EmEL-V: $\mathcal{EL}^{++}$ Ontology Embeddings for Many-to-Many Relationships

Biswesh Mohapatra [a,*], Arundhati Bhattacharya [b], Sumit Bhatia [c], Raghava Mutharaju [b,**] and Srinivasaraghavan G [d]

[a] *INRIA, Paris, France*
E-mail: biswesh.mohapatra@inria.fr
[b] *CSE Department, IIIT-Delhi, New Delhi, India*
E-mails: arundhati18225@iiitd.ac.in, raghava.mutharaju@iiitd.ac.in
[c] *Media and Data Science Research Lab, Adobe Inc., Noida, India*
E-mail: sumit.bhatia@adobe.com
[d] *CSE Department, IIIT Bangalore, Karnataka, India*
E-mail: gsr@iiitb.ac.in

Abstract. Knowledge Graph (KG) embeddings provide a dense, low-dimensional representation of entities and relations in a Knowledge Graph and are used successfully for various applications such as reasoning, and missing link prediction, question answering and search. However, most of the existing KG embeddings only consider the network structure of the graph and ignore the semantics and the characteristics of the underlying ontology that provides crucial information about relationships between entities in the KG. Recent efforts in this direction involve learning embeddings for a description logic (logical underpinning for OWL 2 ontologies) named $\mathcal{EL}^{++}$. However, such methods consider all the relations defined in the ontology to be one-to-one which severely limits their performance and applications. We provide a simple and effective solution, named EmEL-V, to overcome this shortcoming that allows such methods to consider many-to-many relationships while learning embedding representations. Experiments conducted using three $\mathcal{EL}^{++}$ ontologies and one benchmark generated ontology on a reasoning task (class subsumption prediction) show substantial performance improvement over five baselines. Our proposed solution also paves the way for learning embedding representations for even more expressive description logics such as $\mathcal{SROIQ}$. The source code and the instructions to run it are available at https://github.com/kracr/el-embeddings. The ontologies used in the evaluation are available at https://doi.org/10.5281/zenodo.7023568.

Keywords: ontology reasoning, ontology embeddings, EL++ classification, Neuro-Symbolic AI

1. Introduction

Several advancements have been made over the past few decades in optimizing the description logic reasoners and this has lead to the development of very efficient reasoners such as Konclude [1], ELK [2], Pellet [3] and HermiT [4]. These classical reasoners are symbolic systems. They are interpretable and work very well on curated knowledge bases. But, this also makes them brittle and vulnerable to noisy data and minor changes in the logical encoding of the data. Another issue with the classical reasoners is that, although they work very well on small to medium size

[*] Work done as part of Master's thesis at IIIT Bangalore, Karnataka, India.
[**] Corresponding author. E-mail: raghava.mutharaju@iiitd.ac.in.

ontologies [5], they do not scale well to very large and expressive ontologies [6]. On the other hand, connectionist or subsymbolic systems are not only robust to noise and minor changes in the logical encoding structure [7], but can also scale very well. There has been growing interest in combining the techniques to get the best of both the worlds and build AI systems that are robust and interpretable [8].

Generating embeddings for Knowledge Graphs (KGs) [9] is an effort in this direction where the entities and relations of the KG are represented in the vector space to capture the structure of the graph [10]. These embeddings are used in various tasks such as link prediction [11], entity alignment [12], recommendation systems [13], zero-shot learning [14], fake news detection [15], ecotoxicological effect prediction [16] and question answering [17]. Several KG embedding techniques have been proposed [18–23]. However, most of these algorithms are for KGs that are composed of information in the form of RDF (Resource Description Framework) [24] triples consisting of subject (head), predicate (relation) and object (tail). They do not consider the OWL 2 (Web Ontology Language) [25] ontologies and the logical constructs that make up the ontologies. Using the KG embedding algorithms on OWL 2 ontologies to generate the embeddings gave poor results when used for reasoning tasks (discussed in Section 2 and later seen in Section 6). Considering the importance of OWL 2 ontologies in several domains [16, 26], a few attempts have been made to represent the ontology constructs in the vector space [7, 27–30]. Almost all of them work with a tractable fragment of OWL 2, namely, OWL 2 EL. In fact, they work on the description logic [31] $\mathcal{EL}^{++}$ or its subset, which is the formal underpinning for OWL 2 EL. Kulmanov et al., [27] proposed EL embeddings (we name this technique as EmEL) that incorporates the geometric structure of a subset of $\mathcal{EL}^{++}$ description logic ontologies into the embeddings. Mondal et al., [28] later added the role axioms (RBox) supported in $\mathcal{EL}^{++}$ into the embeddings through their proposed EmEL++ method. Ebrahimi et al., [7] have shown that a neural network is in fact capable of learning the structure, and not just the output, of a $\mathcal{EL}^{++}$ reasoning task without assistance. All these embedding based reasoning approaches (including our proposed approach) have the inherent advantage of scalability. After the models have been trained, they can scale to large and complex ontologies. The memory footprint during the reasoning task is also relatively small when compared with classical logic-based reasoners.

While EmEL and EmEL++ have provided a new technique to perform reasoning tasks on the ontologies, they have a fundamental issue that limits their performance on $\mathcal{EL}^{++}$ ontologies and restricts them from being used in more complex description logics such as $\mathcal{SROIQ}$, the logical underpinning for OWL 2 DL. We provide a simple and effective technique to capture the roles participating in many-to-many relations and not just one-to-one as is the case with the current state-of-the-art embedding techniques. This is significant because roles in ontologies connect a concept to multiple concepts. For example, the role R in following two axioms, $A_1 \sqsubseteq \exists R.B_1$, $A_2 \sqsubseteq \exists R.B_2$, participates in many-to-many relation since it connects A_1 to B_1 and A_2 to B_2. The problem of the embedding technique not being able to handle many-to-many relations is exacerbated in more expressive description logics such as $\mathcal{SROIQ}$ since they can potentially involve more logical constructs that support many-to-many relations (for example, cardinality constraints). In our proposed embedding technique, which is an extension of EmEL++, named EmEL-V, we fill this gap.

Our contributions in this work are as follows. (1) We provide a simple and effective method to incorporate roles participating in many-to-many relationships in translation-based embeddings. (2) We show the effectiveness of the method by modifying the ontological embedding EmEL [27]. (3) We create the embeddings of three publicly available ontologies and then compare their performance with existing embeddings. (4) We provide an ablation study on different types of relations on a large synthetic ontology created using OWL2Bench [32]. (5) We also provide a study on inferred axioms of ontologies for the class subsumption prediction task to understand the reasoning ability of our model.

2. Related Work

Various methods have been explored to create embeddings for knowledge graphs in the past. Node2Vec [33] used the concept of representing facts as triples of the form (h,r,t) which became a standard for various other models. Similarity based scoring functions were used in this work. In order to capture the underlying properties of the knowledge bases TransE [19] considered the relations in a KG as a translation operator over the entities and used distance based scoring functions. The success of a simple model like TransE prompted others to look into this direction. For example, TransH [18] further allowed the relations to be many-to-many and reflexive by modelling the relations as a hyperplane in the vector space. DistMult [20] on the other hand used matrix factorisation to relate the entities. PROCRUSTES [34] uses closed-form orthogonal procrustes analysis for KGEs while [35] builds linear model based on decomposition of tensor representation of knowledge graph triples. Above KG embeddings have performed well on knowledge graphs and are considered state-of-the-art methods. However, it has been noticed that these models do not achieve similar performance on ontologies. This has generated interest for newer approaches towards ontology embeddings. Existing works on ontology embedding such as Onto2vec [36] focuses on using word2vec as an underlying model. While the work focuses on encoding the entities and relations, it is unable to handle more complex inferred relations in an ontology. Hence, Ebrahimi et. al., [7] dive into neuro-symbolic solution by providing deep deductive reasoners for the $\mathcal{EL}^{++}$ description logic. Similarly, by taking an alternative approach, [29] has shown that random walk and word embedding-based ontology embeddings can achieve state-of-the-art performance.

