EL Embeddings: Geometric Construction of Models for the Description Logic $\mathcal{EL}^{++}$

Maxat Kulmanov[1] , **Wang Liu-Wei**[1] , **Yuan Yan**[2] and **Robert Hoehndorf**[1*]
[1]Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), Computational Bioscience Research Center (CBRC), King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
[2]Department of Mathematics & Statistics, Dalhousie University, Halifax, Nova Scotia, Canada
{maxat.kulmanov, liuwei.wang, robert.hoehndorf}@kaust.edu.sa, yuan.yan@dal.ca

Abstract

An embedding is a function that maps entities from one algebraic structure into another while preserving certain characteristics. Embeddings are being used successfully for mapping relational data or text into vector spaces where they can be used for machine learning, similarity search, or similar tasks. We address the problem of finding vector space embeddings for theories in the Description Logic $\mathcal{EL}^{++}$that are also models of the TBox. To find such embeddings, we define an optimization problem that characterizes the model-theoretic semantics of the operators in $\mathcal{EL}^{++}$within $\mathbb{R}^n$, thereby solving the problem of finding an interpretation function for an $\mathcal{EL}^{++}$theory given a particular domain Δ. Our approach is mainly relevant to large $\mathcal{EL}^{++}$theories and knowledge bases such as the ontologies and knowledge graphs used in the life sciences. We demonstrate that our method can be used for improved prediction of protein–protein interactions when compared to semantic similarity measures or knowledge graph embeddings.

1 Introduction

There has been a recent proliferation of methods that generate “embeddings” for different types of entities. Often, these embeddings are functions that map entities within a certain structure into a vector space $\mathbb{R}^n$ such that a set of structural characteristics of the original structure are preserved within the vector space. For example, word embeddings are generated for words within a corpus of text based on the distribution of words and their co-mentions [Mikolov *et al.*, 2013].

Knowledge graph embeddings are used to project sets of discrete facts into a vector space over real numbers and aim to preserve some structural properties of the graph within $\mathbb{R}^n$ [Nickel *et al.*, 2016; Wang *et al.*, 2017]. The embeddings can project entities and relations within a knowledge graph into $\mathbb{R}^n$ such that they can naturally be used as features for machine learning tasks such as classification, regression, or clustering, or directly utilize similarity measures within $\mathbb{R}^n$ for determining semantic similarity, performing reasoning by analogy, and thereby predict relations. Most knowledge graph embedding approaches find the embedding function through optimization with respect to an objective function and, optionally, a set of constraints [Wang *et al.*, 2017].

Inference in knowledge graphs is often limited to composition of relations. Model-theoretic languages such as Description Logics can also be used to express relational knowledge while adding operators that cannot easily be expressed in graph-based form (quantifiers, negation, conjunction, disjunction) [Baader, 2003]. In particular the life sciences have developed a large number of ontologies formulated in the Web Ontology Language (OWL) [Grau *et al.*, 2008], and many of the life science ontologies fall in the OWL 2 EL profile [Hoehndorf *et al.*, 2011], which is based on the Description Logics $\mathcal{EL}^{++}$[Motik *et al.*, 2009]. The life science ontologies are used to express domain knowledge and serve as a foundation for analysis and interpretation of biological data, for example through statistical measures [Subramanian *et al.*, 2005] or semantic similarity measures [Pesquita *et al.*, 2009]. Recently, “ontology embeddings” were developed for life science ontologies that map classes, relations, and instances in these ontologies into a vector space while preserving certain syntactic properties of the ontology axioms and their deductive closure [Smaili *et al.*, 2018]. However, embeddings that rely primarily on preserving syntactic properties of knowledge bases within a vector space are limited by the kind of inferences that can be precomputed and expressed in the knowledge representation language, and do not utilize prior knowledge about the semantics of operators during the search for an embedding function.

Here, we introduce EL Embeddings, a method to generate embeddings for ontologies in the Description Logic $\mathcal{EL}^{++}$. EL Embeddings explicitly generate – or approximate – models for an $\mathcal{EL}^{++}$theory and therefore approximate the interpretation function. For this purpose, we formulate the problem of finding a model as an optimization problem over $\mathbb{R}^n$. An alternative view on EL Embeddings is that we extend knowledge graph embeddings with the semantics of conjunction, existential quantification, and the bottom concept.

