Prediction in Knowledge Graphs

Zijun Cui,[1] Pavan Kapanipathi,[2] Kartik Talamadupula,[2] Tian Gao,[2] Qiang Ji[1]
[1]Rensselaer Polytechnic Institute
[2] IBM Research
[1]{cuiz3, jiq}@rpi.edu, [2]{kapanipa, krtalamad, tgao}@us.ibm.com

Abstract

Knowledge graphs (KGs) are of great importance to many real world applications, but they generally suffer from incomplete information in the form of missing relations between entities. Knowledge graph completion (also known as relation prediction) is the task of inferring missing facts given existing ones. Most of the existing work is proposed by maximizing the likelihood of observed instance-level triples. Not much attention, however, is paid to the ontological information, such as type information of entities and relations. In this work, we propose a type-augmented relation prediction (TaRP) method, where we apply both the type information and instance-level information for relation prediction. In particular, type information and instance-level information are encoded as prior probabilities and likelihoods of relations respectively, and are combined by following Bayes' rule. Our proposed TaRP method achieves significantly better performance than state-of-the-art methods on four benchmark datasets: FB15K, FB15K-237, YAGO26K-906, and DB111K-174. In addition, we show that TaRP achieves significantly improved data efficiency. More importantly, the type information extracted from a specific dataset can generalize well to other datasets through the proposed TaRP model.

Introduction

Knowledge graphs (KGs) have gained significant popularity due to successful applications to many different AI tasks such as question answering (Huang et al. 2019), recommendation (Wang et al. 2019a), dialogue generation (Xu, Bao, and Zhang 2020), and natural language inference (Wang et al. 2019b; Kapanipathi et al. 2020). However, KGs are generally incomplete and suffer from missing relations between entities (Socher et al. 2013; West et al. 2014). The task of knowledge graph completion or relation prediction is aimed at tackling this issue, i.e., inferring missing facts given existing ones. For example, in Figure 1, given two entities, e.g., `Helen_Mirren` and `The_Queen`, the relation prediction task predicts if those entities are connected by any of the existing relations in the KG, e.g., `actor`.

Relation prediction methodologies are mostly based on KG embeddings, and can primarily be categorized based on the two kinds of information they use from KGs: (i)

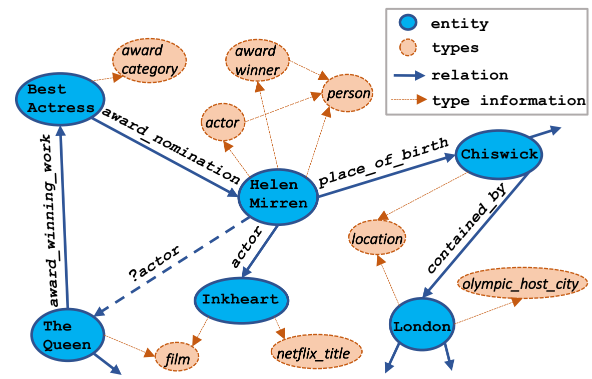

Figure 1: An example in a knowledge graph (KG).

Instance-level information, i.e., existing triples connecting entities through relations, such as `Helen_Mirren` → `place_of_birth` → `Chiswick`; and (ii) Ontological information, i.e., meta information about entities and relations, such as the type information of entities e.g., `Helen_Mirren` is of types **{*actor, award_winner, person*}**. The majority of existing methods use merely instance-level information for learning the embeddings (Sun et al. 2019; Zhang et al. 2019), while a few other models use both instance-level information and ontological information (Hao et al. 2019; Garg et al. 2019; Xie et al. 2016). Ontological information such as type information can intuitively help relation prediction, as most relations may connect two distinct types of entities as domain and range. For example, the relation `place_of_birth` always connects entities of type ***person*** to entities of type ***location***. Integrating such type information into instance-level training triples can benefit the relation prediction task, in particular when there is a lack of sufficient training data for learning embeddings.

A few existing embedding based models with such type information integrated have shown success (Guo et al. 2015; Xie, Liu, and Sun 2016; Ma et al. 2017; Jain et al. 2018; Garg et al. 2019; Hao et al. 2019). However, these models integrate the ontological information through the model training procedure for better learning the embeddings, and are hence prone to the following drawbacks: (i) the type information is not explicitly differentiated from the instance-level information, and a single set of model parameters are learned by considering two kinds of information jointly; (ii) the type information is tightly encoded into the objective function, making the integration highly reliant on the training procedure and hence less flexible in augmenting new embedding techniques. We refer to such integration procedures as *feature-level integration*. Instead, we proposed an effective *decision-level integration*: given the type information and instance-level information encoded as prior probabilities and likelihoods respectively, the proposed decision-level integration combines the two kinds of information by following Bayes' rule.

In this paper, we propose a simple but effective framework to augment existing embedding based models with type information. The contributions of our work are as follows:

- The proposed decision-level integration framework is independent of the embedding based model, and can be flexibly applied for augmenting different embedding based models without additional training.
- The proposed type-augmented relation prediction (TaRP) method achieves better relation prediction performance than state-of-the-art models on three benchmark datasets. Furthermore, we show that by incorporating the type information, TaRP has less dependency on training data, and thus is more data efficient.
- We empirically demonstrate that the type information extracted from a specific dataset can generalize well to other, different datasets through the proposed TaRP model.

