Learning Representations of Bi-level Knowledge Graphs for Reasoning beyond Link Prediction

Chanyoung Chung, Joyce Jiyoung Whang*
School of Computing, KAIST
{chanyoung.chung, jjwhang}@kaist.ac.kr

Abstract

Knowledge graphs represent known facts using triplets. While existing knowledge graph embedding methods only consider the connections between entities, we propose considering the relationships between triplets. For example, let us consider two triplets T_1 and T_2 where T_1 is (Academy_Awards, Nominates, Avatar) and T_2 is (Avatar, Wins, Academy_Awards). Given these two base-level triplets, we see that T_1 is a prerequisite for T_2. In this paper, we define a higher-level triplet to represent a relationship between triplets, e.g., $\langle T_1$, PrerequisiteFor, $T_2\rangle$ where PrerequisiteFor is a higher-level relation. We define a bi-level knowledge graph that consists of the base-level and the higher-level triplets. We also propose a data augmentation strategy based on the random walks on the bi-level knowledge graph to augment plausible triplets. Our model called BiVE learns embeddings by taking into account the structures of the base-level and the higher-level triplets, with additional consideration of the augmented triplets. We propose two new tasks: triplet prediction and conditional link prediction. Given a triplet T_1 and a higher-level relation, the triplet prediction predicts a triplet that is likely to be connected to T_1 by the higher-level relation, e.g., $\langle T_1$, PrerequisiteFor, ?$\rangle$. The conditional link prediction predicts a missing entity in a triplet conditioned on another triplet, e.g., $\langle T_1$, PrerequisiteFor, (Avatar, Wins, ?)$\rangle$. Experimental results show that BiVE significantly outperforms all other methods in the two new tasks and the typical base-level link prediction in real-world bi-level knowledge graphs.

Introduction

A knowledge graph represents the relationships between entities using triplets consisting of a head entity, a relation, and a tail entity. Knowledge graph embedding aims to represent the entities and relations as a set of embedding vectors that can be utilized in many modern AI applications (Ji et al. 2022; Kwak et al. 2022). Most existing knowledge graph embedding methods generate the embedding vectors by focusing solely on how the entities are connected by the relations (Wang et al. 2017; Chung and Whang 2021; Chami et al. 2020). Even though some methods predict missing connections between the entities by rule mining (Meilicke et al. 2019; Sadeghian et al. 2018) or rule-and-path-based learning (Niu et al. 2020), these existing approaches only enable expanding the entity-level connections.

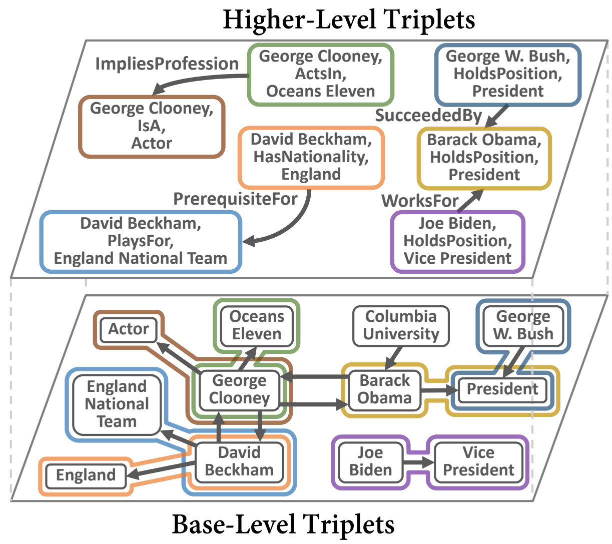

Figure 1: Example of a bi-level knowledge graph consisting of base-level and higher-level triplets in the *FBHE* dataset. The relation labels are omitted in the base-level triplets.

Each triplet in a knowledge graph can have a relationship with another triplet. For example, let us consider two base-level triplets T_1 and T_2 where T_1 is (Joe_Biden, HoldsPosition, Vice_President) and T_2 is (Barack_Obama, HoldsPosition, President). To represent the fact that Joe Biden was a vice president when Barack Obama was a president, we define a higher-level triplet $\langle T_1$, WorksFor, $T_2\rangle$ where WorksFor is a higher-level relation. In this paper, we define a bi-level knowledge graph that includes both the base-level and the higher-level triplets, where the base-level triplets correspond to the original triplets representing the relationships between entities, while the higher-level triplets represent the relationships between the base-level triplets using the higher-level relations. Based on well-known knowledge graphs, *FB15K237* (Toutanova and Chen 2015) and *DB15K* (Garcia-Duran and Niepert 2018), we create three real-world bi-level knowledge graphs named *FBH*, *FBHE*, and *DBHE*. Figure 1 shows a subgraph of a bi-level knowl-

*Corresponding author.

edge graph in *FBHE* where the base-level triplets correspond to the original triplets in *FB15K237* and the higher-level triplets are manually created by defining the higher-level relationships between the base-level triplets.

We propose incorporating the base-level and the higher-level triplets into knowledge graph embedding. Using the bi-level knowledge graphs, we also propose a data augmentation strategy that augments triplets by identifying plausible relation sequences based on random walks. We develop a new knowledge graph embedding method called BiVE (embedding of **Bi**-le**V**el knowledg**E** graphs) that computes embedding vectors by reflecting the structures of the base-level and the higher-level triplets simultaneously, where the augmented triplets are further incorporated. Using the bi-level knowledge graphs, we propose two new tasks: triplet prediction and conditional link prediction. The triplet prediction predicts a triplet that is likely to be connected to a given triplet using a higher-level relation, e.g., $\langle T_1$, WorksFor, ?$\rangle$, whereas the conditional link prediction predicts a missing entity in a triplet where another triplet is provided as a condition, e.g., $\langle T_1$, WorksFor, (?, HoldsPosition, President)$\rangle$. Experimental results show that BiVE significantly outperforms other state-of-the-art knowledge graph completion methods in real-world datasets. Our datasets and codes are available at https://github.com/bdi-lab/BiVE.

Our contributions can be summarized as follows:

- To the best of our knowledge, our work is the first work that introduces the higher-level relationships between triplets in knowledge graphs; we define bi-level knowledge graphs and create three real-world datasets.
- We propose an efficient data augmentation strategy using random walks on a bi-level knowledge graph.
- We develop BiVE to learn embeddings by effectively incorporating the base-level triplets, the higher-level triplets, and the augmented triplets.
- We propose two new tasks, triplet prediction and conditional link prediction, which have never been studied.
- BiVE significantly outperforms 12 different state-of-the-art knowledge graph completion methods.