We demonstrate that the resulting embeddings can be used for determining semantic similarity or suggest axioms that may be entailed by the theory. As large $\mathcal{EL}^{++}$theories are

[*]Contact Author

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
generalized concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
instantiation	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

Table 1: Syntax and semantic of $\mathcal{EL}^{++}$ (omitting role inclusions and concrete domains).

mainly used in the life sciences, we evaluate our approach on a large knowledge base of protein–protein interactions and protein functions. We show that our method can improve the prediction of protein–protein interactions when compared to semantic similarity measures and to knowledge graph embeddings.

2 Related Work

2.1 The Description Logic $\mathcal{EL}^{++}$and Its Application in Life Sciences

The Description Logic $\mathcal{EL}^{++}$ [Baader *et al.*, 2005] is a Description Logic for which subsumption can be decided in polynomial time and which is therefore suitable for representing and reasoning over large ontologies. The syntax and semantics of $\mathcal{EL}^{++}$ is summarized in Table 1 (omitting concrete domains which we will not consider here). $\mathcal{EL}^{++}$also forms the basis of the OWL 2 EL profile of OWL [Motik *et al.*, 2009].

The ABox axioms (instantiation and role assertion) in $\mathcal{EL}^{++}$can be eliminated by replacing $C(a)$ with $\{a\} \sqsubseteq C$ and $r(a,b)$ with $\{a\} \sqsubseteq \exists r.\{b\}$, and every $\mathcal{EL}^{++}$TBox can be normalized into one of four normal forms: $C \sqsubseteq D$, $C \sqcap D \sqsubseteq E$, $\exists R.C \sqsubseteq D$, and $C \sqsubseteq \exists R.D$ (where the bottom concept can only appear on the right-hand side and only in the first three normal forms) [Baader *et al.*, 2005].

$\mathcal{EL}^{++}$ is widely used to represent and reason over life science ontologies such as the Gene Ontology [Ashburner *et al.*, 2000], the Human Phenotype Ontology [Zemojtel *et al.*, 2016], or SNOMED CT [Schulz *et al.*, 2009]. These ontologies are often large and require fast decision procedures for automated reasoning, which $\mathcal{EL}^{++}$can provide [Baader *et al.*, 2005]. The ontologies in the life-science domain are also used as components in knowledge graphs to structure data and provide background knowledge about classes within their domains.

2.2 Knowledge Graph Embeddings

Knowledge graph embedding methods have been developed to map entities and their relations expressed in a knowledge graph into a vector space while preserving relational and other semantic information under certain vector space relations [Nickel *et al.*, 2016]. Translation-based embeddings, such as TransE [Bordes *et al.*, 2013], generate vector space representations of entities and relations in a graph such that $\mathbf{a} + \mathbf{r} \approx \mathbf{b}$ if $r(a,b)$ is a relation in the knowledge graph. Other approaches include methods for exploring the neighborhood of nodes in the graph and encoding these nodes and their relations [Wang *et al.*, 2017].

Knowledge graphs are heterogeneous graphs with an explicit semantics and an inference relation; one way in which the semantics of relations in a knowledge graph can be taken into account when generating knowledge graph embeddings is by pre-computing a limited form of deductive closure on the graph before finding the embeddings [Nickel *et al.*, 2016; Wang *et al.*, 2017]. Such an approach has also been applied successfully in the life sciences where knowledge graph embeddings based on deductively closed graphs have been used for predicting gene–disease associations or drug targets [Alshahrani *et al.*, 2017].

2.3 Semantic Similarity

A related yet alternative approach to using knowledge graph embeddings for relational learning is the use of semantic similarity measures to compare two classes within an ontology, or two instances with respect to the axioms within an ontology [Pesquita *et al.*, 2009]. There is a wide range of semantic similarity measures, most of which operate on graphs or sets constructed from a theory syntactically (e.g., by applying a certain closure on a theory to generate graphs) but can also be applied to model structures such as the canonical models of $\mathcal{ALC}$ theories [Harispe *et al.*, 2015].