Related Work

KG embedding based methods have been widely explored for the KG completion task. The general methodology of the embedding based method is to define a score function for triples within a continuous embedding space. The score function usually takes the form $f_r(\mathbf{e}_h, \mathbf{e}_t)$, where $\mathbf{e}_h, \mathbf{e}_t$ are head and tail entity embeddings. The score function measures the salience of a candidate triple (e_h, r, e_t), and embeddings of entities and relations are learned by optimizing the score function. TransE (Bordes et al. 2013) represents entities and relations in d-dimensional vector space, i.e., $\mathbf{e}_h, \mathbf{e}_r, \mathbf{r} \in R^d$ and learns the embeddings by assuming the translation principle $\mathbf{e}_h + \mathbf{r} \approx \mathbf{e}_t$ with the proposed score function $f_r(\mathbf{e}_h, \mathbf{e}_t) = -||\mathbf{e}_h + \mathbf{r} - \mathbf{e}_t||$. Along the lines of the translation-based methods (first introduced by TransE), many more advanced methods have been proposed, such as TransH (Wang et al. 2014) and TransD (Ji et al. 2015). More recently, RotatE (Sun et al. 2019) and QuatE (Zhang et al. 2019) have been proposed by representing the entities and relations using complex vectors. In addition, neural networks have also been introduced to learn robust embedding based models, such as ConvE (Dettmers et al. 2018) and HypER (Balažević, Allen, and Hospedales 2019). These methods learn embeddings based on instance-level information observed from existing triples, without considering the rich ontological information that exists.

There are existing embedding based methods that explore the usage of the ontological information. DistMult (Yang et al. 2014) considered semantic similarity and associated related entities using Hadamard product of embeddings. Guo et al. (2015) proposed semantically smooth embedding, where the type information is encoded as smoothness constraints. Ma et al. (2017) measured the type-based semantic similarity between entities and relations, and that semantic similarity served as the prior probability. Besides semantic types of entities, underlying hierarchy structures among types are also considered. Xie, Liu, and Sun (2016) proposed type-specific entity projections by applying hierarchical type information, and devised type-embodied knowledge representation learning (TKRL). Hao et al. (2019) introduced instance-view KG and ontology-view KG, and the hierarchy structures among types are explicitly represented within the ontology-view KG. Universal embeddings are then learned by considering the two types of KGs jointly. Zhang et al. (2020b) proposed hierarchy-aware KG embedding for link prediction. In addition, hierarchical type information is extracted as logical propositions for the quantum embeddings (Garg et al. 2019), and symbolic KGs are represented with embedding vectors in a logic structure preserving manner. Jain et al. (2018) considered type information for entity prediction without explicit supervision. Besides the type information, other ontological information is also explored. DKRL (Xie et al. 2016) applies entity descriptions. SSP (Xiao et al. 2017) uses the topic distribution of entity descriptions to construct semantic hyperplanes. All of these models integrate the ontological information into the instance-level information at the feature-level in order to learn the embeddings better.

Besides embedding based methods, path based methods have also been proposed for the KG completion task (Lao, Mitchell, and Cohen 2011; Das et al. 2016; Chen et al. 2018; Zhang et al. 2020a). Lei et al. (2019) utilizes type semantics from the relation to obtain attention that constrains the semantics of the entity. Path based methods suffer from high computational cost because of the path finding procedure; in this paper, we focus on embedding based approaches.

TaRP: Type-augmented Relation Prediction Model

In this section, we present our type-augmented relation prediction (TaRP) framework, which augments existing embedding models with type information from the knowledge graph. Our framework consists of two components: (a) a prior model where we encode the type information as prior probabilities, detailed in Section ; and (b) a likelihood model based on existing instance-level information. Any existing embedding-based model can be applied as the likelihood model, and we briefly describe three embedding-based models used for our experiments in Section . The framework integrates information from the prior and likelihood models using Bayes' rule (detailed in Section).

Type Information Encoding

In this section, we detail our approach to encoding the type information as prior probabilities. We denote a knowledge graph as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ are the entity set, the relation set, and the type set respectively. We first define hierarchy-based type weights for each type of entities $e \in \mathcal{E}$ and relations $r \in \mathcal{R}$. For each triple (e_h, r, e_t) in $\mathcal{G}$, we then define the type-based prior probability of relation r conditioned on the entity pair (e_h, e_t) by measuring the semantic similarity, which is calculated based on the correlation between the type sets of entities and relations.