Recently, it has been pointed out [37] that geometric models are a better way to learn embeddings for ontologies. Hence, EmEL [27] and EmEL++ [28] have used translation technique to create embeddings for ontologies which preserve their underlying structures and characteristics. In order to accomplish this, the models use geometric models to learn embeddings. The classes are considered to be n-balls in an n-dimensional space which are translated by the relation vectors to the n-ball of the corresponding class of the fact. This geometric structure provides a way to incorporate various structural properties of an ontology (for example, the subclass relation). The simplicity of the translation based models for KG embeddings [18–20] to measure the correctness of a fact as a distance between entities after being translated by the relation have made such models very popular. Similarly, the simplicity and effectiveness of EmEL [27] and EmEL++ [28] have made them stand out in ontology embeddings. However, like TransE, these models too restrict their triplets to a one-to-one mapping. Not only do these restrictions affect the performance of these models on $\mathcal{EL}^{++}$ ontologies but also restrict them from being used in more complex description logics such as $\mathcal{SROIQ}$. In this work, we introduce EmEL-V, an extension of EmEL++, to address this restriction.

3. Preliminaries

3.1. Semantics and Normal Forms

$\mathcal{EL}^{++}$ is a lightweight and tractable description logic that provides the formal underpinning for the OWL 2 EL profile [25]. The signature Σ for the $\mathcal{EL}^{++}$ description logic is defined as $\Sigma = \langle N_C, N_R, N_I \rangle$, where N_C, N_R, and N_I are countably infinite, mutually disjoint sets of concept names (including $\top$), role names, and individual names respectively. An $\mathcal{EL}^{++}$ knowledge base consists of a finite set of axioms of the form $C \sqsubseteq C$, $R \sqsubseteq R$ and $R \circ \ldots \circ R \sqsubseteq R$, where C and R are defined by the following grammar.

$$\begin{array}{lll} R & ::= & N_R \\ C & ::= & N_C \quad \mid \quad C \sqcap C \quad \mid \quad \exists R.C \end{array}$$

The semantics of $\mathcal{EL}^{++}$ are defined by the interpretations $I = (\Delta^I, \cdot^I)$ that map the sets N_C, N_R, and N_I to elements, sets and relations in the domain of interpretation Δ^I. For an interpretation I, the function $\cdot^I$ is shown in Table 1 [38].

All the axioms in an $\mathcal{EL}^{++}$ ontology can be converted into fixed forms using the normalization procedure [39]. The class subsumption axioms, also called as the general concept inclusion axioms can be reduced to one of the following four forms. Here C, D and $E \in N_C$.

$$C \sqsubseteq D, \quad C \sqcap D \sqsubseteq E, \quad C \sqsubseteq \exists R.D, \quad \exists R.C \sqsubseteq D$$

All the role inclusion axioms can be reduced to one of the following two forms. Here R, R_1, R_2 and $S \in N_R$.

$$R \sqsubseteq S, \quad R_1 \circ R_2 \sqsubseteq S$$

Table 1
$\mathcal{EL}^{++}$ semantics, where the symbol $\circ$ is used for the standard binary composition.

Description	**Expression**	**Semantics**
Individual	a	$a \in \Delta^{\mathcal{I}}$
Top	$\top$	$\Delta^{\mathcal{I}}$
Bottom	$\bot$	$\emptyset$
Concept	C	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Existential Restriction	$\exists R.C$	$\left\{ a \mid \text{there is } b \in \Delta^{\mathcal{I}} \text{ such that } (a, b) \in R^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}} \right\}$
Concept Subsumption	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Role Subsumption	$R \subseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Role Chain	$R_1 \circ \ldots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \ldots R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$

Further, the class and role assertion axioms in the ABox can be converted into TBox axioms as follows.

$$C(a) \longrightarrow \{a\} \sqsubseteq C, \quad R(a,b) \longrightarrow \{a\} \sqsubseteq \exists R.\{b\}$$

Thus, with the above transformations, all the axioms in an $\mathcal{EL}^{++}$ ontology can be reduced to one of the normal forms and the task of embedding ontologies in the vector space requires us to learn mapping functions for classes and relations that are part of the normal forms.

3.2. Types of Relationships

A role $R \in N_R$ can participate in any of the following types of relationships. Please note that these relationship types may not be a standard nomenclature in description logics, but these categories are helpful to discuss our contributions.

1. One-to-One. If R participates in a relationship involving exactly two classes, it is referred to as a one-to-one relation. For example, in the axiom, $A \sqsubseteq \exists R.B$, R connects the concepts A and B. Assuming that R is not connected to any other entities in the knowledge base, this will be a one-to-one relation.
2. One-to-Many. If R connects one class to multiple different classes, this type of relationship is referred to as one-to-many. For example, R is participating in a one-to-many relation with respect to the following axioms, $A \sqsubseteq \exists R.B$ and $A \sqsubseteq \exists R.C$.
3. Many-to-One. If R connects multiple classes to a single class, it is referred to as a many-to-one relation. For example, the role R participates in a many-to-one relation with respect to the axioms, $A_1 \sqsubseteq \exists R.B$ and $A_2 \sqsubseteq \exists R.B$.
4. Many-to-Many. If R participates in multiple classes on either side, it is referred to as a many-to-many relation. For example, R is participating in a many-to-many relation with respect to the axioms, $A_1 \sqsubseteq \exists R.B_1$ and $A_2 \sqsubseteq \exists R.B_2$.

4. Ontology Embeddings

Kulmanov et al. [27] introduced the concept of incorporating geometric structure of ontologies into the embeddings. They proposed embeddings for the $\mathcal{EL}^{++}$ description logic (EmEL) that captures the underlying structures and characteristics of the ontology by treating ontology classes as n-balls in n-dimensional space. These n-balls are represented by a center which is an n-dimensional vector and a radius which is a scalar. The relations in the ontology are considered as n-dimensional vectors which are used to translate the class(represented by n-balls) from one location in the space to another. The center and the radius of each class (n-ball), along with the relations can be learnt over multiple iterations. They make up the embeddings for the ontology. Figure 1(a) and Figure 1(b) show the geometric representation of classes and relations in 2-dimensional space.

Hence, they define a geometric ontology embedding η as a pair (f_η, r_η) of functions that map classes and relations in ontology O into $\mathbb{R}^n$. Thus f_η : C ∪ R $\mapsto \mathbb{R}^n$ and r_η : C $\mapsto \mathbb{R}$. Here C is a set of class and R is a set of relation and O is defined as $(C, R, \mathbb{I}$; ax) where $\mathbb{I}$ are individual symbols, C is set of class symbols, R is set of relation symbols and ax are axioms (facts). Basically, f_η(c) represents center of class c, r_η(c) represents radius of class c and f_η(r) represents vector of relation r.

Each axiom, ax, is transformed into its equivalent normal form using a set of conversion rules from [39]. These rules help transform the set of axioms in the ontology into one of four forms without any loss of information. These are (1) Subclass axiom: $C \sqsubseteq D$ (2) Intersection axiom: $C \sqcap D \sqsubseteq E$ (3) Existential restriction (right-hand side): $C \sqsubseteq \exists R.D$ and (4) Existential restriction (left-hand side): $\exists R.C \sqsubseteq D$ where C, D, E $\in \mathbb{C}$ and R $\in \mathbb{R}$.