In life sciences, semantic similarity measures can be applied predictively [Pesquita *et al.*, 2009]; ontologies provide biological features, and similarity between the biological features can be indicative of an underlying biological relation. For example, semantic similarity between proteins linked to functions in the Gene Ontology [Ashburner *et al.*, 2000] can be used to determine protein–protein interactions based on the biological assumption that interacting proteins are likely to have similar functions [Kulmanov and Hoehndorf, 2017]; similarly, semantic similarity measures are used to identify candidate genes associated with diseases [Albrecht and Schlicker, 2007]. Widely-applied semantic similarity measures in life sciences include Resnik's similarity [Resnik, 1995] or the weighted Jaccard index [Pesquita *et al.*, 2009]. Recently, semantic similarity is also measured based on knowledge graph embeddings, for example for predicting protein–protein interactions [Smaili *et al.*, 2018].

3 Geometric Models for $\mathcal{EL}^{++}$

3.1 Relation Model and Normalization

Our aim is to extend knowledge graph embeddings so that they incorporate the $\mathcal{EL}^{++}$operators (conjunction, existential quantification) and can express the bottom concept $\bot$. We use a relational embedding model, TransE [Bordes *et al.*, 2013], to map relations into $\mathbb{R}^n$. We chose TransE due to its simplicity; however, our method can accommodate different relational models.

Let $O = (\mathbb{C}, \mathbb{R}, \mathbb{I}; ax)$ be an $\mathcal{EL}^{++}$ontology consisting of a set of class symbols $\mathbb{C}$, relation symbols $\mathbb{R}$, individual symbols $\mathbb{I}$, and a set of axioms ax. We first transform ax into a normal form following [Baader *et al.*, 2005]; we eliminate the ABox by replacing each individual symbol with a singleton class and rewriting relation assertions $r(a, b)$ and class assertions $C(a)$ as $\{a\} \sqsubseteq \exists r.\{b\}$ and $\{a\} \sqsubseteq C$. Using the conversion rules in [Baader *et al.*, 2005] we transform the set of axioms into one of four forms where $C, D, E \in \mathbb{C}$ and $R \in \mathbb{R}$: $C \sqsubseteq D$; $C \sqcap D \sqsubseteq E$; $C \sqsubseteq \exists R.D$; $\exists R.C \sqsubseteq D$.

3.2 Objective Functions

If an $\mathcal{EL}^{++}$theory T has a model then it also has an infinite model, and therefore it also has a model with a universe of $\mathbb{R}^n$ for any n (Löwenheim–Skolem upwards) [Barwise and Etchemendy, 2002]. The embedding function our model aims to find is intended to approximate the interpretation function $\mathcal{I}$ in the $\mathcal{EL}^{++}$semantics (Table 1). Specifically, our embedding function η aims to map each class C to an open n-ball in $\mathbb{R}^n$ $(\eta(C))$ and every binary relation r to a vector in $\mathbb{R}^n$. We define a geometric ontology embedding η as a pair (f_η, r_η) of functions that map classes and relations in O into $\mathbb{R}^n$, $f_\eta : C \cup R \mapsto \mathbb{R}^n$ and $r_\eta : C \mapsto \mathbb{R}$. The function $f_\eta(x)$ maps a class to its center or maps a relation to its embedding vector, and $r_\eta(x)$ maps a class x to the radius associated with it.

We formulate one loss function for each of the normal forms so that the embedding η preserves the semantics of $\mathcal{EL}^{++}$geometrical within $\mathbb{R}^n$. The total loss for finding η is the sum of the loss functions for all the normal forms. We first assume that none of the classes are $\bot$. The first loss function (Eqn. 1) aims to capture the notion that, if $C \sqsubseteq D$, then $\eta(C)$ should lie in $\eta(D)$. For all loss functions we use a margin parameter γ; if $\gamma < 0$ then $\eta(C)$ lies properly inside $\eta(D)$. Also, we add normalization loss for all class embeddings in the loss functions, essentially moving the centers of all n-balls representing classes to lie on the unity sphere.

$$\begin{aligned} & loss_{C \sqsubseteq D}(c, d) = \\ & \max(0, \|f_\eta(c) - f_\eta(d)\| + r_\eta(c) - r_\eta(d) - \gamma) \\ & + |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1| \end{aligned} \tag{1}$$