Related Work

Some knowledge graph completion methods use multi-hop paths between distant entities (Niu et al. 2020; Lin et al. 2015; Jiang et al. 2020; Lin, Socher, and Xiong 2018; Das et al. 2018) and rule-based or logic-based methods identify frequently observed patterns (Meilicke et al. 2019; Demeester, Rocktäschel, and Riedel 2016; Guo et al. 2016; Yang, Yang, and Cohen 2017; Sadeghian et al. 2018; Nayyeri et al. 2021). The main difference between these methods and BiVE is that the existing methods only consider the relationships between entities, whereas BiVE considers not only the relationships between entities but also the relationships between triplets. Also, the way of expressing the relationships between entities or triplets in BiVE is not restricted to the first-order-logic-like expression. For example, the rule-based methods consider the relationships between connected entities, e.g., $\forall x, y, z : (x, r_1, y) \wedge (y, r_2, z) \Rightarrow (x, r_3, z)$ where there should exist a path connecting x, y, and z in the knowledge graph. On the other hand, BiVE represents relationships like $(x, r_1, y) \overset{\widehat{r}}{\Rightarrow} (p, r_2, q)$ where x, y and p, q are not necessarily connected by the base-level triplets, and also r_1, r_2, and $\overset{\widehat{r}}{\Rightarrow}$ can be any relation not restricted to the first-order-logic-like relation.

Even though there have been many attempts to discover meaningful patterns in a knowledge graph and utilize them to complete missing links (Lao and Cohen 2010), such attempts have rarely been studied in the context of data augmentation. Recently, rule-based data augmentation for knowledge graph embedding has been proposed (Li et al. 2021)[1]. While this method uses logical rules using the base-level triplets, our data augmentation employs random walks on a bi-level knowledge graph.

To exploit enriched information about triplets, some knowledge graph embedding methods utilize attributes of entities (Wu and Wang 2018; Kristiadi et al. 2019) or ontological concepts (Hao et al. 2019). HINGE (Rosso, Yang, and Cudré-Mauroux 2020) has been proposed to represent hyper-relational facts where a triplet has additional key-value pairs to present extra information about each triplet. Even though these methods consider enriched information about triplets, they do not consider the relationships between triplets.

In information retrieval, a neural fact contextualization method has been proposed to rank a set of candidate facts for a given triplet (Voskarides et al. 2018). Also, a way of representing a triplet in an embedding space is studied by considering the concept of a line graph (Fionda and Pirrò 2020). Recently, ATOMIC (Sap et al. 2019) has been proposed to provide commonsense knowledge for if-then reasoning, whereas ASER (Zhang et al. 2020) has been proposed to construct an eventuality knowledge graph. Although these methods consider triplet-level operations, the goal of their methods is different from ours and none of these considers the bi-level knowledge graphs.

Bi-Level Knowledge Graphs

Let us represent a knowledge graph as $G = (\mathcal{V}, \mathcal{R}, \mathcal{E})$ where $\mathcal{V}$ is a set of entities, $\mathcal{R}$ is a set of relations, and $\mathcal{E} = \{(h, r, t) : h \in \mathcal{V}, r \in \mathcal{R}, t \in \mathcal{V}\}$ is a set of triplets. Let us call G a base-level knowledge graph and call $(h, r, t) \in \mathcal{E}$ a base-level triplet. We formally define the higher-level triplets as follows.

Definition 1 (Higher-Level Triplets) *Given a base-level knowledge graph $G = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, a set of higher-level triplets is defined by $\mathcal{H} = \{\langle T_i, \widehat{r}, T_j \rangle : T_i \in \mathcal{E}, \widehat{r} \in \widehat{\mathcal{R}}, T_j \in \mathcal{E}\}$ where $\mathcal{E}$ is a set of base-level triplets and $\widehat{\mathcal{R}}$ is a set of higher-level relations connecting the base-level triplets.*

We define a bi-level knowledge graph as follows.

Definition 2 (Bi-Level Knowledge Graph) *Given a base-level knowledge graph $G = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, a set of higher-level*

[1]We could not include this method as a baseline in our experiments because the authors of (Li et al. 2021) could not provide their source codes due to some confidentiality restrictions.

$\widehat{r}$	$\langle T_i, \widehat{r}, T_j \rangle$
PrerequisiteFor	T_i: (BAFTA, Nominates, The_King's_Speech) T_j: (The_King's_Speech, Wins, BAFTA)
WorksFor	T_i: (Joe_Biden, HoldsPosition, Vice_President) T_j: (Barack_Obama, HoldsPosition, President)
ImpliesTimeZone	T_i: (Czech, TimeZone, Central_European) T_j: (Prague, TimeZone, Central_European)
NextAlmaMater	T_i: (Ford, StudiesIn, University_of_Michigan) T_j: (Ford, StudiesIn, Yale_University)

Table 1: Examples of Higher-Level Relations and Triplets. The first two examples are from *FBHE*, and the last two examples are from *DBHE*.

relations $\widehat{\mathcal{R}}$, and a set of higher-level triplets $\mathcal{H}$, a bi-level knowledge graph is defined as $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \widehat{\mathcal{R}}, \mathcal{H})$.

To define a bi-level knowledge graph, we add the higher-level triplets $\mathcal{H}$ to the base-level knowledge graph G by introducing the higher-level relations $\widehat{\mathcal{R}}$. We create real-world bi-level knowledge graphs *FBH* and *FBHE* based on *FB15K237* from *Freebase* (Bollacker et al. 2008) and *DBHE* based on *DB15K* from *DBpedia* (Auer et al. 2007). Table 1 shows some examples of the higher-level relations and triplets. *FBH* contains the higher-level relations that can be inferred inside the base-level knowledge graph, e.g., PrerequisiteFor, whereas *FBHE* and *DBHE* contain some externally-sourced knowledge, e.g., WorksFor and NextAlmaMater. For example, we crawl Wikipedia articles to find information about the (vice)presidents of the United States and the alma mater information of politicians. Note that the base-level knowledge graphs of *FBH* and *FBHE* are *FB15K237*. *FBHE* extends *FBH* by including the externally-sourced higher-level relationships. The authors of this paper manually defined the higher-level relations and added the higher-level triplets to *FB15K237* and *DB15K*, which took six weeks.

Using a bi-level knowledge graph, we define the triplet prediction problem as follows.