EmEL formulates a loss function for each of the four normal forms in order to preserve the semantics of $\mathcal{EL}^{++}$ in the embeddings. The loss functions are as follows.

$$\begin{aligned}\text{loss}_{C\sqsubseteq D}(c,d) = \max(0, \parallel f_\eta(c) - f_\eta(d) \parallel + r_\eta(c) - r_\eta(d) - \gamma)\\ + \mid \parallel f_\eta(c) \parallel -1 \mid + \mid \parallel f_\eta(d) \parallel -1 \mid\end{aligned} \tag{1}$$

In Eqn 1, we try to preserve the subclass property of the entities. Here the euclidean distance between the centers of C and D should be less than the difference between the radius of D and C. Once this is achieved, we ensure that the n-ball representing D is bigger than that of C and that n-ball of C lies completely inside D. Here γ is a hyperparameter called margin. $\mid \parallel f_\eta(c) \parallel -1 \mid + \mid \parallel f_\eta(d) \parallel -1 \mid$ ensures that the n-balls lie in the unity sphere.

$$\begin{aligned}\text{loss}_{C\sqcap D\sqsubseteq E}(c,d,e) = \max(0, \parallel f_\eta(c) - f_\eta(d) \parallel - r_\eta(c) - r_\eta(d) - \gamma)\\ + \max(0, \parallel f_\eta(c) - f_\eta(e) \parallel - r_\eta(c) - \gamma)\\ + \max(0, \parallel f_\eta(d) - f_\eta(e) \parallel - r_\eta(d) - \gamma)\\ + \mid \parallel f_\eta(c) \parallel -1 \mid + \mid \parallel f_\eta(d) \parallel -1 \mid \parallel f_\eta(e) \parallel -1 \mid\end{aligned} \tag{2}$$

In Eqn 2, we incorporate the intersection property. The first term ensures that C and D are not disjoint sets. While second and third terms force the center of E to lie in the intersection of D.

$$\begin{aligned}\text{loss}_{C\sqsubseteq \exists R.D}(c,d,r) = \max(0, \parallel f_\eta(c) + f_\eta(r) - f_\eta(d) \parallel + r_\eta(c) - r_\eta(d) - \gamma)\\ + \mid \parallel f_\eta(c) \parallel -1 \mid + \mid \parallel f_\eta(d) \parallel -1 \mid\end{aligned} \tag{3}$$

$$\begin{aligned}\text{loss}_{\exists R.C\sqsubseteq D}(c,d,r) = \max(0, \parallel f_\eta(c) - f_\eta(r) - f_\eta(d) \parallel - r_\eta(c) - r_\eta(d) - \gamma)\\ + \mid \parallel f_\eta(c) \parallel -1 \mid + \mid \parallel f_\eta(d) \parallel -1 \mid\end{aligned} \tag{4}$$

Eqn 3 and Eqn 4 describe the loss function for the third and fourth normal forms respectively. Every point that lies within an n-ball representing a class is a potential instance of that class. The loss functions capture this by applying relations as translations on these points (following the TransE [19] relation model). The relation vector $f_\eta(r)$ when added to the center of class C should be at a maximum distance of the sum of radii of C and D from the center of D. Eqn 4 reverses the direction of translation from Eqn 3.

$$\begin{aligned}\text{loss}_{C\sqcap D\sqsubseteq \bot}(c,d) = \max(0, r_\eta(c) + r_\eta(d) - \parallel f_\eta(c) - f_\eta(d) \parallel + \gamma)\\ + \mid \parallel f_\eta(c) \parallel -1 \mid + \mid \parallel f_\eta(d) \parallel -1 \mid\end{aligned} \tag{5}$$

Eqn 5 describes the loss function for disjoint classes C and D while Eqn 6 refers to the specific loss function for bottom class whose radius must be equal to zero.

$$\text{loss}_{C\sqsubseteq\bot}(c) = r_\eta(c) \tag{6}$$

EmEL++ [28] added role constructors to EmEL. Eqn 7 and Eqn 8 provide loss functions added in EmEL++. For role inclusion property, in Eqn 7 relation R is a sub-role of relation S. Eqn 7 makes sure that the length of both the relations R and S remain the same while keeping the angle between them zero. It ensures the role inclusion property by assuming that the vectors should be of the same length and pointing towards same direction if they connect similar classes together.

$$\begin{aligned}\text{loss}_{R\sqsubseteq S}(r,s) = {} & \max(0, \| e_v(s) - e_v(r) \| - \gamma) \\ & + \left| 1 - \frac{e_v(s).e_v(r)}{\| e_v(s) \| \| e_v(r) \|} \right| \\ & + | \| e_v(s) \| - 1| + | \| e_v(r) \| - 1| \end{aligned} \tag{7}$$

Eqn 8 looks at the role composition property. Here combination of R_1 and R_2 should be equivalent to relation S. Hence, the sum of vectors R_1 and R_2 should equal the vector corresponding to the relation S. Similarly the direction made by adding R_1 and R_2 should be the same as the direction of S.

$$\begin{aligned}\text{loss}_{R_1\circ R_2\sqsubseteq S}(r_1,r_2,s) = {} & \max(0, \| e_v(s) - e_v(r_1) - e_v(r_2) \| - \gamma) \\ & + \left| 1 - \frac{e_v(s).(e_v(r_1) + e_v(r_2))}{\| e_v(s) \| \| e_v(r_1) + e_v(r_2) \|} \right| \\ & + | \| e_v(s) \| - 1| + | \| e_v(r_1) \| - 1| + | \| e_v(r_2) \| - 1| \end{aligned} \tag{8}$$

The translation approach of EmEL and EmEL++ is not suitable for more expressive description logics such as $\mathcal{SROIQ}$ because the translation operator on relations makes them one-to-one. This limits the capabilities of the model as most of the relations are many-to-many. We propose a modification to their approach, named EmEL-V, to overcome the issue.

5. Approach

To address the one-to-one relation restriction, we used a simple yet powerful technique that provides a foundation for further work in embeddings based description logic reasoning. We consider the relations to have a variance (uncertainty) leading to the translation having various possible regions in the vector space. This lets us model one-to-many and many-to-many relations in the ontology. As a result, we can model complex properties such as cardinality.

We consider the variance to be a hard bound. The translation of n-ball C on relation vector R could now be within σ distance of n-ball D in a tuple (C, R, D) where C, D $\in \mathbb{C}$ and R $\in \mathbb{R}'$. Hence all the points within σ distance from the translated space are related to C through R. This removes the one-to-one limitation of previous methods. We call this model EmEL-V.

Every relation has its own σ which is learnt during training. In order to avoid σ from becoming infinite, we keep the absolute value of σ as a loss component for regularisation. Figure 2 shows a visual representation of EmEL-V. The definition of the geometric ontology embedding η now becomes a tuple $(f_\eta, r_\eta, \sigma_\eta)$ of functions that map classes and relations in ontology O into R^n, where f_η : C $\cup$ R $\mapsto \mathbb{R}^n$, r_η : C $\mapsto \mathbb{R}$ and σ_η : R $\mapsto \mathbb{R}$. The modified loss

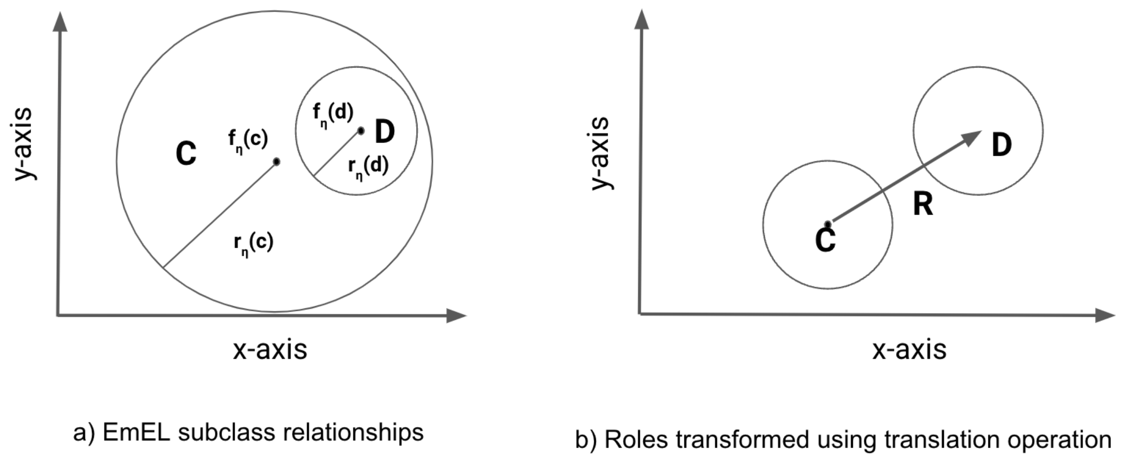

Fig. 1. Pictorial representation of the classes and relations in their geometric space. (a) shows the representation of subclass relation where D $\sqsubseteq$ C and thus the n-ball of D lies inside n-ball of C. (b) shows class C getting translated to D using relation R for a tuple (C,R,D) in the ontology.