The loss function for the second normal form (Eqn. 2), $C \sqcap D \sqsubseteq E$, should capture the notion that the intersection or overlap of the n-balls representing C and D should lie within the n-ball representing E; while the overlap between $\eta(C)$ and $\eta(D)$ is not in general an n-ball, the loss should characterize the smallest n-ball which includes the intersection of $\eta(C)$ and $\eta(D)$ and minimizes its non-overlap with $\eta(E)$. Let $h = \frac{r_\eta(c)^2 - r_\eta(d)^2 + \|f_\eta(c) - f_\eta(d)\|^2}{2\|f_\eta(c) - f_\eta(d)\|}$, then the center and radius of the smallest n-ball containing the intersection of $\eta(C)$ and $\eta(D)$ are $f_\eta(c) + \frac{h}{\|f_\eta(c) - f_\eta(d)\|}(f_\eta(d) - f_\eta(c))$ and $\sqrt{r_\eta(c)^2 - h^2}$, respectively. However, we found it difficult to implement this loss due to very large gradients and therefore use the approximation of this loss given in Eqn. 2. The first term in Eqn. 2 is a penalty when the n-balls representing C and D are disjoint; the second and third terms force the center of $\eta(E)$ to lie inside the intersection of $\eta(C)$ and $\eta(D)$; the fourth term makes the radius of $\eta(E)$ to be larger than the radius of the smallest n-balls of the intersecting classes; this radius is strictly larger than the radius of the smallest n-ball containing the intersection and therefore satisfies the condition that the intersection should lie within $\eta(E)$.

$$\begin{aligned} & loss_{C \sqcap D \sqsubseteq E}(c, d, e) = \\ & \max(0, \|f_\eta(c) - f_\eta(d)\| - r_\eta(c) - r_\eta(d) - \gamma) \\ & + \max(0, \|f_\eta(c) - f_\eta(e)\| - r_\eta(c) - \gamma) \\ & + \max(0, \|f_\eta(d) - f_\eta(e)\| - r_\eta(c) - \gamma) \\ & + \max(0, \min(r_\eta(c), r_\eta(d)) - r_\eta(e) - \gamma) \\ & + |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1| + |\,\|f_\eta(e)\| - 1| \end{aligned} \tag{2}$$

The first two normal forms do not include any quantifiers or relations. Every point that lies properly within an n-ball representing a class is a potential instance of that class, and we apply relations as transformations on these points (following the TransE relation model). Therefore, relations are transformations on n-balls. Equations 3 and 4 capture this intention.

$$\begin{aligned} & loss_{C \sqsubseteq \exists R.D}(c, d, r) = \\ & \max(0, \|f_\eta(c) + f_\eta(r) - f_\eta(d)\| + r_\eta(c) - r_\eta(d) - \gamma) \\ & + |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1| \end{aligned} \tag{3}$$

$$\begin{aligned} & loss_{\exists R.C \sqsubseteq D}(c, d, r) = \\ & \max(0, \|f_\eta(c) - f_\eta(r) - f_\eta(d)\| - r_\eta(c) - r_\eta(d) - \gamma) \\ & + |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1| \end{aligned} \tag{4}$$

In the normal forms for $\mathcal{EL}^{++}$, $\bot$ can only occur on the right-hand side in three of the normal forms. We formulate separate loss functions for the cases in which $\bot$ appear. First, $C \sqcap D \sqsubseteq \bot$ states that C and D are disjoint and therefore $\eta(C)$ and $\eta(D)$ should not overlap. Equation 5 captures disjointness loss.

$$\begin{aligned} & loss_{C \sqcap D \sqsubseteq \bot}(c, d, e) = \\ & \max(0, r_\eta(c) + r_\eta(d) - \|f_\eta(c) - f_\eta(d)\| + \gamma) \\ & + |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1| \end{aligned} \tag{5}$$

Loss 6 captures the intuition that a class is unsatisfiable by minimizing the radius r_η of the class.

$$loss_{C \sqsubseteq \bot}(c) = r_\eta(c) \tag{6}$$

Since we use TransE as model for relations, the radius of an n-ball cannot change after a transformation by a relation. Therefore, we use the same loss (Eqn. 7) for $\exists R.C \sqsubseteq \bot$.

$$loss_{\exists R.C \sqsubseteq \bot}(c, r) = r_\eta(c) \tag{7}$$

While our model does not need negatives, we can use negatives as in translating embeddings to improve predictive performance. For this purpose, we add an optional loss for $C \not\sqsubseteq \exists R.D$ as in Equation 8.