Definition 3 (Triplet Prediction) *Given a bi-level knowledge graph $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \widehat{\mathcal{R}}, \mathcal{H})$ where $\mathcal{H} = \{\langle T_i, \widehat{r}, T_j \rangle : T_i \in \mathcal{E}, \widehat{r} \in \widehat{\mathcal{R}}, T_j \in \mathcal{E}\}$, the triplet prediction problem is defined as $\langle T_i, \widehat{r}, ? \rangle$ or $\langle ?, \widehat{r}, T_j \rangle$ where the goal is to predict the missing base-level triplet.*

Also, we define the conditional link prediction as follows.

Definition 4 (Conditional Link Prediction) *Given a bi-level knowledge graph $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \widehat{\mathcal{R}}, \mathcal{H})$ where $\mathcal{H} = \{\langle T_i, \widehat{r}, T_j \rangle : T_i \in \mathcal{E}, \widehat{r} \in \widehat{\mathcal{R}}, T_j \in \mathcal{E}\}$, let $T_i := (h_i, r_i, t_i)$ and $T_j := (h_j, r_j, t_j)$. The conditional link prediction problem is to predict a missing entity in a base-level triplet conditioned on another base-level triplet. Specifically, the problem is defined as $\langle T_i, \widehat{r}, (h_j, r_j, ?) \rangle$ or $\langle T_i, \widehat{r}, (?, r_j, t_j) \rangle$ or $\langle (h_i, r_i, ?), \widehat{r}, T_j \rangle$ or $\langle (?, r_i, t_i), \widehat{r}, T_j \rangle$.*

Data Augmentation by Random Walks on a Bi-Level Knowledge Graph

Consider a bi-level knowledge graph in the training set $\widehat{G}_{\text{train}} = (\mathcal{V}, \mathcal{R}, \mathcal{E}_{\text{train}}, \widehat{\mathcal{R}}, \mathcal{H}_{\text{train}})$ where $\mathcal{E}_{\text{train}}$ and $\mathcal{H}_{\text{train}}$ are the base-level and the higher-level triplets in the training set respectively. We add reverse relations to $\mathcal{R}$ and add reversed triplets to $\mathcal{E}_{\text{train}}$, i.e., for every $r \in \mathcal{R}$, we add r^{-1} that has the reverse direction of r and for every $(h, r, t) \in \mathcal{E}_{\text{train}}$, we add (t, r^{-1}, h) to $\mathcal{E}_{\text{train}}$. Similarly, for every $\widehat{r} \in \widehat{\mathcal{R}}$, we add $\widehat{r}^{-1}$ and add the reversed higher-level triplets to $\mathcal{H}_{\text{train}}$. All these reverse relations and reversed triplets are added only for data augmentation.

From an entity h, we randomly visit one of its neighbors by following a base-level or a higher-level triplet. To search for diverse patterns, we do not allow going back to an entity that has already been visited. Let us define a random walk path to be the sequence of visited entities, visited relations, and visited higher-level relations. Consider two base-level triplets $T_i = (h_i, r_i, t_i)$ and $T_j = (h_j, r_j, t_j)$ and a higher-level triplet $\langle T_i, \widehat{r}, T_j \rangle$. From any entity in T_i, we can go to any entity in T_j and vice versa by following r_i, $\widehat{r}$, and r_j or their reverse relations. For example, one possible random walk path is $(h_i, r_i, \widehat{r}, r_j, t_j)$. Another possible random walk path is $(t_j, r_j^{-1}, \widehat{r}^{-1}, r_i, t_i)$. Assume that we have a base-level triplet $T_0 = (h_0, r_0, h_i)$. Starting from h_0, we can make a longer path, e.g., $(h_0, r_0, h_i, r_i, \widehat{r}, r_j, t_j)$. We define the length of a random walk path to be the number of entities in the sequence except the starting entity.

Given the maximum length of a random walk path L, we repeat the random walks by varying the length $l = 2, \cdots, L$ and repeat the random walks n times for every l. In our experiments, we set L=3 and n=50,000,000. Let w denote the sequence of a random walk path of all possible lengths, where we randomly select a starting entity for every w. If there are multiple identical random walk paths, we remove the duplicates to prevent unexpected bias. Let p_k be the k-th unique sequence of relations and higher-level relations extracted from w, i.e., we make p_k by removing all entities from w, e.g., if $w = (h_0, r_0, h_i, r_i, \widehat{r}, r_j, t_j)$ then $p_k = (r_0, r_i, \widehat{r}, r_j)$. We call p_k the relation sequence. Since p_k only traces the relations, different random walk paths can be mapped into the same p_k. Using p_k, we rewrite a random walk path $w = (h, \cdots, t)$ to $w = (h, p_k, t)$ where the relation sequence of the original path w is mapped into p_k, h is the starting entity and t is the last entity. Let $\mathcal{W}$ denote the multiset of all random walk paths of all possible lengths. We define the confidence score of (p_k, r) as

$$c(p_k, r) := \frac{|\{(h, r, t) : (h, p_k, t) \in \mathcal{W}, (h, r, t) \in \mathcal{E}_{\text{train}}\}|}{|\{(h, p_k, t) : (h, p_k, t) \in \mathcal{W}\}|}.$$

We select the pairs of (p_k, r) that satisfies $c(p_k, r) \geq \tau$ where we set $\tau = 0.7$. Let $\mathcal{S}_{kr} := \{(h, r, t) : (h, p_k, t) \in \mathcal{W}, c(p_k, r) \geq \tau, (h, r, t) \notin \mathcal{E}_{\text{train}}\}$ where $\mathcal{S}_{kr}$ indicates a set of missing triplets (h, r, t) even though $c(p_k, r) \geq \tau$. Then, let $\mathcal{S} := \cup_k \cup_r \mathcal{S}_{kr}$ where $\mathcal{S}$ is a set of augmented triplets. We add the triplets in $\mathcal{S}$ to a bi-level knowledge graph to augment triplets that are likely to be present. Figure 2 shows an example of a random walk path of length

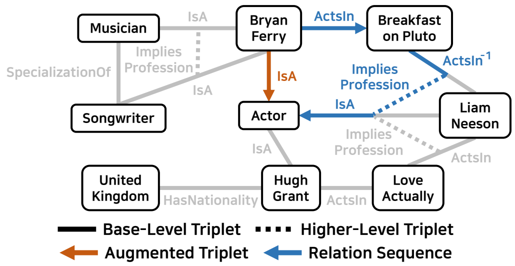

Figure 2: Random walk path in a bi-level knowledge graph and an augmented triplet in *FBH*. We add missing triplets whose confidence scores are greater than a certain threshold.