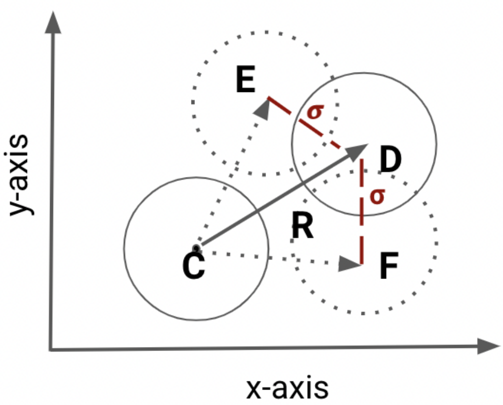

Fig. 2. Visualisation of EmEL-V. The variance σ lets the the entity C relate to multiple entities with the relation R. Any entity which falls within σ distance of C+R are also related to C through R. E and F are the boundary entities for C and R.

functions are provided in Eqn 9 and Eqn 10. Note that the loss function for other normal forms remain the same as in EmEL++.

$$\begin{aligned} \text{loss}_{C\sqsubseteq\exists R.D}(c,d,r) = \max(0, \| f_\eta(c) + f_\eta(r) - f_\eta(d) \| + r_\eta(c) - r_\eta(d) - \sigma_\eta(r) - \gamma) \\ + | \| f_\eta(c) \| - 1| + | \| f_\eta(d) \| - 1| + \sigma_\eta(r) \end{aligned} \tag{9}$$

$$\begin{aligned} \text{loss}_{\exists R.C\sqsubseteq D}(c,d,r) = \max(0, \| f_\eta(c) - f_\eta(r) - f_\eta(d) \| - r_\eta(c) - r_\eta(d) - \sigma_\eta(r) - \gamma) \\ + | \| f_\eta(c) \| - 1| + | \| f_\eta(d) \| - 1| + \sigma_\eta(r) \end{aligned} \tag{10}$$

6. Evaluation

We compare the proposed EmEL-V model with eight different models for learning knowledge graph and ontology embeddings. We use synthetic as well as real world datasets for our evaluation. We also study how the different methods for learning ontology embeddings perform on the three types of relations (one-to-one, one-to-many, and many-to-many).

6.1. Experiment Setup

6.1.1. Datasets

We use the following publicly available ontologies of varying sizes for our experiments. All the ontologies used in our evaluation are available at https://doi.org/10.5281/zenodo.7023568.

1. **GALEN** [40] is an ontology of medical terms aiming to capture the medical intuition of 'disease' or 'disorder'. It is based on a modelling theory embodied in the GALEN Representation And Integration Language (GRAIL) Kernel. It consists of 84,537 axioms with 1,010 relations and 24,353 classes.
2. **Gene Ontology (GO)** [41] captures information about genes, their molecular functions and biological processes. It is the world's largest source of information on the functions of genes. It is an initiative to unify the representation of gene and gene product attributes across all species. This knowledge is both human-readable and machine-readable. It consists of 130,094 axioms with 45,907 classes and 16 relations.
3. **SNOMED CT** [42] is a comprehensive ontology of clinical terms. The primary purpose of SNOMED CT is to encode the meanings that are used in health information and to support the effective clinical recording of data with the aim of improving patient care. It has 989,186 axioms, 307,712 classes and 60 relations. It is maintained and distributed by SNOMED International, an international non-profit standards development organization.

Further, in order to study in detail how the proposed EmEL-V performs in comparison to the EmEL and EmEL++ models on the three kinds of relations, we extract three variants from GALEN, GO, and SNOMED CT ontologies as follows.

1. One-to-One (1:1): a class is associated with exactly one other class through a relation.
2. One-to-Many (1:N): a class is associated with more than one class through a relation.
3. Many-to-Many (N:M): multiple classes are associated with a single class (and vice-versa) through a relation

Table 6.1.1 provides information on the total number of relations and axioms in the obtained datasets. All these ontologies are first normalized by converting the axioms into one of the six normal forms discussed in Section 3. We then identify unique relations from the *existential restriction* axioms and categorize each relation into one-to-one, one-to-many, and many-to-many. For each ontology and each relation-type, we then create a dataset containing all of the subclass and intersection axioms, and the existential axioms corresponding to that category of relations.

Additionally, we also create a large synthetic ontology using Owl2Bench [32] generator in order to study the performance of our model on ontologies containing only one-to-one relations, only one-to-many relations and only many-to-many relations. Since, the three ontologies used contain uneven distribution of these relations, we decided to create a large synthetic ontology to get a better understanding of the effect of our model on different relation types. This study further helps us understand how our method helps provide the flexibility to the relations which were previously restricted. Table 2 has information on the synthetic ontology.

6.1.2. Baselines

We compare the performance of the proposed EmEL-V model with the following different baselines.

1. **TransE** [19]. This model introduced the notion of *translation* in knowledge graph embeddings where relations help *translate* the source entity vector to the target entity vector in the vector space.
2. **TransH** [18]. The translation operator as introduced in TransE is limited to one-to-one relationships. To overcome this limitation, TransH it considers relations as hyperplanes and uses projection of entities on those hyperplanes to relate to other entities.

Table 2
Owl2Bench dataset size

Relation Type	Classes	Relations	Axioms
One-to-one	19488	26	23582
One-to-many	17797	4	23751
many-to-many	23341	30	57747

Table 3
Characteristics of the three variants of GALEn, GO, and SNOMED ontologies. 1:1 indicates the variant with one-to-one relationships, 1:N indicates the variant with one-to-many relationships, and N:M indicates the variant with many-to-many relationships.

Dataset	Relation type	Relations	Axioms
GALEN(1:1)	One-to-One	180	41896
GALEN(1:N)	One-to-Many	17	41028
GALEN(N:M)	Many-to-Many	199	77882
GO(1:1)	One-to-One	1	89095
GO(1:N)	One-to-Many	0	89094
GO(N:M)	Many-to-Many	7	121546
SNOMED(1:1)	One-to-One	0	474407
SNOMED(1:N)	One-to-Many	1	675048
SNOMED(N:M)	Many-to-Many	56	788545

3. **DistMult** [20] is a matrix factorization based model that has been found to perform well empirically on compositional reasoning tasks.
4. **ProcrustES** [34] uses closed-form Orthogonal Procrustes Analysis, relational matrices and non-negative-sampling training for learning knowledge graph embeddings.
5. **TuckER** [35] trains a linear model based on Tucker decomposition of binary tensor representation of knowledge graph triples.
6. **EmEL** [27] is amongst the first method for learning embeddings of ontologies.
7. **EmEL++** [28] is an extension EmEL embeddings with additional capabilities to model properties like role inclusion and role chain.
8. **OWL2Vec*** [29] is the current state-of-the-art which uses random walk on the graph and word embeddings for learning embedding representation of ontologies. Although we include a comparison with OWL2Vec* in our evaluation, it must be noted that the approach used by OWL2Vec* can be complementary to our approach by combining the word embedding with our proposed embedding technique.

6.1.3. *Evaluation Metrics*

Typically, link prediction has been the downstream task of choice for evalutaing the quality of knowledge graph embeddings learnt by different models. While link prediction is suitable for evaluating methods [18–20, 43] that rely only on the network structure of the graph, it is not a suitable task for testing the capability of the model in using ontological information. In order to study the ability of the proposed EmEL-V model to capture, and use, the ontological information, we chose the class subsumption prediction as our downstream task. Subsumption involves checking whether the subclass relation exists between two classes and requires making explicit use of the normalized axioms to infer the subclass relation. Note that the task of subsumption can be reduced to a distance-based operation in the embedding vector space. Given a test instance of the form $C \sqsubseteq D$, we use D as source class and rank all other classes in the given ontology in an increasing order of their distance from D in the vector space. Based on the rank at which C is present in the ranked list, we can evaluate different models for learning ontology embeddings. We posit that an embedding model that successfully captures the ontological information should be able to assign very close vector representations to the two classes in a subclass relation, hence, producing a lower rank for C.

In order to compare the performance of different models, we use *Hits* at ranks 1, 10 and 100 as our evaluation metrics that report the fraction of the test cases where the given class C falls under Hits@ 1, 10 and 100 in rank list respectively. We also consider *median rank* and *90^{th} percentile rank* to compare the overall performance of the different models. A median rank of *m* indicates that for 50% of the test cases, the correct answer was found below rank *m*. Similarly 90^{th} percentile rank indicates the rank below which the correct class was found for 90% of the test cases.