$$\begin{aligned} & loss_{C \not\sqsubseteq \exists R.D}(c, d, r) = \\ & \max(0, r_\eta(c) + r_\eta(d) - \|f_\eta(c) + f_\eta(r) - f_\eta(d)\| + \gamma) \\ & + |\,\|f_\eta(c)\| - 1| + |\,\|f_\eta(d)\| - 1| \end{aligned} \tag{8}$$

Finally, we add the constraints $r_\eta(\top) = \infty$ to capture the intuition that the interpretation of $\top$ is $\Delta^\mathcal{I} = \mathbb{R}^n$, and $r_\eta(x) \geq 0$ for all x.

3.3 Embeddings and Models

Theorem 1 (Correctness)**.** *Let T be a theory in $\mathcal{EL}^{++}$. If $\gamma \leq 0$ and $loss_n(\eta(T)) = 0$ then T has a model.*

We outline a proof of this theorem. First, we set $\Delta = \mathbb{R}^n$. By construction, $\top = \Delta^\mathcal{I} = \mathbb{R}^n$. We interpret each class C as the set of points lying within the open n-ball $\eta(C)$, $C^\mathcal{I} = \{x \in \mathbb{R}^n | \, \|f_\eta(C) - x\| < r_\eta(C)\}$ and every binary relation r as a set of tuples $r^\mathcal{I} = \{(x,y) | x + f_\eta(r) = y\}$. We need to show that the conditions in Table 1 are satisfied if $loss(T) = 0$. The loss is the sum of losses for the four normal forms, all of which are non-negative.

The remaining conditions are preserved for each of the four normal forms and their respective losses: by construction, normal form 1 ensures that, if $C \sqsubseteq D$ is in the TBox, then $C^\mathcal{I} \subseteq D^\mathcal{I}$; the loss of normal form 2, $C \sqcap D \sqsubseteq E$, constructs the smallest n-ball containing the intersection of $\eta(C)$ and $\eta(D)$ and ensures that this n-ball lies within the $\eta(E)$; normal form 3, $C \sqsubseteq \exists R.D$, applies a relation transformation to all instances x of C (i.e., it constructs $f_\eta(x) + f_\eta(R)$ for all elements x of the n-ball $\eta(C)$) and ensures that each instance of C lies within $\eta(D)$, therefore ensuring that $\{x \in \mathbb{R}^n | \, \|f_\eta(C) - x\| < r_\eta(C)\} \subseteq \{x \in \mathbb{R}^n | \, \|f_\eta(D) + f_\eta(R) - x\| < r_\eta(D)\}$ and therefore $\eta(C) \subseteq \{x \in \mathbb{R}^n | \exists y \in \Delta^\mathcal{I} : (x,y) \in R^\mathcal{I} \wedge y \in D^\mathcal{I}\}$; normal form 4 trivially satisfies $\exists R.C \sqsubseteq D$. It follows similarly from the loss functions 5–7 that $\bot$ is interpreted as $\emptyset$; the only case requiring more attention is $C \sqcap D \sqsubseteq \bot$ where it is possible that the hyperspheres bounding the n-balls $\eta(C)$ and $\eta(D)$ touch. In our interpretation, we assume that hyperballs are open so that the n-balls are disjoint even if their bounding hyperspheres touch.

3.4 Training and Implementation

While our algorithm can find, or approximate, a model without any negative samples for any of the four normal forms, these models are usually underspecified. We intend to use our embeddings for relational learning which benefits from a representation in which asserted and implied axioms can be discriminated from those that should not hold true. Therefore, we follow a similar strategy for sampling negatives as in TransE and randomly generate corrupted axioms in third normal form ($C \sqsubseteq \exists R.D$) by replacing either C or D with a class C' or D' such that neither $C' \sqsubseteq \exists R.D$ nor $C \sqsubseteq \exists R.D'$ are asserted axioms in the ontology.