2 and an augmented triplet in *FBH*, where the walk starts from Bryan_Ferry. Let $p_1 =$ (ActsIn, ActsIn^{-1}, ImpliesProfession, IsA). Since the confidence score of (p_1, IsA) is 0.99, we add a triplet (Bryan_Ferry, IsA, Actor) which was missing in the original training set.

Embedding of Bi-Level Knowledge Graphs

A knowledge graph embedding method defines a scoring function $f(\mathbf{h}, \mathbf{r}, \mathbf{t})$ of a triplet (h, r, t), where $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ are embedding vectors of h, r, and t respectively; a higher score indicates a more plausible triplet. In BiVE, the loss incurred by the base-level triplets, L_{base}, is defined as follows:

$$L_{\text{base}} := \sum_{(h,r,t)\in\mathcal{E}_{\text{train}}} g(-f(\mathbf{h},\mathbf{r},\mathbf{t})) + \sum_{(h',r',t')\in\mathcal{E}'_{\text{train}}} g(f(\mathbf{h}',\mathbf{r}',\mathbf{t}'))$$

where $g(x) = \log(1 + \exp(x))$ and $\mathcal{E}'_{\text{train}}$ is a set of corrupted triplets. We can use any knowledge graph embedding scoring function for $f(\cdot)$. We implement BiVE with two different scoring functions for $f(\cdot)$: QuatE (Zhang et al. 2019) for BiVE-Q and BiQUE (Guo and Kok 2021) for BiVE-B.

Given $T_i = (h_i, r_i, t_i)$, let $\mathbf{T}_i$ denote an embedding vector of T_i where the dimension is $\hat{d}$. We define $\mathbf{T}_i := \boldsymbol{W}[\mathbf{h}_i; \mathbf{r}_i; \mathbf{t}_i]$ where $\mathbf{h}_i$, $\mathbf{r}_i$, and $\mathbf{t}_i$ denote the embedding vectors of h_i, r_i, and t_i respectively, the dimension of each of these embedding vectors is d, and $\boldsymbol{W}$ is a projection matrix of size $\hat{d} \times 3d$ which projects the vertically concatenated vector to the $\hat{d}$-dimensional space. Similarly, $\mathbf{T}_j = \boldsymbol{W}[\mathbf{h}_j; \mathbf{r}_j; \mathbf{t}_j]$ where $T_j = (h_j, r_j, t_j)$. We define the loss incurred by the higher-level triplets, L_{high}, as follows:

$$L_{\text{high}} := \sum_{\langle T_i,\widehat{r},T_j\rangle} g(-f(\mathbf{T}_i,\widehat{\mathbf{r}},\mathbf{T}_j)) + \sum_{\langle T_i{}',\widehat{r}',T_j{}'\rangle} g(f(\mathbf{T}'_i,\widehat{\mathbf{r}}',\mathbf{T}'_j))$$

where $\langle T_i, \widehat{r}, T_j\rangle \in \mathcal{H}_{\text{train}}$, $\langle T_i{}', \widehat{r}', T_j{}'\rangle \in \mathcal{H}'_{\text{train}}$, $\widehat{\mathbf{r}}$ is the embedding vector of $\widehat{r}$, the dimension of $\widehat{\mathbf{r}}$ is $\hat{d}$, and $\langle T_i{}', \widehat{r}', T_j{}'\rangle$ is a corrupted higher-level triplet made by randomly replacing T_i or T_j with one of the triplets in $\mathcal{E}_{\text{train}}$.

We define the loss of the augmented triplets, L_{aug}, as

$$L_{\text{aug}} := \sum_{(h,r,t)\in\mathcal{S}} g(-f(\mathbf{h},\mathbf{r},\mathbf{t})) + \sum_{(h',r',t')\in\mathcal{S}'} g(f(\mathbf{h}',\mathbf{r}',\mathbf{t}'))$$

where $\mathcal{S}'$ is the set of corrupted triplets.

Finally, our loss function of BiVE is defined by

$$L_{\text{BiVE}} := L_{\text{base}} + \lambda_1 \cdot L_{\text{high}} + \lambda_2 \cdot L_{\text{aug}}$$

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\widehat{\mathcal{R}}\rvert$	$\lvert\mathcal{H}\rvert$	$\lvert\widehat{\mathcal{E}}\rvert$
FBH	14,541	237	310,117	6	27,062	33,157
FBHE	14,541	237	310,117	10	34,941	33,719
DBHE	12,440	87	68,296	8	6,717	8,206

Table 2: Statistics of a bi-level knowledge graph $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \widehat{\mathcal{R}}, \mathcal{H})$. $\lvert\widehat{\mathcal{E}}\rvert$ is the number of base-level triplets which are involved in the higher-level triplets.

where λ_1 is a hyperparameter indicating the importance of the higher-level triplets and λ_2 indicates the importance of the augmented triplets. By optimizing L_{BiVE}, BiVE learns embeddings by considering the structures of the base-level triplets, the higher-level triplets, and the augmented triplets.

Let us describe the scoring functions of BiVE for triplet prediction and conditional link prediction. To solve a triplet prediction problem, $\langle T_i, \widehat{r}, ?\rangle$, we compute $F_{\text{tp}}(X) := f(\mathbf{T}_i, \widehat{\mathbf{r}}, \mathbf{X})$ for every base-level triplet $X \in \mathcal{E}_{\text{train}}$ where $\mathbf{X}$ is a learned embedding vector of X. To solve a conditional link prediction problem, $\langle T_i, \widehat{r}, (h_j, r_j, ?)\rangle$, we compute $F_{\text{clp}}(x) := f(\mathbf{h}_j, \mathbf{r}_j, \mathbf{x}) + \lambda_1 \cdot f(\mathbf{T}_i, \widehat{\mathbf{r}}, \boldsymbol{W}[\mathbf{h}_j; \mathbf{r}_j; \mathbf{x}])$ for every $x \in \mathcal{V}$ where $\mathbf{x}$ is a learned embedding of x.