Table 4
Performance of Different Methods for Galen

Model	Hits@1	Hits@10	Hits@100	Median	90th% Rank
TransE	0.00	0.00	0.00	10748	21308
TransH	0.00	0.00	0.00	11721	21825
DistMult	0.00	0.00	0.00	12600	21823
ProcrustES	0.00	0.00	0.00	9641	96824
TuckER	0.00	0.00	0.01	12229	88293
EmEL	0.01	0.11	0.19	6039	21221
EmEL++	0.02	0.11	0.16	6623	20635
OWL2Vec*	0.01	**0.14**	**0.33**	**946**	25927
EmEL-V	**0.10**	0.12	0.26	3540	**19669**

6.1.4. Implementation Notes

In order to implement the proposed EmEL-V method for learning the embedding representations of ontologies, we use Pytorch [44] and its embedding layers. For the three knowledge base embedding models (TransE, TransH, and DistMult), we use the models as implemented in the Pykeen framework [45]. For EmEL and EmEl++ embeddings, source code provided by the authors was used. Similarly, we used the source code with default parameters for ProcrustES, TuckER and OWL2Vec* provided by their respective authors. In order to learn the embeddings for different models, we first normalize the ontologies, i.e., convert the axioms into one of the four normal forms discussed in Section 4. All individuals in the ontology are considered as nominal classes (containing one instance) and the embeddings learn to make their radii zero resulting in a point in the vector space. Next, we remove some of the subclass relation pairs for validation (20%) and testing (10%). Remaining 70% sub-class relation pairs are used for training the embedding functions. The source code for the experiments is available at https://github.com/kracr/el-embeddings.

6.2. Empirical Results

6.2.1. Results on GO, GALEN and SNOMED Ontologies

The evaluation results of the model on the subsumption task for the GO, GALEN, and SNOMED ontologies are presented in Tables 4, 5 and 6 respectively. For all the three cases, we note that EmEL-V outperforms all the traditional KGE models across all the five metrics. We also note that the two $\mathcal{EL}$ specific methods, i.e., EmEL and EmEL++, perform better than other traditional KG Embeddings i.e. TransE, TransH and DistMult indicating the importance of capturing underlying structures of the ontologies. EmEL++, in general, tends to perform slightly better than EmEL on various metrics as well. EmEL-V surpasses the performance of EmEL and EmEL++ on almost every metric. We do note that OWL2Vec* is a very strong baseline for all the datasets and there is no clear winner among OWL2Vec* and EmEL-V. It is interesting to note that EmEL-V outperforms OWL2Vec* on metrics like Hits@1 and 90*th* percentile rank while lagging behind in metrics such as Hits@10, Hits@100 and median-rank. Detailed analysis on each dataset is done below.

EmEL-V improves the performance on the Galen ontology compared to EmEL and EmEL++. A notable improvement is seen in the Hits@1 and Hits@100 values i.e. Hits@1 score increases from 0.02 in EmEL++ to 0.10 in EmEL-V which is quite significant. Similarly, we see an increase of Hits@100 score from 0.19 in EmEL to

0.26 in EmEL-V. The median rank becomes almost half of the previous best model and a significant improvement is also seen in 90th percentile rank. Although OWL2Vec* performs significantly better in metrics like Hits@10, Hits@100 and median rank. While, EmEL-V is significantly better at Hits@1 and 90th percentile rank indicating that OWL2Bench is either performing extremely well for some classes or its extremely bad for other classes. EmEL-V on the other hand is more balanced. Similarly, we observe from Table 5 that EmEL-V significantly outperforms all the KG embeddings, EmEL and EmEL++. OWL2Vec* on the other hand outperforms EmEL-V except for Hits@1 and 90th percentile rank. Table 6 presents the results for SNOMED CT, the largest of the three ontologies considered in this work. The Hits@1, Hits@10 and Hits@100 scores for EmEL-V are higher than any of the other models and large improvements observed especially in the Median and 90th percentile rank metrics. This can be attributed to the fact that SNOMED CT has a large number of relations that have many-to-many properties. EmEL-V provides the model the freedom to incorporate those properties resulting in a substantially improved embedding. It is interesting to notice that EmEL-V performs better than OWL2Vec* on every metric other than median rank.

Table 5
Performance of Different Methods for GO

Model	Hits@1	Hits@10	Hits@100	Median	90th% Rank
TransE	0.00	0.00	0.00	20079	40177
TransH	0.00	0.00	0.00	26280	41996
DistMult	0.00	0.00	0.00	22493	40425
ProcrustES	0.00	0.00	0.02	32490	98037
TuckER	0.00	0.00	0.01	22114	227121
EmEL	**0.01**	0.08	0.15	9504	36447
EmEL++	**0.01**	0.09	0.15	7232	34148
OWL2Vec*	**0.01**	**0.15**	**0.44**	**498**	414905
EmEL-V	**0.01**	0.09	0.17	7542	**33892**

Table 6
Performance of Different Methods for SNOMED CT

Model	Hits@1	Hits@10	Hits@100	Median	90th% Rank
TransE	0.00	0.00	0.00	150876	274465
TransH	0.00	0.00	0.00	157186	278455
DistMult	0.00	0.00	0.00	151624	275982
ProcrustES	0.00	0.00	0.01	186604	825073
TuckER	0.00	0.00	0.00	212708	878045
EmEL	0.00	0.03	0.08	80289	277874
EmEL++	0.00	0.03	0.06	87413	261359
OWL2Vec*	0.02	0.14	**0.36**	**498**	167171
EmEL-V	**0.08**	**0.18**	**0.36**	42759	**134829**

6.2.2. Results on Synthetic Dataset Generated Using OWL2Bench

Encouraged by the performance of our model on a larger ontology like SNOMED CT, we decided to create a large synthetic ontology using [32]. Creating an synthetic ontology helps maintain a good ratio of different relation types compared to those found in ontologies like Galen, Go and SNOMED CT. Hence our OWL2Bench dataset has a good ratio of relations that are one-to-one, one-to-many and many-to-many. We now separate the ontology and train our models such that we use only one-to-one relation, only one-to-many relations or only many-to-many relations. This study helps us understand the strength and weakness of our proposed model and also helps us understand the real reason behind the better performance of our model compared to EmEL and EmEL++ which were restricted to

Table 7
Performance on OWL2Bench dataset containing only one-one relations

Model	Hits@1	Hits@10	Hits@100	Median	90th % Rank
EmEL	0.00029	0.00982	0.01488	11860	21120
EmEL++	0.00000	0.00833	0.01160	**11369**	20874
OWL2Vec*	0.00029	**0.01160**	**0.0205**	16324	158745
EmEL-V	**0.00059**	0.01071	0.01458	11447	**20831**

one-to-one relations. We only compare with other ontology embedding models as we have seen that KG embeddings perform poorly when trained on ontologies.

As seen in Table 7, when trained with ontology containing only one-to-one relations, while EmEL-V performs significantly better than other models in Hits@1, it doesn't do that well on other metrics. This is understandable, since EmEL-V doesn't provide any additional advantage to EmEL and EmEL++ when ontology only contains one-to-one relations.

Table 8
Performance on OWL2Bench dataset containing only one-many relations

Model	Hits@1	Hits@10	Hits@100	Median	90th Percentile
EmEL	0.00029	0.01447	0.01891	11733	21094
EmEL++	0.00118	0.01743	**0.03309**	11843	20985
OWL2Vec*	0.00118	0.01791	0.02364	10692	170777
EmEL-V	**0.00177**	**0.01832**	0.02216	**9994**	**19713**

On the contrary, we see EmEL-V performing better than all ontology embeddings in Table 8. This is the result when the ontology contains only one-to-many relations. Hence, it is evident from the result that adding variance to the relations in a translation based ontology helps provide the necessary freedom for modelling one-to-many relations.

Table 9
Performance on OWL2Bench dataset containing only many-many relations

Model	Hits@1	Hits@10	Hits@100	Median	90th Percentile
EmEL	0.00011	0.00584	0.00966	12009	21047
EmEL++	0.00011	**0.00596**	0.01020	11350	21054
OWL2Vec*	0.00011	0.00584	**0.01228**	144120	171420
EmEL-V	**0.00023**	0.00584	0.00811	**11330**	**20951**

Similarly, in Table 9 we see EmEL-V scoring highest in Hits@1, Median Rank and 90th Percentile Rank while performing not being far behind in the other two metrics. This again shows the effectiveness of our model when compared to other ontology embeddings. Interestingly, we also notice that OWL2Vec* doesn't perform as well when the ontology is big i.e., it did not achieve better performance on datasets like SNOMED CT and OWL2Bench.