Hierarchy-based type weights Type sets in most KGs have an underlying hierarchy, such as the structure among types **{*actor, award_winner, person*}** in Figure 1. Such hierarchy can reflect the abstractness of a type. We hypothesize that more specific types, e.g., ***actor***, are more useful than abstract types, e.g., ***person***. In order to capture this intuition and leverage a hierarchical structure, we use the notion of *hierarchy-based type weights*. Given an entity e, its type set is denoted as $T_e \subset \mathcal{T}$. A hierarchical structure among a subset of the possible types is of the format $H = /t_1/t_2/..../t_k/.../t_K$[1], where $t_k \in T_e$, K is the total number of hierarchy levels, t_K is the most specific semantic type, and t_1 is the most abstract semantic type. Instead of treating the types in the hierarchy H equally, we weight the types based on their positions in the hierarchy. The weight of type t_k in relation to its position in the hierarchy H of entity e is defined as:

$$w_e^H(t_k) = \frac{\exp(k-1)}{\sum_{j=0}^{K-1} \exp(j)} \tag{1}$$

Multiple hierarchical structures can exist for a given entity, with each hierarchy including a subset of the possible types. For example, in Figure 1, entity $e = \texttt{Helen\_Mirren}$ has three possible hierarchical structures among its types: H_1 = ***/person/actor***, H_2 = ***/person/award_winner***, and H_3 = ***/person***. Type ***person*** is included in all three hierarchies. Following Eq.1, we have $w_e^{H_i}(\textbf{\textit{person}}) = 0.27$, $i = \{1, 2\}$, and $w_e^{H_3}(\textbf{\textit{person}}) = 1$. For each type $t \in T_e$, we calculate its hierarchy-based weight as $w_e(t) = \min(w_e^{H_1}(t), w_e^{H_2}(t), ..., w_e^{H_N}(t))$, where N is the total number of hierarchies containing type t. In the example above, we thus have $w_e(\textbf{\textit{person}}) = 0.27$, $w_e(\textbf{\textit{actor}}) = 0.73$, and $w_e(\textbf{\textit{award\_winner}}) = 0.73$. For the type sets of relations, we consider both the head type set $T_{r,head}$ and tail type set $T_{r,tail}$, defined as:

$$\begin{aligned} T_{r,head} &= \cup_{e \in \text{Head}(r)} T_e \\ T_{r,tail} &= \cup_{e \in \text{Tail}(r)} T_e \end{aligned} \tag{2}$$

where $\text{Head}(r) = \{e_h | (e_h, r, e_t) \in \mathcal{G}, \forall e_t \in \mathcal{E}\}$ indicates the set of head entities of relation r; and $\text{Tail}(r) = \{e_t | (e_h, r, e_t) \in \mathcal{G}, \forall e_h \in \mathcal{E}\}$ indicates the set of tail entities of relation r. We then calculate the type weights as:

$$\begin{aligned} w_{r,head}(t) &= \sum_{e \in \text{Head}(r)} w_e(t), \quad \text{for } t \in T_{r,head} \\ w_{r,tail}(t) &= \sum_{e \in \text{Tail}(r)} w_e(t), \quad \text{for } t \in T_{r,tail} \end{aligned} \tag{3}$$

Type-based prior probability Given a triple $(e_h, r, e_t) \in \mathcal{G}$, we measure the semantic similarity between entities and relations based on the correlation between their type sets. The similarity score $s(\cdot, \cdot)$ is calculated as

$$\begin{aligned} s(e_h, r) &= \frac{\sum_{t \in T_{r,head} \cap T_{e_h}} w_{r,head}(t)}{\sum_{t \in T_{r,head}} w_{r,head}(t)} \\ s(e_t, r) &= \frac{\sum_{t \in T_{r,tail} \cap T_{e_t}} w_{r,tail}(t)}{\sum_{t \in T_{r,tail}} w_{r,tail}(t)} \end{aligned} \tag{4}$$

where $T_{r,head} \cap T_{e_h} = \{t | t \in T_{r,head} \text{ and } t \in T_{e_h}\}$ and $T_{r,tail} \cap T_{e_t} = \{t | t \in T_{r,tail} \text{ and } t \in T_{e_t}\}$, and $0 \le s(\cdot, \cdot) \le 1$. The prior probability $p(r|\mathcal{T}(e_h, e_r, \mathcal{R}))$ is then defined based on similarity scores as,

$$p(r|\mathcal{T}(e_h, e_t, \mathcal{R})) \triangleq \frac{s(e_h, r)s(e_t, r)}{\sum_{r' \in \mathcal{R}} s(e_h, r')s(e_t, r')} \tag{5}$$

where $\mathcal{T}(e_h, e_t, \mathcal{R})$ denotes the type information related to the entity pair (e_h, e_t) and the relation set $\mathcal{R}$. To obtain a valid prior probability, for $\forall r \in \mathcal{R}$, both the head type set $T_{r,head}$ and the tail type set $T_{r,tail}$ are required to be non-empty. For the cases where the type set of an entity e is empty (e can be either head entity e_h or the tail entity e_t), i.e., $T_e = \emptyset$, we assign uniform similarity scores, i.e., $s(e, r') = 1, \forall r' \in \mathcal{R}$. Different from the existing work (Ma et al. 2017) that measures semantic similarity by treating each type equally, our type-based prior probability with hierarchy-based type weights encodes not only the type information, but also the underlying hierarchies among types.

$|T_{r,head}|$ or $|T_{r,tail}|$ of relation r can be large containing noisy types, which may weaken the relation prediction performance of the prior model. To remove noisy types, we introduce a threshold η. Type $t \in T_{r,*}$ will be kept if

$$w_{r,*}(t) \ge W_{r,*}^{min} + \eta \times (W_{r,*}^{max} - W_{r,*}^{min}) \tag{6}$$

where $W_{r,*}^{max}$ and $W_{r,*}^{min}$ are the maximum and the minimum weight of the type set $T_{r,*}$ respectively. If $w_{r,*}(t)$ is not sufficiently high (i.e., Eq.6 is not satisfied), type t will be removed from $T_{r,*}$ without further consideration. $* = \{\text{head}, \text{tail}\}$. Threshold η varies across datasets, and is chosen based on performance on the validation set.