Experimental Results

We use three real-world bi-level knowledge graphs presented in Table 2, where $\lvert\widehat{\mathcal{E}}\rvert$ is the number of base-level triplets involved in the higher-level triplets. We split $\mathcal{E}$ and $\mathcal{H}$ into training, validation, and test sets with a ratio of 8:1:1. We use three standard evaluation metrics: the filtered MR (Mean Rank), MRR (Mean Reciprocal Rank), and Hit@10 (Wang et al. 2017). Higher MRR and Hit@10 and a lower MR indicate better results. We repeat experiments ten times for each method and report the average of each metric. We set $d = 200$ and $\hat{d} = 200$. We use 12 different baseline methods: ASER (Zhang et al. 2020), MINERVA (Das et al. 2018), Multi-Hop (Lin, Socher, and Xiong 2018), Neural-LP (Yang, Yang, and Cohen 2017), DRUM (Sadeghian et al. 2018), AnyBURL (Meilicke et al. 2019), PTransE (Lin et al. 2015), RPJE (Niu et al. 2020), TransD (Ji et al. 2015), ANALOGY (Liu, Wu, and Yang 2017), QuatE (Zhang et al. 2019) and BiQUE (Guo and Kok 2021). For TransD and ANALOGY, we use the implementations in OpenKE (Han et al. 2018). More details of datasets and methods are described in the Supplementary Material.[2]

Triplet Prediction

While BiVE solves a triplet prediction problem using the scoring function $F_{\text{tp}}(X)$, none of the baseline methods can deal with the higher-level triplets. To feed the higher-level triplets to the baseline methods, we create a new knowledge graph G_T where a base-level triplet is converted into an entity and a higher-level triplet is converted into a triplet. Let $T_i = (h_i, r_i, t_i) \in \mathcal{E}_{\text{train}}$ denote a base-level triplet. We define $G_T := (\mathcal{E}_{\text{train}}, \widehat{\mathcal{R}}, \mathcal{H}_{\text{train}})$, where each T_i is considered

[2]More details are available at https://bdi-lab.kaist.ac.kr.

		FBH			*FBHE*			*DBHE*	
	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)
ASER	74541.6	0.011	0.015	57916.0	0.050	0.070	18157.6	0.042	0.075
MINERVA	109055.1	0.093	0.113	85571.5	0.220	0.300	20764.3	0.177	0.221
Multi-Hop	108731.7	0.105	0.117	83643.8	0.255	0.311	20505.8	0.191	0.230
Neural-LP	115016.6	0.070	0.073	90000.4	0.238	0.274	21130.5	0.170	0.209
DRUM	115016.6	0.069	0.073	90000.3	0.261	0.274	21130.5	0.166	0.209
AnyBURL	108079.6	0.096	0.108	83136.8	0.191	0.252	20530.8	0.177	0.214
PTransE	111024.3	0.069	0.071	86793.2	0.249	0.274	18888.7	0.158	0.195
RPJE	113082.0	0.070	0.072	89173.1	0.267	0.274	20290.4	0.166	0.206
TransD	74277.3	0.052	0.104	52159.4	0.238	0.280	16698.1	0.116	0.189
ANALOGY	93383.4	0.072	0.107	60161.5	0.286	0.318	18880.0	0.150	0.211
QuatE	145603.8	0.103	0.114	94684.4	0.101	0.209	26485.0	0.157	0.179
BiQUE	81687.5	0.104	0.115	61015.2	0.135	0.205	19079.4	0.163	0.185
BiVE-Q	**18.7**	**0.748**	**0.853**	<u>33.1</u>	<u>0.531</u>	<u>0.683</u>	<u>56.6</u>	<u>0.315</u>	<u>0.523</u>
BiVE-B	<u>19.7</u>	<u>0.731</u>	<u>0.837</u>	**27.9**	**0.555**	**0.718**	**4.7**	**0.644**	**0.914**

Table 3: Results of Triplet Prediction. The best scores are boldfaced and the second best scores are underlined. Our models, BiVE-Q and BiVE-B, significantly outperform all other baseline methods in terms of all metrics on all datasets.

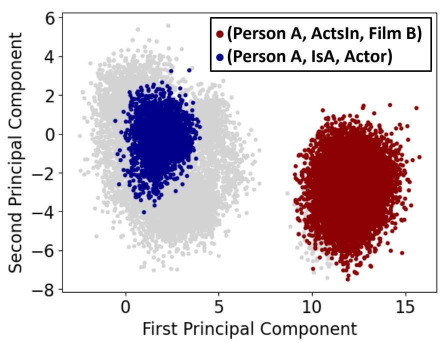

Figure 3: Embedding vectors of base-level triplets in $\langle T_i$, ImpliesProfession, $T_j\rangle$ where T_i is (Person A, ActsIn, Film B) and T_j is (Person A, IsA, Actor) in *FBH*. Both $\mathbf{T}_i$ and $\mathbf{T}_j$ embedding vectors from BiVE are well-clustered.

as an entity. If we train the baseline methods using G_T, the triplet prediction task can be considered as a link prediction problem on G_T. However, in this case, it is not guaranteed that all T_i involved in the triplets in $\mathcal{H}_{\text{test}}$ appear in $\mathcal{H}_{\text{train}}$ because we randomly split $\mathcal{H}$ into training, validation, and test sets. Therefore, for the baseline methods, the problem becomes an inductive setting instead of a transductive setting. Indeed, among the baseline methods, Neural-LP and DRUM are inductive methods and we include these methods because they can conduct inductive inference. We assume that the candidates of a triplet prediction problem should be included in the training set of the base-level knowledge graph, which aligns with a realistic setting. By taking into account both the base-level knowledge graph and the higher-level triplets simultaneously, the problem becomes a transductive setting for BiVE. This shows that simply converting the higher-level triplets into G_T cannot replace our model.

Table 3 shows the results of triplet prediction. We see that BiVE-Q and BiVE-B significantly outperform all other state-of-the-art baseline methods in terms of all the three metrics on all three real-world datasets. Note that the number of candidates of a triplet prediction problem is equal to the number of base-level triplets in $\mathcal{E}_{\text{train}}$. We visualize the embedding vectors generated by BiVE-Q on *FBH* in Figure 3. We take all higher-level triplets in the form of $\langle T_i$, ImpliesProfession, $T_j\rangle$ and visualize the embedding vectors of T_i and T_j using Principal Component Analysis. In Figure 3, we only highlight the base-level triplets T_i and T_j where T_i is (Person A, ActsIn, Film B) and T_j is (Person A, IsA, Actor). We see that both $\mathbf{T}_i$ and $\mathbf{T}_j$ embedding vectors are well-clustered, meaning that BiVE generates embeddings by appropriately reflecting the structure of the higher-level triplets.