We randomly initialize the embeddings for classes and relations. We then sample formulas for each loss function in mini-batches and update the embeddings with respect to the sum of the loss functions (see Algorithm 1). We implement the algorithm in two parts. First, the processing of ontologies in OWL format and normalization into the $\mathcal{EL}^{++}$normal forms are performed using the OWL API and the APIs provided by the jCel reasoner which implements the $\mathcal{EL}^{++}$normalization rules [Mendez, 2012]. Training of embeddings and optimization is done using Python and the TensorFlow library, and we use the Adam optimizer [Kingma and Ba, 2014] for updating embeddings[1].

The runtime of the training process is linear in the number of axioms in the input ontology while inference time is linear in the number of all entities. The first step in the training process is eliminating the ABox by creating singleton classes for instances and transforming axioms into normal forms. This transformation can be performed in linear time with respect to the size of the TBox [Baader *et al.*, 2005]. The second step is to compute loss functions for all axioms in our training data. Computing the loss functions is linear in the size of embeddings (which is a constant) and we compute and optimize loss functions for all axioms every epoch. Consequently, training runtime complexity is $\mathrm{O}\big(\text{epochs} \cdot n \cdot m\big)$ where n is the number of all axioms, m is the embedding size, and epochs is the number of training iterations. In order to do inference for a query, we need to compute our similarity function for all entities. Therefore, inference time complexity is linear in the number of entities and the similarity function is computed in $\mathrm{O}\big(m\big)$ (with the embedding size m).

Algorithm 1: Algorithm used for training EL Embeddings

```
input : An ontology $O = (C, R, I; ax)$ in OWL
        format; margin $\gamma$; dimension $n$; epochs
        $epochs$; batchsize $bs$
output: embeddings $(f_\eta, r_\eta)$
// eliminate ABox
$C \leftarrow C \cup \{a\}$ for each $a \in I$
$ax \leftarrow ax \cup (\{a\} \sqsubseteq \exists r.\{b\})$ for each $r(a,b) \in ax$
$ax \leftarrow ax \cup (\{a\} \sqsubseteq C)$ for each $C(a) \in ax$
// apply $\mathcal{EL}^{++}$normalization rules
$(ax_{nf1}, ax_{nf2}, ax_{nf3}, ax_{nf4}) = normalize(ax)$
// separate axioms with $\bot$ for NF 1, 2 and 4
$(ax_{nf1}, ax_{bot1}) = separate(ax_{nf1})$
$(ax_{nf2}, ax_{bot2}) = separate(ax_{nf3})$
$(ax_{nf4}, ax_{bot4}) = separate(ax_{nf4})$
// generate negatives for proteins in NF 3
$neg_{nf3} = negatives(ax_{nf3})$
$D \leftarrow \{ax_{nf1}\} \cup \{ax_{nf2}\} \cup \{ax_{nf3}\} \cup \{ax_{nf4}\} \cup$
    $\{ax_{bot1}\} \cup \{ax_{bot2}\} \cup \{ax_{bot4}\} \cup \{neg_{nf3}\}$
// initialize embeddings
$f_\eta(c) = uniform(0,1)$ for each $c \in C$
$r_\eta(c) = uniform(0,1)$ for each $c \in C$
$f_\eta(r) = uniform(0,1)$ for each $r \in R$
for $e \in epochs$ do
    // Randomly sample a minibatch of size bs for each
       loss type
    $(s_{nf1}, s_{nf2}, s_{nf3}, s_{nf4}, s_{bot1}, s_{bot2}, s_{bot4}, s_{neg}) =$
       $sample(D, bs)$
    // Update embeddings w.r.t.
    $\sum \nabla loss(s_{nf1}, s_{nf2}, s_{nf3}, s_{nf4}, s_{bot1}, s_{bot2},$
       $s_{bot4}, s_{neg})$
end
```

[1] All code is freely available on https://github.com/bio-ontology-research-group/el-embeddings/

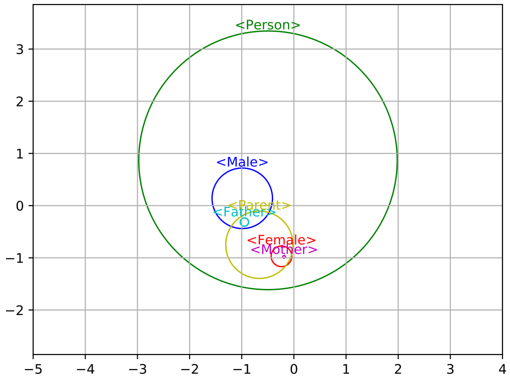

Figure 1: Visualization of embeddings in the family domain example.