Embedding Based Models

Embedding based models represent relations and entities in a continuous embedding space. We denote the embedding of the head and tail entities as $\mathbf{e}_h$ and $\mathbf{e}_t$ respectively. The embedding of relation r is $\mathbf{r}$. A score function $f_r(\mathbf{e}_h, \mathbf{e}_t)$ is usually defined as a measurement of the salience of a triple (e_h, r, e_t). Embeddings are then learned by optimizing the score function based on instance-level triples. We consider three embedding based models:

- **TransE** (Bordes et al. 2013):

$$f_r(\mathbf{e}_h, \mathbf{e}_t) = -||\mathbf{e}_h \cdot \mathbf{r} - \mathbf{e}_t|| \tag{7}$$

  where $\mathbf{e}_h, \mathbf{e}_r \in \mathbb{R}^d$ and $\mathbf{r} \in \mathbb{R}^d$.

- **RotatE** (Sun et al. 2019):

$$f_r(\mathbf{e}_h, \mathbf{e}_t) = -||\mathbf{e}_h \circ \mathbf{r} - \mathbf{e}_t|| \tag{8}$$

  where $\mathbf{e}_h, \mathbf{e}_r \in \mathbb{C}^d$ and $\mathbf{r} \in \mathbb{C}^d$.

[1]For each dataset, we process the hierarchy into such a format if needed.

- **QuatE** (Zhang et al. 2019):

$$f_r(\mathbf{e}_h, \mathbf{e}_t) = -||\mathbf{e}_h \otimes \frac{\mathbf{r}}{|\mathbf{r}|} - \mathbf{e}_t|| \tag{9}$$

  where $\mathbf{e}_h, \mathbf{e}_r \in \mathbb{H}^d$ and $\mathbf{r} \in \mathbb{H}^d$.

The learned embeddings for the entities and relations from these models therefore contain only instance-level information without the type information. For each triple (e_h, r, e_t), we define the likelihood of relation based on their corresponding embeddings as

$$p(e_h, e_t|r) \triangleq \exp(f_r(\mathbf{e}_h, \mathbf{e}_t)) \tag{10}$$

It is intuitive that the likelihood of relation will be small with a low score $f_r(\mathbf{e}_h, \mathbf{e}_t)$.

Type Information Integration

The final step is to integrate the type information with the instance-level information at the decision-level. Given a triple (e_h, r, e_t), we obtain the prior probability of relation r based on the type information, i.e., $p(r|\mathcal{T}(e_h, e_t, \mathcal{R}))$ as described in Section . Next, we obtain the likelihood of relation based on embeddings-based model that learns from instance-level, i.e., $p(e_h, e_t|r)$ (Section). Combining them together, we obtain the posterior probability of relation r by following the Bayes' rule, i.e.,

$$\begin{aligned} &p(r|e_h, e_t, \mathcal{T}(e_h, e_t, \mathcal{R})) \\ &\propto p(e_h, e_t|r)p(r|\mathcal{T}(e_h, e_t, \mathcal{R})) \end{aligned} \tag{11}$$

The posterior probability $p(r|e_h, e_t, \mathcal{T}(e_h, e_t, \mathcal{R}))$ thus contains both the type information and the instance-level information. Our proposed decision-level integration is independent of the embedding techniques. This is the prominent differentiating factor compared to the existing works, e.g.,(Ma et al. 2017; Xie, Liu, and Sun 2016), that tightly integrates the type information at the feature-level (integrated in the objective function) making it less flexible to evolving embedding techniques.

Experiments

To evaluate the performance of our type-augmented relation prediction (TaRP) approach, we first perform ablation studies on the prior model; and then evaluate the performance of the TaRP model. We demonstrate the effectiveness of the TaRP model by comparing it to three baseline embedding based models: TransE, RotatE, and QuatE. In addition, we show that by incorporating type information, TaRP is much more data efficient than existing methods. Furthermore, we demonstrate the generalization ability of type information. In the end, we compare our approach to state-of-the-art models that also apply ontological information.

Datasets We consider three benchmark datasets for the relation prediction task: FB15K (Bordes et al. 2013), YAGO26K-906 (Hao et al. 2019) and DB111K-74 (Hao et al. 2019). FB15K is a popular benchmark dataset for the KG completion task, and its type information has been explored by most of the prior work, such as (Ma et al. 2017; Guo et al. 2015; Xie, Liu, and Sun 2016). YAGO26K-906 and DB111K-906 are two very recent datasets containing explicit ontological information, and have not been widely considered by related work.

FB15K consists of triples extracted from the FreeBase knowledge graph (Bollacker et al. 2008). The same type information is applied as introduced in (Xie, Liu, and Sun 2016) for FB15K. Both YAGO26K-906 and DB111K-174 contain two types of KGs: instance KG and ontology KG, which are connected to each other through type links. The instance KGs of YAGO26K-906 and DB111K-174 consist of triples extracted from the YAGO knowledge graph (Rebele et al. 2016) and the DBpedia knowledge graph (Lehmann et al. 2015) respectively; and are applied for the relation prediction task. Type links and ontology KGs are collected as type information. Statistical information about the three datasets is shown in Table 1.