Conditional Link Prediction

To solve a conditional link prediction problem, BiVE uses the scoring function $F_{\text{clp}}(x)$. On the other hand, the baseline methods cannot directly solve the conditional link prediction problem. To allow the baseline methods to solve $\langle T_i, \widehat{r}, (h_j, r_j, ?)\rangle$[3], we define a scoring function of the baseline methods as follows: $F(x) := f(\mathbf{h}_j, \mathbf{r}_j, \mathbf{x}) + f(\mathbf{T}_i, \widehat{\mathbf{r}}, z(h_j, r_j, x))$ for all $x \in \mathcal{V}$ where the former is computed on the original base-level knowledge graph, the latter is computed on G_T, $z(h_j, r_j, x)$ returns an embedding vector of (h_j, r_j, x), and $f(\cdot)$ is the scoring function of each baseline method. We cannot get $f(\mathbf{T}_i, \widehat{\mathbf{r}}, z(h_j, r_j, x))$ if $(h_j, r_j, x) \notin \mathcal{E}_{\text{train}}$. In that case, we compute the score using the randomly initialized vectors in PTransE, RPJE, TransD, ANALOGY, QuatE, and BiQUE, whereas we set $f(\mathbf{T}_i, \widehat{\mathbf{r}}, z(h_j, r_j, x)) = 0$ for the other baseline methods by considering the mechanisms of how each of the baseline methods assigns scores. In Table 4, BiVE-Q and BiVE-B significantly outperform all other baseline methods in conditional link prediction on all three real-world datasets. In Table 5, we show some example problems of conditional link prediction in *FBHE* and the predictions made by BiVE-Q where it correctly predicts the answers. When we consider

[3]We consider all four problems by changing the position of ?.

	FBH			*FBHE*			*DBHE*		
	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)
ASER	1183.9	0.251	0.316	970.7	0.289	0.382	1893.5	0.225	0.348
MINERVA	3817.8	0.328	0.415	3018.5	0.407	0.492	2934.1	0.362	0.433
Multi-Hop	1878.2	0.421	0.578	1447.3	0.443	0.615	1012.3	0.442	0.652
Neural-LP	185.9	0.433	0.648	146.2	0.466	0.716	32.2	0.517	0.756
DRUM	262.7	0.394	0.555	207.6	0.413	0.620	49.0	0.470	0.732
AnyBURL	228.5	0.380	0.563	166.0	0.418	0.607	81.7	0.403	0.594
PTransE	214.8	0.440	0.686	167.0	0.516	0.752	19.3	0.505	0.780
RPJE	212.5	0.440	0.686	159.0	0.528	0.753	19.3	0.504	0.779
TransD	190.1	0.300	0.496	165.6	0.363	0.529	35.5	0.436	0.708
ANALOGY	341.0	0.182	0.291	113.3	0.409	0.581	279.1	0.140	0.253
QuatE	163.7	0.346	0.494	1546.4	0.124	0.189	551.6	0.208	0.309
BiQUE	111.0	0.423	0.641	90.1	0.387	0.617	29.5	0.378	0.677
BiVE-Q	7.0	0.752	0.906	**11.0**	**0.698**	**0.839**	12.5	0.606	0.828
BiVE-B	**6.6**	**0.762**	**0.911**	12.8	0.696	0.834	**3.2**	**0.801**	**0.958**

Table 4: Results of Conditional Link Prediction. The best scores are boldfaced and the second best scores are underlined. Our models, BiVE-Q and BiVE-B, significantly outperform all baseline methods in terms of all metrics on all datasets.

Problem	Prediction by BiVE-Q
⟨(?, HasAFriendshipWith, Kelly_Preston), EquivalentTo, (Kelly_Preston, HasAFriendshipWith, George_Clooney)⟩	George_Clooney
⟨(?, HasAFriendshipWith, Kelly_Preston), EquivalentTo, (Kelly_Preston, HasAFriendshipWith, Tom_Cruise)⟩	Tom_Cruise
⟨(Joe_Jonas, IsA, ?), ImpliesProfession, (Joe_Jonas, IsA, Actor)⟩	Voice_Actor
⟨(Joe_Jonas, IsA, ?), ImpliesProfession, (Joe_Jonas, IsA, Musician)⟩	Singer-songwriter
⟨(Bucknell_University, HasAHeadquarterIn, Pennsylvania), ImpliesLocation, (?, Contains, Bucknell_University)⟩	Pennsylvania
⟨(Bucknell_University, HasAHeadquarterIn, United_States), ImpliesLocation, (?, Contains, Bucknell_University)⟩	United_States
⟨(Saturn_Award, Nominates, Avatar), PrerequisiteFor, (Avatar, Wins, ?)⟩	Saturn_Award
⟨(Academy_Award, Nominates, Avatar), PrerequisiteFor, (Avatar, Wins, ?)⟩	Academy_Award

Table 5: Examples of Conditional Link Prediction on *FBHE*. BiVE correctly predicts the answers for all the above problems.

	FBHE			*DBHE*		
	MR	MRR	Hit@10	MR	MRR	Hit@10
ASER	1489.3	0.206	0.323	2218.8	0.102	0.197
MINERVA	3828.4	0.210	0.339	3530.7	0.201	0.297
Multi-Hop	2284.0	0.324	0.500	2489.4	0.266	0.404
Neural-LP	1942.5	0.315	0.486	2904.8	0.233	0.357
DRUM	1945.6	0.317	0.490	2904.7	0.237	0.359
AnyBURL	342.0	0.310	0.526	879.1	0.220	0.364
PTransE	2077.6	0.207	0.333	3346.0	0.186	0.277
RPJE	1754.6	0.232	0.368	2991.7	0.230	0.341
TransD	166.3	0.298	0.527	429.0	0.239	0.423
ANALOGY	227.3	0.287	0.486	621.5	0.201	0.323
QuatE	139.0	0.354	0.581	409.6	0.264	0.440
BiQUE	134.9	**0.356**	0.583	**376.6**	0.274	**0.446**
BiVE-Q	125.2	0.354	0.584	405.4	0.265	0.438
BiVE-B	**123.5**	**0.356**	**0.586**	377.3	**0.275**	0.444

Table 6: Results of Base-Level Link Prediction.

a problem in the form of $\langle T_i, \widehat{r}, (h_j, r_j, ?)\rangle$, even though we have the same problem of $(h_j, r_j, ?)$, the answer becomes different depending on T_i. This is the difference between the typical base-level link prediction and the conditional link prediction.