6.2.3. How the models fare on different relation types?

To do further ablation studies on the impact of EmEL-V, we also look at the one-one, one-many and many-many relations of the three ontologies, i.e., GO, GALEN and SNOMED. While these datasets are not as well balanced as the synthetic ontology generated using OWL2Bench, we wanted to study the performance of various models for different relation types. The results reported in Tables [10, 11, 12] reveal that while *EmEL-V* does relatively better than *EmEL* and *EmEL++*, it lacks behind *OWL2Vec**. This is inline with our previous findings where the *OWL2Vec** performs better than *EmEL-V* when the ontology is small in size while *EmEL-V* performs better than *OWL2Vec** when the ontology is bigger as in the case of OWL2Bench and the full SNOMED ontology.

Table 10
Performance on GALEN dataset with different kinds of relations

Type of Relation	Model	Hits@1	Hits@10	Hits@100	Median Rank	90th % Rank
One-to-One	**EmEL++**	**0.03174**	0.13795	0.17757	9758	21526
	EmEL	0.00048	0.00835	0.04057	8905	**21082**
	OWL2Vec*	0.01758	0.10115	0.23888	**3194**	29838
	EmEL-V	0.02625	**0.14487**	**0.25585**	5146	21379
One-to-Many	**EmEL++**	**0.03704**	0.14742	0.19688	8879	21381
	EmEL	0.00219	0.00999	0.04264	8223	**20779**
	OWL2Vec*	0.02044	0.09110	0.24641	**2736**	29650
	EmEL-V	0.03363	**0.15936**	**0.27583**	5868	21639
Many-to-Many	**EmEL++**	0.02260	0.11824	0.16189	8067	21265
	EmEL	0.00077	0.00591	0.03107	8652	20637
	OWL2Vec*	**0.03802**	**0.22453**	**0.50143**	**99**	20923
	EmEL-V	0.01926	0.14238	0.26781	3983	**20468**

Table 11
Performance on GO dataset with different kinds of relations

Type of Relations	Model	Hits@1	Hits@10	Hits@100	Median Rank	90th % Rank
One-to-One	**EmEL++**	0.00864	0.03019	0.05297	12091	34522
	EmEL	0.00011	0.00022	0.00157	22329	40206
	OWL2Vec*	0.00842	**0.07643**	**0.24568**	**1918**	40882
	EmEL-V	**0.00898**	0.03221	0.05432	10225	**34239**
One-to-Many	**EmEL++**	0.01100	0.03030	0.04837	10621	34909
	EmEL	0.01010	0.03064	0.05084	9253	34713
	OWL2Vec*	**0.07811**	**0.35293**	**0.57648**	**37**	42730
	EmEL-V	0.00909	0.03243	0.05544	10394	**33694**
Many-to-Many	**EmEL++**	0.00765	0.03826	0.06738	8346	34337
	EmEL	0.00362	0.01678	0.03390	13930	37236
	OWL2Vec*	**0.08893**	**0.39515**	**0.62476**	**22**	**32766**
	EmEL-V	0.00765	0.08581	0.12077	9802	34700

6.2.4. Performance of Different Models on Inferred data

In order to test the model on a real world scenario, we follow Mondal et. al., [30] to create new relations inferred from the existing ontologies. These inferred relations are results of complex reasoning on the existing ontologies i.e. Galen, Go and Snomed. We compare the performance of our model with other ontology specific models like in previous ablation study to understand the real ability of the model to reason over the ontologies. The models used for testing the inferred data are the same as the ones used in the ablation study.

The results are provided in Tables[13, 14, 15]. As seen, in the previous experiments, *EmEL-V* does better than *EmEL* and *EmEL++* in all scenarios and is most of the time performing better than *OWL2Vec** as well. While Table 13 and Table 15 show that *EmEL-V* performs exceedingly well compared to *OWL2Vec** for Galen and SNOMED CT datasets. Table 14 shows that *OWL2Vec** performs better than *EmEL-V* on GO inferred data. However, our model is almost comparable in all the metrics with *OWL2Vec** while outperforming *EmEL* and *EmEL++* by huge margins. As a result, we are able to see the ability of the model to generalise and reason over the ontologies even if the new inferred relations have been realised through complex reasoning over existing relations.

Table 12
Performance on SNOMED dataset with different kinds of relations

Type of Relations	Model	Hits@1	Hits@10	Hits@100	Median Rank	90th % Rank
One-to-One	**EmEL++**	0.01020	0.06138	0.08724	94814	253990
	EmEL	0.00000	0.00000	0.00030	153352	276774
	OWL2Vec*	**0.01103**	**0.09074**	**0.11930**	**8023**	**218294**
	EmEL-V	0.01001	0.06593	0.10503	109906	264535
One-to-Many	**EmEL++**	0.00000	0.00003	0.00028	154357	276641
	EmEL	0.00000	0.00001	0.00034	154384	277002
	OWL2Vec*	0.00802	**0.06349**	**0.10021**	**1904**	299029
	EmEL-V	**0.00991**	0.04431	0.05391	139990	**273503**
Many-to-Many	**EmEL++**	0.00751	0.03032	0.03891	125418	269872
	EmEL	0.00000	0.00005	0.00033	153489	276837
	OWL2Vec*	**0.01920**	**0.12930**	**0.32019**	**502**	**167171**
	EmEL-V	0.01116	0.04719	0.05937	130670	270890

Table 13
Performance on GALEN Inferred test data

Type of Relation	Model	Hits@1	Hits@10	Hits@100	Median Rank	90% Rank
one-one	**EmEL**	0.0802	0.2587	0.3300	6567.3235	19211.8
	EmEL++	0.0050	0.0240	0.0745	7625.7630	19071
	Owl2vec*	0.0156	0.0895	0.2340	11063.7174	30333.8
	EmEL-V	**0.0822**	**0.3341**	**0.5259**	**4226.0897**	**17169.6**
one-many	**EmEL**	0.0865	0.2867	0.3774	1019	19396
	EmEL++	0.0068	0.0275	0.0782	4947	18747.8
	Owl2vec*	0.0161	0.0885	0.2370	3725	30101.4
	EmEL-V	**0.0973**	**0.3488**	**0.5548**	**48**	**17402.6**
many-many	**EmEL**	0.0375	0.1464	0.1952	7020	21120.6
	EmEL++	0.0248	0.0551	0.0937	7733	20391.8
	OWL2Vec*	0.0371	**0.2343**	**0.5006**	**100**	20957
	EmEL-V	**0.0429**	0.1929	0.3239	3646	**20305**

6.2.5. *Summary of Experiments and Findings*

We tested our model on three different types of existing ontologies which have varying properties. The results on all the three datasets, show that EmEL-V performs better than all the traditional knowledge graph embedding based models and also far exceeds in performance compared to other EmEL based models. EmEL-V performs comparably with OWL2Vec* on the different metrics. The general trend when comparing EmEL-V with OWL2Vec* is that EmEL-V performs better in Hits@1 and 90% Rank suggesting that the embeddings learnt are correct across wider classes compared to OWL2Vec* which performs better in Hits@10, Hits@100 and Median metrics showing that they are a bit less accurate but the embeddings learnt are concentrated in the same space. However, in case of Snomed CT, we see that EmEL-V performs better than OWL2Vec* on almost all the metrics. We notice a similar trend with our synthetically created ontology where EmEL-V performs better than all the other models suggesting that our model is able to learn embeddings to greater extent in larger ontologies like Snomed CT and OWL2Bench ontology. Hence, it shows our model's superior performance over rest of the models when trained on complete ontologies.