Dataset	#Rel.	#Ent.	#Types
FB15K	1,345	14,951	663
YAGO26K-906	34	26,078	226
DB111K-174	298	98,336	242

Table 1: Statistics of dataset.

On all three datasets, for each relation, the obtained head type set and tail type set are non-empty. For each entity from FB15K and DB111K-74, the type set is non-empty. On YAGO26K-906, only 8,948 entities have non-empty type sets. As a result, $4,149(10.6\%)$ testing triples have type information for both head and tail entities; $30,839(78.9\%)$ triples have type information for only head entity or only tail entity; and $4,086(10.5\%)$ triples don't have type information for both head and tail entities. For the cases where the type set of entity e is empty (e can be either head entity e_h or tail entity e_t), we assign uniform similarity scores, i.e., $s(e, r') = 1, \forall r' \in \mathcal{R}$.

Evaluation protocol For each triple (e_h, r, e_t) in the testing set, we replace the relation r with every relation $r' \in \mathcal{R}$. We calculate the posterior probabilities $p(r'|e_h, e_t, \mathcal{T}(e_h, e_t, \mathcal{R}))$ of all replacement triples and rank these probabilities in descending order. We apply the filter setting (Ma et al. 2017). Two standard measures are considered as evaluation matrices: mean rank (MR) and Hits@N. A higher Hits@N and a lower MR mean better performance. In all the experiments, we report both Hits@1 and Hits@10.

Experimental Settings TaRP has one hyper-parameter threshold η. On each dataset, we select the threshold η from $\{0, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$ that achieves the best relation prediction performance (Hits@1) on the validation set. On FB15K and YAGO26K-906, $\eta = 0.1$. On DB111K-174, $\eta = 0$. We report the averaged size of type set over all the entities or relations as shown in Table 2[2]. For baseline embedding based models, we directly reuse the best configurations provided by previous studies (Sun et al. 2019; Zhang et al. 2019).

[2] $|T_{r,head}|$ and $|T_{r,tail}|$ are calculated with optimal thresholds applied.

Dataset	Entity	Relation	
	$\|T_e\|$	$\|T_{r,head}\|$	$\|T_{r,tail}\|$
FB15K	12	20	19
YAGO26K-906	9	6	5
DB111K-174	2	7	12

Table 2: Averaged size of the type set.

Ablation Studies on the Prior Model

We perform ablation studies to show the effectiveness of: 1) the hierarchy-based type weights; 2) the type information.

Effectiveness of hierarchy-based type weights To demonstrate the effectiveness of the proposed hierarchy-based type weights, we consider uniform weights for comparison, and calculate the prior probabilities based on types with uniform weights. We compare the relation prediction performance of the prior model with hierarchy-based weights to the performance of the prior model with uniform weights. We perform the evaluation on FB15K dataset. Results are shown in Table 3 where the prior model with hierarchy-based weights achieves much better performance than the prior model with uniform weights. These results empirically demonstrate the effectiveness of the hierarchy-based type weights.

Type weights	FB15K		
	MR	Hits@1	Hits@10
Uniform	26	4.95	43.88
Hierarchy-based	**2.9**	**64.10**	**97.10**

Table 3: Effectiveness of the hierarchy-based type weights.

Effectiveness of type information To study the effectiveness of the type information, we evaluate the relation prediction performance of the prior model by considering the type information of 1) only head entity (H); 2) only tail entity (T); 3) both head and tail entities (H+T). Results in Table 4 shows that considering the type information of head and tail entities jointly, the prior model achieves the best performance. These results depict that both the type information of head and tail entities are effective in relation prediction.

Type Info.	FB15K		
	MR	Hits@1	Hits@10
H	23.2	8.00	46.90
T	20.3	9.00	50.20
H+T	**2.9**	**64.10**	**97.10**

Table 4: Effectiveness of the type information.

Evaluation of the TaRP Model

We evaluate the TaRP model by first comparing it to three baseline embedding based models. In addition, we show the data efficiency of the proposed TaRP model by reducing the number of training triples. More importantly, we perform cross-dataset evaluation and empirically demonstrate the generalization ability of the type information.

Comparisons to baseline models As introduced in Section 3, we consider three baseline embedding based models: TransE, RotatE, and QuatE. The embeddings of entities and relations are obtained by directly running baseline models with reported best hyper-parameter settingsIn addition, to demonstrate the effectiveness of the proposed decision-level integration, we enrich the existing training sets by adding type information as addition training triples; and train the embedding based models on enriched training sets for comparison. On YAGO26K-906 and DB111K-174, triples from ontology KG and type links can be directly used as additional training triples. On FB15K, given an entity e and its hierarchical type $/t_1/t_2/.../t_K$, we collect type triples as (e, r_1, t_K) with $r_1 = \texttt{type}$ and $\{(t_k, r_2, t_{k-1})\}_{k=2}^{K}$ with $r_2 = \texttt{is\_a}$. The embedding based models trained on enriched training sets can thus learn embeddings based on both existing triples and the type information. In other words, the type information is fused with instance-level information at the feature-level. We denote the embedding based models learned from enriched training sets as: TransE(w/Type), RotatE(w/Type) and QuatE(w/Type). By combining the prior model with three embedding based models separately, we obtain three TaRP models: TaRP-T, TaRP-R, and TaRP-Q. The results are shown in Table 5.