Base-Level Link Prediction

We present the performance of the typical base-level link prediction in Table 6. Since the base-level knowledge graphs of *FBH* and *FBHE* are identical, the performance of all baseline methods is the same on *FBH* and *FBHE*. The base-level link prediction performance of BiVE on *FBH* and *FBHE* is also very similar to each other. Overall, our BiVE models show comparable results to the baseline methods for the typical link prediction task; our BiVE models have the extra capability of dealing with the triplet prediction and conditional link prediction tasks.

Data Augmentation of BiVE

We analyze the augmented triplets that are added by our data augmentation strategy. In Table 7, we show some examples of a relation sequence p_k, a relation r, and the confidence of (p_k, r), the number of augmented triplets based on (p_k, r) denoted by $|\mathcal{S}_{kr}|$, and examples of the augmented triplets in *FBHE* and *DBHE*. According to our random walk policy, we do not allow going back to an entity that has already been visited. Thus, even though a relation and its reverse relation are consecutively appeared in a relation sequence in Table 7, it does not mean that we return back to the previous entity; instead, it means that the walk steps another entity adjacent to the corresponding relation. In Table 8, we show

Relation Sequence p_k	Relation r	$c(p_k, r)$	$\|S_{kr}\|$	Examples of the Augmented Triplets
NominatesIn, NominatesIn^{-1}, ActsIn, **ImpliesProfession**, IsA	IsA	0.86	610	(Patty_Duke, IsA, Actor)
Plays, Plays^{-1}, **ImpliesSports**$^{-1}$, HasPosition	HasPosition	0.78	295	(Bayer_04, HasPosition, Forward)
Contains, Contains^{-1}, **ImpliesLocation**$^{-1}$, HasAHeadquarterIn	Contains	0.72	81	(United_States, Contains, Charlottesville)
Program^{-1}, Program, Language	Language	0.70	148	(David_Copperfield, Language, English)
Genre, **ImpliesGenre**$^{-1}$, Genre, Genre^{-1}, **ImpliesGenre**$^{-1}$, Genre	Genre	0.78	120	(Kenny_Rogers, Genre, Pop_Rock)
IsPartOf, IsPartOf, **ImpliesLocation**, IsPartOf	IsPartOf	0.76	69	(San_Pedro, IsPartOf, California)
IsPartOf, IsPartOf^{-1}, **ImpliesLocation**$^{-1}$, IsPartOf^{-1}, TimeZone	TimeZone	0.75	122	(Brockton, TimeZone, Eastern_Time_Zone)
IsProducedBy^{-1}, IsProducedBy, **ImpliesProfession**, IsA	IsA	0.73	80	(Jim_Wilson, IsA, Film_Producer)

Table 7: Examples of the Augmented Triplets in *FBHE* and *DBHE*. The first four examples are from *FBHE*, and the last four examples are from *DBHE*. The higher-level relations are boldfaced.

	FBH	*FBHE*	*DBHE*
No. of unique (p_k, r)	340,194	349,120	149,365
No. of (p_k, r) with $c(p_k, r) \geq 0.7$	35,803	39,727	7,030
No. of augmented triplets	16,601	17,463	2,026
$\|\mathcal{S} \cap \mathcal{E}_{\text{valid}}\| + \|\mathcal{S} \cap \mathcal{E}_{\text{test}}\|$	5,237	5,380	316

Table 8: Statistics of the Augmented Triplets.

		FBH	*FBHE*	*DBHE*
TP	$L_{\text{base}} + L_{\text{high}}$	19.2	28.1	65.4
	$L_{\text{base}} + L_{\text{high}} + L_{\text{aug}}$	18.7	33.1	56.6
CLP	$L_{\text{base}} + L_{\text{high}}$	8.3	12.5	12.4
	$L_{\text{base}} + L_{\text{high}} + L_{\text{aug}}$	7.0	11.0	12.5
BLP	L_{base}	139.0	139.0	409.6
	$L_{\text{base}} + L_{\text{high}}$	138.4	138.4	408.1
	$L_{\text{base}} + L_{\text{aug}}$	124.7	125.2	404.9
	$L_{\text{base}} + L_{\text{high}} + L_{\text{aug}}$	124.7	125.2	405.4

Table 9: Ablation study of BiVE with different combinations of the loss terms. The average MR scores on triplet prediction (TP), conditional link prediction (CLP), and the base-level link prediction (BLP).

statistics of the augmented triplets. Among the diverse combinations of a relation sequence p_k and a relation r, we consider the (p_k, r) pairs whose confidence scores are greater than or equal to 0.7. It is interesting to see that there exist considerable overlaps between the set $\mathcal{S}$ of the augmented triplets and $\mathcal{E}_{\text{valid}}$ and $\mathcal{E}_{\text{test}}$, indicating that our augmented triplets include many ground-truth triplets that are missing in the training set.

Ablation Study of BiVE

In BiVE, we have three different types of loss terms: L_{base}, L_{high}, and L_{aug}. Using different combinations of these loss terms, we measure the performance of BiVE to check the importance of each loss term. Table 9 shows the average MR scores of BiVE-Q with different combinations of the loss terms in three tasks: triplet prediction (TP), conditional link prediction (CLP), and base-level link prediction (BLP). Note that TP and CLP require at least two terms, L_{base} and L_{high}. Also, Table 10 shows the performance of BiVE-Q per higher-level relation in *DBHE*, where Freq. indicates the number of higher-level triplets in $\mathcal{H}_{\text{test}}$ associated with $\widehat{r}$. Among the eight higher-level relations in *DBHE*, NextAlmaMater and TransfersTo require externally-sourced knowledge. While EquivalentTo is the easiest one, the performance on the other higher-level relations varies depending on the tasks and the metrics.

		Triplet Prediction			Conditional LP		
$\widehat{r}$	Freq.	MR	MRR	Hit10	MR	MRR	Hit10
EquivalentTo	98	17.5	0.416	0.679	2.2	0.744	0.977
ImpliesLanguage	29	35.6	0.292	0.578	18.4	0.632	0.786
ImpliesProfession	210	71.3	0.427	0.569	11.5	0.704	0.844
ImpliesLocation	163	42.2	0.219	0.463	9.4	0.502	0.816
ImpliesTimeZone	44	20.6	0.354	0.631	17.8	0.604	0.707
ImpliesGenre	84	113.8	0.177	0.345	32.6	0.408	0.681
NextAlmaMater	14	71.0	0.161	0.379	2.5	0.651	0.971
TransfersTo	29	67.0	0.140	0.374	5.7	0.527	0.537

Table 10: Performance of BiVE per higher-level relation in *DBHE*. Freq. indicates the number of higher-level triplets in $\mathcal{H}_{\text{test}}$ associated with $\widehat{r}$.