Later, we performed experiments by separating the OWL2Bench ontologies based on the type of relations, i.e., relations which are only one-one, one-many or many-many. This ablation study was done to check our claim that

Table 14
Performance on GO Inferred test data

Type of Relation	Model	Hits@1	Hits@10	Hits@100	Median Rank	90% Rank
one-one	**EmEL**	0.0764	0.2303	0.2766	4682.5	22155
	EmEL++	0.0000	0.0011	0.0033	22248	40241
	OWL2Vec*	0.0633	**0.2878**	**0.4916**	**113**	429599.5
	EmEL-V	**0.0832**	0.2802	0.3763	1551	**19464**
one-many	**EmEL**	0.0708	0.1638	0.2167	4294.5	22885.5
	EmEL++	0.0684	0.1665	0.2476	3268.5	21747.5
	OWL2Vec*	**0.0842**	**0.3897**	**0.6308**	**23**	434521.5
	EmEL-V	0.0811	0.2853	0.3799	1638	**19429**
many-many	**EmEL**	0.0537	0.1803	0.2325	4518.5	27585
	EmEL++	0.0193	0.0591	0.1099	7079.5	31260.5
	OWL2Vec*	**0.0970**	**0.4250**	**0.6532**	**19**	427221.5
	EmEL-V	0.0593	0.2487	0.3492	2777.5	**24728**

Table 15
Performance on SNOMED Inferred test data

Type of Relation	Model	Hits@1	Hits@10	Hits@100	Median Rank	90% Rank
one-one	**EmEL**	0.1178	0.3596	0.4045	7714	175313.9
	EmEL++	0.0000	0.0000	0.0003	154536.5	277698.8
	OWL2Vec*	0.1012	0.4729	**0.6693**	147	**240192**
	EmEL-V	**0.1377**	**0.4798**	0.6532	**12**	172843.7
one-many	**EmEL**	0.0000	0.0000	0.0003	156435.5	276631.8
	EmEL++	0.0000	0.0000	0.0003	153766.5	277090.1
	OWL2Vec*	0.0413	**0.2801**	0.3406	**405**	329202.9
	EmEL-V	**0.0528**	0.2519	**0.3659**	7161.5	**231117.4**
many-many	**EmEL**	0.0742	0.1763	0.2018	60204.5	248682.4
	EmEL++	0.0000	0.0000	0.0004	154558.5	276635.9
	OWL2Vec*	0.0905	0.1789	0.3411	**207**	300192.7
	EmEL-V	**0.0922**	**0.2818**	**0.3881**	9590.5	**238861.5**

EmEL-V has the ability to learn one-many and many-many types of relations. We noticed that while EmEL-V outperformed other models on only two metrics in case of ontologies containing only one-one relations, it had the best performance on four different metrics for one-to-many and on three metrics for many-to-many relations. Thus it shows the model's effectiveness on one-to-many and many-to-many relationship types.

We also do the ablation study on the three ontologies - GO, Galen and Snomed CT. Here, we noticed that while EmEL-V performs much better than EmEL and EmEL++, it's performance is only comparable to OWL2Vec* which is slightly better in most of the metrics. This is consistent with the previous finding that OWL2Vec* performs better on smaller ontologies and since we divided the ontologies to smaller ones based on the type of the relations, OWL2Vec* performs slightly better than EmEL-V.

Finally, we used inferred axioms obtained using classical logic based reasoners and tested the models on them. We notice that for each ontology, EmEL-V outperforms others in one-to-many and many-to-many relations across ontologies while other models perform better on one-to-one relations. This shows the ability of our model to generalise and reason over complex relations better than other models.

7. Conclusion and Future Work

The existing Knowledge Graph and ontology embedding approaches assume that relations are one-to-one. This limits the possibility of using these embeddings for more expressive ontologies and for complex reasoning tasks. We have provided a simple yet effective method that overcomes this obstacle and helps embeddings to capture many-to-many relations. Through our evaluation, we have shown that our model outperforms the existing geometry based embedding models in the class subsumption prediction task across three ontologies of varying sizes and characteristics. Its performance is also comparable with respect to *OWL2Vec** which is a word embedding and random walk based ontology embedding technique. Through our rigorous experimentation and various ablation studies, we are able to show the effect of the addition of variance on the learning of the model. We also notice the ability of the model to work better on larger ontologies where it exceeded the performance of the current state-of-the-art model *OWL2Vec**. The ability of the model to reason over complex relations is supported by the experiments on the inferred axioms. The technique that we described here, i.e., considering variance in relations, can be used in other knowledge graph embeddings such as TransE as well. The flexibility to model many-to-many relations also opens up the possibility of extending this work for more expressive description logics such as $\mathcal{SROIQ}$. Results show that EmEL-V successfully takes into account the many-to-many relations while creating the embeddings. The source code of EmEL-V and the ontologies used in the evaluation are publicly available at https://github.com/kracr/el-embeddings and https://doi.org/10.5281/zenodo.7023568 respectively.

References

[1] A. Steigmiller and B. Glimm, Pay-As-You-Go Description Logic Reasoning by Coupling Tableau and Saturation Procedures, *Journal of Artificial Intelligence Research* **54** (2015), 535–592. doi:10.1613/jair.4897.

[2] Y. Kazakov, M. Krötzsch and F. Simancik, The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with $\mathscr{EL}$ Ontologies, *Journal of Automated Reasoning* **53**(1) (2014), 1–61. doi:10.1007/s10817-013-9296-3.

[3] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur and Y. Katz, Pellet: A Practical OWL-DL Reasoner, *Journal of Web Semantics* **5**(2) (2007), 51–53–. doi:10.1016/j.websem.2007.03.004.

[4] B. Glimm, I. Horrocks, B. Motik, G. Stoilos and Z. Wang, HermiT: An OWL 2 Reasoner, *Journal of Automated Reasoning* **53**(3) (2014), 245–269. doi:10.1007/s10817-014-9305-1.

[5] N. Guarino, D. Oberle and S. Staab, *What Is an Ontology?*, in: *Handbook on Ontologies*, Springer Berlin Heidelberg, 2009, pp. 1–17. doi:10.1007/978-3-540-92673-3_0.

[6] G. Singh, R. Mutharaju and P. Kapanipathi (eds), Proceedings of the Semantic Reasoning Evaluation Challenge (SemREC 2021) co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual Event, October 27th, 2021, in *CEUR Workshop Proceedings*, Vol. 3123, CEUR-WS.org, 2022. http://ceur-ws.org/Vol-3123.

[7] M. Ebrahimi, A. Eberhart, F. Bianchi and P. Hitzler, Towards Bridging the Neuro-Symbolic Gap: Deep Deductive Reasoners, *Applied Intelligence* **51**(9) (2021), 6326–6348–. doi:10.1007/s10489-020-02165-6.

[8] M.K. Sarker, L. Zhou, A. Eberhart and P. Hitzler, Neuro-Symbolic Artificial Intelligence, *AI Commununications* **34**(3) (2021), 197–209–. doi:10.3233/AIC-210084.

[9] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, A.N. Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J.F. Sequeda, S. Staab and A. Zimmermann, Knowledge Graphs, *ACM Computing Surveys* **54**(4) (2021), 71:1–71:37. doi:10.1145/3447772.

[10] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Transactions on Knowledge and Data Engineering* **29**(12) (2017), 2724–2743. doi:10.1109/TKDE.2017.2754499.

[11] A. Rossi, D. Barbosa, D. Firmani, A. Matinata and P. Merialdo, Knowledge Graph Embedding for Link Prediction: A Comparative Analysis, *ACM Transactions on Knowledge Discovery from Data* **15**(2) (2021), 14:1–14:49. doi:10.1145/3424672.

[12] Z. Sun, Q. Zhang, W. Hu, C. Wang, M. Chen, F. Akrami and C. Li, A Benchmarking Study of Embedding-Based Entity Alignment for Knowledge Graphs, *Proceedings of the VLDB Endowment* **13**(12) (2020), 2326–2340–. doi:10.14778/3407790.3407828.

[13] P. Ristoski, J. Rosati, T.D. Noia, R.D. Leone and H. Paulheim, RDF2Vec: RDF graph embeddings and their applications, *Semantic Web* **10**(4) (2019), 721–752. doi:10.3233/SW-180317.

[14] J. Chen, F. Lécué, Y. Geng, J.Z. Pan and H. Chen, Ontology-guided Semantic Composition for Zero-shot Learning, in: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, D. Calvanese, E. Erdem and M. Thielscher, eds, 2020, pp. 850–854. doi:10.24963/kr.2020/87.