From Table 5, we can see that all three TaRP models achieve performance improvement on all three benchmark datasets compared to the corresponding baseline embedding based models. In particular, on FB15K and DB111K-174, the improvement is significant. For instance, TaRP-R obtains 92.91% for Hits@1 on FB15K, achieving 12.71% improvement compared to RotatE. On the other hand, on both YAGO26K-906 and DB111K-174, embedding based models trained on enriched training sets achieve improved performance compared to baseline embedding based models. However, for most of the embedding based models trained on enriched training sets, the achieved performance improvement is not as significant as the improvement achieved by the proposed TaRP model. TaRP models achieve overall better performance than the embedding based models trained on enriched training sets. For example, on DB111K-174, QuatE(w/Type) obtains 60.49% for Hits@1; though higher than the 58.60% obtained by QuatE, this is still significantly worse than the 76.60% obtained by TaRP-Q. In addition, on FB15K, the embedding-based models trained with type triples perform worse than the embedding-based models trained without type triples. The reason may be that type triples collected from FB15K can contain errors[3] which leads to decreased performance. Our proposed prior model directly applies the type information, and hence errors introduced by type triples do not affect the TaRP models.

These results show that by incorporating the type information, the TaRP model can always achieve better performance with different baseline embedding based models.

[3]For example, for type ***/book/author***, the collected type triple (***author***, `is_a`, ***book***) is not true.

Method	FB15K			YAGO26K-906			DB111K-174		
	MR	Hits@1	Hits@10	MR	Hits@1	Hits@10	MR	Hits@1	Hits@10
TransE	3.64	76.50	92.30	1.12	90.70	99.92	4.76	66.60	86.70
RotatE	2.38	80.20	97.80	1.10	92.84	99.90	4.53	65.90	93.80
QuatE	4.01	82.20	94.90	1.33	91.65	98.96	8.56	58.60	88.90
TransE(w/Type)	3.32	79.37	91.56	1.12	90.70	99.93	4.16	67.64	91.91
RotatE(w/Type)	3.67	73.63	96.44	**1.08**	**93.31**	99.93	3.47	70.08	96.42
QuatE(w/Type)	3.98	80.82	92.97	1.32	91.98	99.09	7.63	60.49	89.14
TaRP-T	1.84	88.90	99.00	1.10	90.80	**99.98**	1.61	74.80	99.40
TaRP-R	**1.16**	**92.91**	**99.84**	**1.08**	92.84	**99.98**	**1.52**	76.50	**99.50**
TaRP-Q	1.64	91.60	99.50	1.14	92.93	99.79	1.56	**76.60**	99.40

Table 5: Evaluation of the Type-augmented Relation Prediction(TaRP) model

More importantly, the TaRP models achieve overall better performance than the embedding based models trained with type triples, indicating that the proposed decision-level integration procedure is a more effective integration. In addition, through the proposed integration approach, the type information can be directly combined with embedding based models without additional training.

Data efficiency of the TaRP model We consider the embeddings that are learned from a small subset of training triples. Given insufficient training data, the quality of the learned embeddings will be lower. We compare the TaRP model where embeddings are learned from only a subset of training triples to the embedding based model that is trained on complete training sets. We perform this evaluation on FB15K. RotatE is applied as the baseline embedding based model. We extract the subset of training triples with respect to each relation individually. Results are shown in Table 6. As shown, on FB15K, by integrating the type information, TaRP-R achieves better performance than RotatE with only 20% of the training data. These results show that TaRP achieves better performance than the embedding-based model using much lesser training data. By leveraging type information, TaRP has lesser dependency on the data, i.e., is more data efficient.

Method	FB15K		
	MR	Hits@1	Hits@10
RotatE(100% D)	2.38	80.20	97.80
TaRP-R(20% D)	1.90	83.20	99.20
TaRP-R(40% D)	1.74	85.90	99.60
TaRP-R(60% D)	1.73	84.90	99.70
TaRP-R(80% D)	1.71	85.50	99.70

Table 6: Data efficiency of the TaRP model(D:training data)