Conclusion

We define a bi-level knowledge graph by introducing the higher-level relationships between triplets. We propose BiVE, which takes into account the structures of the base-level triplets, the higher-level triplets, and the augmented triplets. Experimental results show that BiVE significantly outperforms state-of-the-art methods in terms of the two newly defined tasks: triplet prediction and conditional link prediction. We believe our method can contribute to advancing many knowledge-based applications, including conditional QA (Sun, Cohen, and Salakhutdinov 2022) and multi-hop QA (Fang et al. 2020), with a special emphasis on mixing a neural language model and structured knowledge (Yasunaga et al. 2021).

Acknowledgments

This research was supported by IITP grants funded by the Korean government MSIT 2022-0-00369, 2020-0-00153 (Penetration Security Testing of ML Model Vulnerabilities and Defense) and NRF of Korea funded by the Korean Government MSIT 2018R1A5A1059921, 2022R1A2C4001594.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference*, 722–735.

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.

Chami, I.; Wolf, A.; Juan, D.-C.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6901–6914.

Chung, C.; and Whang, J. J. 2021. Knowledge Graph Embedding via Metagraph Learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2212–2216.

Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; and McCallum, A. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *Proceedings of the 6th International Conference on Learning Representations*.

Demeester, T.; Rocktäschel, T.; and Riedel, S. 2016. Lifted Rule Injection for Relation Embeddings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1389–1399.

Fang, Y.; Sun, S.; Gan, Z.; Pillai, R.; Wang, S.; and Liu, J. 2020. Hierarchical Graph Network for Multi-hop Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 8823–8838.

Fionda, V.; and Pirrò, G. 2020. Learning Triple Embeddings from Knowledge Graphs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 3874–3881.

Garcia-Duran, A.; and Niepert, M. 2018. KBLRN: End-to-End Learning of Knowledge Base Representations with Latent, Relational, and Numerical Features. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, 372–381.

Guo, J.; and Kok, S. 2021. BiQUE: Biquaternionic Embeddings of Knowledge Graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8338–8351.

Guo, S.; Wang, Q.; Wang, L.; Wang, B.; and Guo, L. 2016. Jointly Embedding Knowledge Graphs and Logical Rules. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 192–202.

Han, X.; Cao, S.; Lv, X.; Lin, Y.; Liu, Z.; Sun, M.; and Li, J. 2018. OpenKE: An Open Toolkit for Knowledge Embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 139–144.

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 and Data Mining*, 1709–1719.

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*, 687–696.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2022. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2): 494–514.

Jiang, Y.; Wang, X.; Fan, H.; Liu, Q.; Du, B.; and Zhu, H. 2020. Modeling Relation Path for Knowledge Graph via Dynamic Projection. In *The 32nd International Conference on Software Engineering and Knowledge Engineering*, 65–70.

Kristiadi, A.; Khan, M. A.; Lukovnikov, D.; Lehmann, J.; and Fischer, A. 2019. Incorporating Literals into Knowledge Graph Embeddings. In *Proceedings of the 18th International Semantic Web Conference*, 347–363.

Kwak, J. H.; Lee, J.; Whang, J. J.; and Jo, S. 2022. Semantic Grasping via a Knowledge Graph of Robotic Manipulation: A Graph Representation Learning Approach. *IEEE Robotics and Automation Letters*, 7(4): 9397–9404.

Lao, N.; and Cohen, W. W. 2010. Relational Retrieval Using a Combination of Path-Constrained Random Walks. *Machine Learning*, 81: 53–67.

Li, G.; Sun, Z.; Qian, L.; Guo, Q.; and Hu, W. 2021. Rule-based Data Augmentation for Knowledge Graph Embedding. *AI Open*, 2: 186–196.

Lin, X. V.; Socher, R.; and Xiong, C. 2018. Multi-Hop Knowledge Graph Reasoning with Reward Shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3243–3253.

Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; and Liu, S. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 705–714.

Liu, H.; Wu, Y.; and Yang, Y. 2017. Analogical Inference for Multi-Relational Embeddings. In *Proceedings of the 34th International Conference on Machine Learning*, 2168–2178.

Meilicke, C.; Chekol, M. W.; Ruffinelli, D.; and Stuckenschmidt, H. 2019. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3137–3143.

Nayyeri, M.; Xu, C.; Alam, M. M.; Lehmann, J.; and Yazdi, H. S. 2021. LogicENN: A Neural Based Knowledge Graphs Embedding Model with Logical Rules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Niu, G.; Zhang, Y.; Li, B.; Cui, P.; Liu, S.; Li, J.; and Zhang, X. 2020. Rule-Guided Compositional Representation Learning on Knowledge Graphs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2950–2958.

Rosso, P.; Yang, D.; and Cudré-Mauroux, P. 2020. Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction. In *Proceedings of The Web Conference 2020*, 1885–1896.

Sadeghian, A.; Armandpour, M.; Ding, P.; and Wang, D. Z. 2018. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 15347–15357.

Sap, M.; Bras, R. L.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 3027–3035.

Sun, H.; Cohen, W.; and Salakhutdinov, R. 2022. ConditionalQA: A Complex Reading Comprehension Dataset with Conditional Answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 3627–3637.

Toutanova, K.; and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 57–66.

Voskarides, N.; Meij, E.; Reinanda, R.; Khaitan, A.; Osborne, M.; Stefanoni, G.; Kambadur, P.; and de Rijke, M. 2018. Weakly-Supervised Contextualization of Knowledge Graph Facts. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 765–774.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724–2743.

Wu, Y.; and Wang, Z. 2018. Knowledge Graph Embedding with Numeric Attributes of Entities. In *Proceedings of the 3rd Workshop on Representation Learning for NLP*, 132–136.

Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2316–2325.

Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; and Leskovec, J. 2021. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 535–546.

Zhang, H.; Liu, X.; Pan, H.; Song, Y.; and Leung, C. W.-K. 2020. ASER: A Large-scale Eventuality Knowledge Graph. In *Proceedings of The Web Conference 2020*, 201–211.

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019. Quaternion Knowledge Graph Embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2735–2745.