[15] J.Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li and J. Liu, Content Based Fake News Detection Using Knowledge Graphs, in: *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, D. Vrandecic, K. Bontcheva, M.C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L. Kaffee and E. Simperl, eds, Lecture Notes in Computer Science, Vol. 11136, Springer, 2018, pp. 669–683. doi:10.1007/978-3-030-00671-6_39.

[16] E.B. Myklebust, E. Jiménez-Ruiz, J. Chen, R. Wolf and K.E. Tollefsen, Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings, *Semantic Web* **13**(3) (2022), 299–338. doi:10.3233/SW-222804.

[17] X. Huang, J. Zhang, D. Li and P. Li, Knowledge Graph Embedding Based Question Answering, in: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, J.S. Culpepper, A. Moffat, P.N. Bennett and K. Lerman, eds, ACM, 2019, pp. 105–113. doi:10.1145/3289600.3290956.

[18] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, *Proceedings of the AAAI Conference on Artificial Intelligence* **28**(1) (2014). doi:10.1609/aaai.v28i1.8870. https://ojs.aaai.org/index.php/AAAI/article/view/8870.

[19] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: *Advances in Neural Information Processing Systems*, Vol. 26, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K.Q. Weinberger, eds, Curran Associates, Inc., 2013. doi:https://dl.acm.org/doi/10.5555/2999792.2999923. https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[20] B. Yang, W.-t. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, USA, Conference Track Proceedings* (2015). doi:10.48550/arXiv.1412.6575. http://arxiv.org/abs/1412.6575.

[21] M. Nickel, V. Tresp and H.-P. Kriegel, A Three-Way Model for Collective Learning on Multi-Relational Data, in: *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, Omnipress, Madison, WI, USA, 2011, pp. 809–816–. ISBN 9781450306195. doi:10.5555/3104482.3104584.

[22] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier and G. Bouchard, Complex Embeddings for Simple Link Prediction, in: *Proceedings of The 33rd International Conference on Machine Learning*, M.F. Balcan and K.Q. Weinberger, eds, Proceedings of Machine Learning Research, Vol. 48, PMLR, New York, New York, USA, 2016, pp. 2071–2080. doi:10.5555/3045390.3045609. http://proceedings.mlr.press/v48/trouillon16.html.

[23] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning Entity and Relation Embeddings for Knowledge Graph Completion, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, AAAI Press, 2015, pp. 2181–2187–. ISBN 0262511290. doi:10.1609/aaai.v29i1.9491.

[24] G. Schreiber and Y. Raimond, RDF 1.1 Primer, 2014. https://www.w3.org/TR/rdf11-primer/.

[25] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider and S. Rudolph, OWL 2 Web Ontology Language Primer (Second Edition), 2012. https://www.w3.org/TR/owl2-primer/.

[26] R. Gupta, A.Y. Halevy, X. Wang, S.E. Whang and F. Wu, Biperpedia: An Ontology for Search Applications, *Proc. VLDB Endow.* **7**(7) (2014), 505–516. doi:10.14778/2732286.2732288. http://www.vldb.org/pvldb/vol7/p505-gupta.pdf.

[27] M. Kulmanov, W. Liu-Wei, Y. Yan and R. Hoehndorf, EL Embeddings: Geometric Construction of Models for the Description Logic EL++, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 6103–6109. doi:10.24963/ijcai.2019/845.

[28] S. Mondal, S. Bhatia and R. Mutharaju, EmEL++: Embeddings for EL++ Description Logic, in: *Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021)*, Vol. 2846.

[29] J. Chen, P. Hu, E. Jimenez-Ruiz, O.M. Holter, D. Antonyrajah and I. Horrocks, "OWL2Vec*: embedding of OWL ontologies", in: *"Machine Learning"*, "Springer International Publishing", "Cham", "2021", pp. "81–96". doi:"https://doi.org/10.1007/s10994-021-05997-6".

[30] G. Singh, S. Mondal, S. Bhatia and R. Mutharaju, Neuro-Symbolic Techniques for Description Logic Reasoning (Student Abstract), *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(18) (2021), 15891–15892. https://ojs.aaai.org/index.php/AAAI/article/view/17942.

[31] M. Krötzsch, F. Simancik and I. Horrocks, A Description Logic Primer, 2013.

[32] G. Singh, S. Bhatia and R. Mutharaju, OWL2Bench: A Benchmark for OWL 2 Reasoners, in: *The Semantic Web – ISWC 2020*, J.Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne and L. Kagal, eds, Springer International Publishing, Cham, 2020, pp. 81–96. ISBN 978-3-030-62466-8. doi:10.1007/978-3-030-62466-8_6.

[33] A. Grover and J. Leskovec, Node2vec: Scalable Feature Learning for Networks, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 855–864–. ISBN 9781450342322. doi:10.1145/2939672.2939754.

[34] X. Peng, G. Chen, C. Lin and M. Stevenson, Highly Efficient Knowledge Graph Embedding Learning with Orthogonal Procrustes Analysis, in: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, 2021, pp. 2364–2375. doi:10.18653/v1/2021.naacl-main.187. https://aclanthology.org/2021.naacl-main.187.

[35] I. Balazevic, C. Allen and T. Hospedales, TuckER: Tensor Factorization for Knowledge Graph Completion, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5185–5194. doi:10.18653/v1/D19-1522. https://aclanthology.org/D19-1522.

[36] F.Z. Smaili, X. Gao and R. Hoehndorf, Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations, *Bioinformatics* **34**(13) (2018), i52–i60. doi:10.1093/bioinformatics/bty259.

[37] Lütfü Özçep, M. Leemhuis and D. Wolter, Cone Semantics for Logics with Negation, in: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, ed., International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 1820–1826, Main track. doi:10.24963/ijcai.2020/252.

[38] P. Hitzler, M. Krötzsch and S. Rudolph, *Foundations of Semantic Web Technologies*, 1st edn, Chapman & Hall/CRC, 2009. ISBN 9781420090505.

[39] F. Baader, S. Brandt and C. Lutz, Pushing the EL Envelope, in: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, pp. 364–369–. doi:10.5555/1642293.1642351.

[40] A. Rector, J. Rogers and P. Pole, The GALEN high level ontology, *Studies in Health Technology and Informatics* **34** (1996), 174–178. doi:10.3233/978-1-60750-878-6-174.

[41] G.O. Consortium, The Gene Ontology (GO) database and informatics resource, *Nucleic Acids Research* **32** (2004), 258–261. doi:10.1093/nar/gkh036.

[42] K. Donnelly, SNOMED-CT: The advanced terminology and coding system for eHealth, *Studies in health technology and informatics* **121** (2006), 279–90.

[43] S. Bhatia, B. Chatterjee, D. Nathani and M. Kaul, A Persistent Homology Perspective to the Link Prediction Problem, in: *Complex Networks and Their Applications VIII - Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019, Lisbon, Portugal, December 10-12, 2019*, H. Cherifi, S. Gaito, J.F. Mendes, E. Moro and L.M. Rocha, eds, Studies in Computational Intelligence, Vol. 881, Springer, 2019, pp. 27–39. doi:10.1007/978-3-030-36687-2_3.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds, Curran Associates, Inc., 2019, pp. 8024–8035. doi:https://doi.org/10.48550/arXiv.1912.01703. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[45] M. Ali, H. Jabeen, C.T. Hoyt and J. Lehman, The KEEN Universe: An Ecosystem for Knowledge Graph Embeddings with a Focus on Reproducibility and Transferability, in: *International Semantic Web Conference, Springer*, 2019, pp. 3–18. doi:10.1007/978-3-030-30796-7_1.

[46] V. Golovko, A. Kroshchanka, M. Kovalev, V. Taberko and D. Ivaniuk, Neuro-Symbolic Artificial Intelligence: Application for Control the Quality of Product Labeling, in: *Open Semantic Technologies for Intelligent System*, V. Golenkov, V. Krasnoproshin, V. Golovko and E. Azarov, eds, Springer International Publishing, Cham, 2020, pp. 81–101. ISBN 978-3-030-60447-9. doi:10.1007/978-3-030-60447-9_6.

[47] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali and A. ten Teije, Modular Design Patterns for Hybrid Learning and Reasoning Systems: a taxonomy, patterns and use cases, Vol. 51, pp. 6528–6546. doi:10.1007/s10489-021-02394-3.