Cross-dataset evaluation of the TaRP model To demonstrate the generalization ability of the type information, we perform cross-dataset evaluation. In particular, the type information is collected from outside of the dataset. Given two knowledge graphs $\mathcal{G}^1 = \{\mathcal{E}^1, \mathcal{R}^1, \mathcal{T}^1\}$ and $\mathcal{G}^2 = \{\mathcal{E}^2, \mathcal{R}^2, \mathcal{T}^2\}$, for each entity $e \in \mathcal{E}^1$, we perform exact string matching to find if there exists a matched entity $e \in \mathcal{E}^2$. If so, we collect the type information for $e \in \mathcal{E}^1$ as $T_e = \{t_k\}_{k=1}^K$, where $t_k \in \mathcal{T}^2$ and K is the total number of types for entity e. For each dataset, we transfer the type information from the other two datasets individually. Given the transferred type information, we collect the relations whose head type set and tail type set are both non-empty in order to perform valid type information encoding. Only the testing triples that contain such relations are considered for the evaluation. In the end, on the FB15K dataset, 785 qualified relations are obtained resulting in $55,804$ testing triples. On the YAGO26K-906 dataset, 32 qualified relations are obtained resulting in $37,401$ testing triples. On the DB111K-27 dataset, 134 qualified relations are obtained resulting in $55,037$ testing triples. For comparison, we extract the type information from within-dataset. The prior models with type information extracted from FB15K, YAGO26K-906, and DB111K-174 are denoted as FB prior, YG prior, and DB prior respectively. In addition, for each dataset, we consider a union prior where we combine the type sets of each entity extracted from the other two datasets with the existing type set collected from within-dataset. The embedding based models are directly trained on the training triples from within-dataset. RotatE is applied as the baseline embedding based model. For each dataset, we combine the baseline embedding based model with four different prior models individually, resulting in four TaRP models: TaRP-R(FB prior), TaRP-R(YG prior), TaRP-R(DB prior) and TaRP-R(Union prior). Results are shown in Table 7. As we can see from the Table 7, TaRP-R with type information collected from cross-datasets can still achieve performance improvement compared to the baseline embedding based model. For example, on FB15K, 3.3% improvement is achieved with the type information transferred from YAGO26K-906. Furthermore, the TaRP-R with union-prior achieves better performance than the TaRP-R with type information collected within-datasets by leveraging additional type information collected from cross-datasets. From these results, we can see that the type information extracted from a specific dataset can generalize well to different datasets. In addition, through the proposed decision-level integration, the embedding based model can be easily combined with different type information without additional training.

Comparisons to State-Of-The-Art Methods

We compare TaRP to additional SoTA models that also apply ontological information. In particular, on FB15K, we

Method	FB15K			YAGO26K-906			DB111K-174		
	MR	Hits@1	Hits@10	MR	Hits@1	Hits@10	MR	Hits@1	Hits@10
RotatE	1.76	82.47	98.59	1.09	92.55	99.92	2.59	79.42	97.21
TaRP-R(FB prior)	1.37	92.88	99.79	1.06	94.29	99.98	2.06	80.55	97.88
TaRP-R(YG prior)	1.74	85.77	98.97	1.05	95.75	99.99	2.22	79.90	97.62
TaRP-R(DB prior)	5.99	84.95	98.25	1.06	94.62	99.99	1.39	86.20	99.22
TaRP-R(Union prior)	1.19	92.90	99.83	1.04	95.81	99.99	1.34	86.57	99.40

Table 7: Cross-dataset evaluation of the TaRP model

compared to DKRL (Xie et al. 2016), TKRL (Xie, Liu, and Sun 2016), SSP (Xiao et al. 2017), and TransT (Ma et al. 2017). TKRL and TransT apply type information. DKRL and SSP apply contextual information like descriptions of entities. The results are shown in Table 8. * indicates the reported performance. On YAGO26K-906 and DB111K-174, we compare to the state-of-the-art model, JOIE (Hao et al. 2019). We train JOIE on two datasets with its reported best hyper-parameter configurations, and the results are shown in Table 9. From Table 8 and Table 9, we can see that TaRP-R achieves the best performance, in particular for Hits@10. By integrating type information, the ranks of triples are concentrated within rank1- rank10. Hence, TaRP-R achieves very high Hits@10 and significantly outperforms SoTA on all three datasets.

Methods	MR	Hits@1	Hits@10
DKRL(CNN)+TransE	2.03*	-	90.8*
TKRL(RHE)	1.73*	92.8*	-
SSP(Std.)	1.22*	-	89.2*
SSP(Joint)	1.47*	-	90.9*
TransT	1.19*	-	94.1*
TaRP-R	**1.16**	**92.9**	**99.8**

Table 8: Comparisons with SOTA on FB15k

Method	YAGO26K-906		
	MR	Hits@1	Hits@10
JOIE	1.47	90.1	97.1
TaRP-R	**1.08**	**92.8**	**99.9**

Method	DB111K-174		
	MR	Hits@1	Hits@10
JOIE	2.22	71.8	89.6
TaRP-R	**1.52**	**76.5**	**99.5**

Table 9: Comparisons with SOTA on YAGO26K-906 and DB111K-174

Discussion

Though majority of the related works that are aligned well with our proposed method performed evaluations on FB15K(as shown in Table 8), FB15K contains several shortcomings, such as data leakage problem. To address the potential concerns on the evaluation regarding to the problems within the FB15K, we consider the FB15K-237 (Dettmers et al. 2018), which is an improved version of FB15K. We perform two evaluations on FB15K-237: 1) compare the proposed TaRP model to the baseline models; 2) compare to the SOTA method. We firstly evaluate the effectiveness the proposed approach on FB15K-237 by comparing to baseline models. As we can see from Table 10, all

	MR	Hits@1	Hits@10
TransE	1.51	93.18	98.27
RotatE	1.88	93.89	99.18
QuatE	1.65	90.83	98.58
TaRP-T	**1.17**	**94.64**	99.76
TaRP-R	1.19	94.25	**99.79**
TaRP-Q	1.24	92.51	99.73

Table 10: Evaluation of the TaRP model on FB15K-237

three TaRP models achieve performance improvement on FB15K-237 compared to the corresponding baseline embedding based models. Particularly, MR is reduced significantly from 1.88 to 1.19 by augmenting the RotatE with type information through the proposed framework. We then compare the TaRP model to the SOTA model: HAKE (Zhang et al. 2020b)We train HAKE with its reported hyper-parameter settings. From Table 11, we can see that the TaRP-R significantly outperforms the HAKE model.

	MR	Hits@1	Hits@10
HAKE	1.85	92.85	99.13
TaRP-R	**1.19**	**94.25**	**99.79**

Table 11: Comparisons with SOTA on FB15K-237

Conclusion

In this paper, we propose an effective type-augmented relation prediction (TaRP) method, where we apply both type information and instance-level information for relation prediction in knowledge graphs. The type information and instance-level information are encoded as prior probabilities and likelihoods of relations respectively, and are combined at the decision-level. Our approach significantly outperforms state-of-the-art methods. Additionally, by leveraging type information, the TaRP model is able to be more data efficient than existing models. Furthermore, type information extracted from a specific dataset is shown to generalize well to other datasets.

Acknowledgements

This work is supported by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons). Part of this work is also supported by DARPA grant FA8750-17-2-0132.

References

Balažević, I.; Allen, C.; and Hospedales, T. M. 2019. Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*, 553–565. Springer.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Chen, W.; Xiong, W.; Yan, X.; and Wang, W. 2018. Variational knowledge graph reasoning. *arXiv preprint arXiv:1803.06581* .

Das, R.; Neelakantan, A.; Belanger, D.; and McCallum, A. 2016. Chains of reasoning over entities, relations, and text using recurrent neural networks. *arXiv preprint arXiv:1607.01426* .

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Garg, D.; Ikbal, S.; Srivastava, S. K.; Vishwakarma, H.; Karanam, H.; and Subramaniam, L. V. 2019. Quantum embedding of knowledge for reasoning. In *Advances in Neural Information Processing Systems*, 5594–5604.

Guo, S.; Wang, Q.; Wang, B.; Wang, L.; and Guo, L. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 84–94.

Hao, J.; Chen, M.; Yu, W.; Sun, Y.; and Wang, W. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1709–1719.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 105–113.

Jain, P.; Kumar, P.; Chakrabarti, S.; et al. 2018. Type-sensitive knowledge base inference without explicit type supervision. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 75–80.

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 687–696.

Kapanipathi, P.; Thost, V.; Patel, S. S.; Whitehead, S.; Abdelaziz, I.; Balakrishnan, A.; Chang, M.; Fadnis, K.; Gunasekara, C.; Makni, B.; Mattei, N.; Talamadupula, K.; and Fokoue, A. 2020. Infusing Knowledge into the Textual Entailment Task Using Graph Convolutional Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* .

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 529–539. Association for Computational Linguistics.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic web* 6(2): 167–195.

Lei, K.; Zhang, J.; Xie, Y.; Wen, D.; Chen, D.; Yang, M.; and Shen, Y. 2019. Path-based reasoning with constrained type attention for knowledge graph completion. *Neural Computing and Applications* 1–10.

Ma, S.; Ding, J.; Jia, W.; Wang, K.; and Guo, M. 2017. Transt: Type-based multiple embedding representations for knowledge graph completion. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 717–733. Springer.

Rebele, T.; Suchanek, F.; Hoffart, J.; Biega, J.; Kuzey, E.; and Weikum, G. 2016. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International Semantic Web Conference*, 177–185. Springer.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, 926–934.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197* .

Wang, X.; He, X.; Cao, Y.; Liu, M.; and Chua, T.-S. 2019a. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 950–958.

Wang, X.; Kapanipathi, P.; Musa, R.; Yu, M.; Talamadupula, K.; Abdelaziz, I.; Chang, M.; Fokoue, A.; Makni, B.; Mattei, N.; and Witbrock, M. 2019b. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. *Proceedings of the AAAI Conference on Artificial Intelligence* 33: 7208–7215. ISSN 2159-5399. doi:10.1609/aaai.v33i01.33017208.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Aaai*. Citeseer.

West, R.; Gabrilovich, E.; Murphy, K.; Sun, S.; Gupta, R.; and Lin, D. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web*, 515–526.

Xiao, H.; Huang, M.; Meng, L.; and Zhu, X. 2017. SSP: semantic space projection for knowledge graph embedding with text descriptions. In *Thirty-First AAAI conference on artificial intelligence*.

Xie, R.; Liu, Z.; Jia, J.; Luan, H.; and Sun, M. 2016. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Xie, R.; Liu, Z.; and Sun, M. 2016. Representation Learning of Knowledge Graphs with Hierarchical Types. In *IJCAI*, 2965–2971.

Xu, H.; Bao, J.; and Zhang, G. 2020. Dynamic Knowledge Graph-based Dialogue Generation with Improved Adversarial Meta-Learning. *arXiv preprint arXiv:2004.08833* .

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* .

Zhang, L.; Yu, M.; Gao, T.; and Yu, Y. 2020a. MCMH: Learning Multi-Chain Multi-Hop Rules for Knowledge Graph Reasoning. *Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* .

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, 2731–2741.

Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020b. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 3065–3072. AAAI Press.