4 Experiments

4.1 Example: Family Domain

We first construct a simple test knowledge base to test our model. We use the family domain in which we generate a knowledge base that contains examples for each of the normal forms (Eqn. 9–20). We chose a margin $\gamma = 0$ and an embedding dimension of 2 so that we can visualize the generated embeddings in $\mathbb{R}^2$. Figure 1 shows the resulting embeddings.

$$Male \sqsubseteq Person \tag{9}$$

$$Female \sqsubseteq Person \tag{10}$$

$$Father \sqsubseteq Male \tag{11}$$

$$Mother \sqsubseteq Female \tag{12}$$

$$Father \sqsubseteq Parent \tag{13}$$

$$Mother \sqsubseteq Parent \tag{14}$$

$$Female \sqcap Male \sqsubseteq \bot \tag{15}$$

$$Female \sqcap Parent \sqsubseteq Mother \tag{16}$$

$$Male \sqcap Parent \sqsubseteq Father \tag{17}$$

$$\exists hasChild.Person \sqsubseteq Parent \tag{18}$$

$$Parent \sqsubseteq Person \tag{19}$$

$$Parent \sqsubseteq \exists hasChild.\top \tag{20}$$

4.2 Protein–Protein Interactions

Prediction of interactions between proteins is a common task in molecular biology that relies on information about sequences as well as functional information [Pesquita *et al.*, 2009; Kulmanov and Hoehndorf, 2017]. The information about the functions of proteins is represented through the Gene Ontology (GO) [Ashburner *et al.*, 2000], a large manually-created ontology with over 45,000 classes and 100,000 axioms. GO can be formalized in OWL 2 EL and therefore falls in the $\mathcal{EL}^{++}$formalism [Golbreich and Horrocks, 2007]. Common approaches to predicting protein–protein interactions (PPIs) include network-based approaches and the use of semantic similarity measures [Kulmanov and Hoehndorf, 2017].

We use the PPI dataset provided by the STRING database [Roth *et al.*, 2016] to construct a knowledge graph of proteins and their interactions. We construct two graphs for human and yeast organisms with relations for which a confidence score of 700 or more is assigned in STRING (following recommendations in STRING [Roth *et al.*, 2016]); if an interaction between two proteins P_1 and P_2 exists in STRING, we assert $interacts(P_1, P_2)$. We further add the associations of proteins with functions from the GO, provided by STRING, together with all classes and relations from GO. For this we use two representation patterns. First, we generate an OWL representation in which proteins are instances, and if protein P is associated with the function F we add the axiom $\{P\} \sqsubseteq \exists hasFunction.F$ (based on the ABox axiom $(\exists hasFunction.F)(P)$). We use this information together with the native OWL version of GO provided by the OBO Foundry repository [Smith *et al.*, 2007]; when applying knowledge graph embeddings to this representation, we use the RDF serialization of the complete OWL knowledge base as the knowledge graph. While the OWL-based representation is suitable for our EL Embeddings, knowledge graph embeddings and semantic similarity measures would benefit from a graph-based representation. We therefore create a second representation in which we replace all axioms of the type $X \sqsubseteq \exists R.Y$ in GO with a relation $R(X, Y)$, and link proteins to their functions using a $hasFunction$ relation (i.e., if protein P has function F, we assert $R(P, F)$).

We generate a training, testing, and validation split (80%/10%/10%) from interaction pairs of proteins. We use the TransE [Bordes *et al.*, 2013] implementation in the PyKEEN framework [Jabeen *et al.*, 2019] on both representations (native OWL/RDF and the “plain” representation) to generate knowledge graph embeddings and use them for link prediction. We implement two semantic similarity measures, Resnik’s similarity [Resnik, 1995] and Lin’s similarity [Harispe *et al.*, 2015], together with the best-match average strategy for combining pairwise class similarities [Pesquita *et al.*, 2009; Harispe *et al.*, 2015], and compute the similarity between proteins based on their associations with GO classes. To predict PPIs with EL Embeddings, we predict whether axioms of the type $\{P\} \sqsubseteq \exists hasFunction.\{F\}$ hold. We use the similarity-based function in Eqn. 21 for this prediction.

$$\begin{aligned} sim(c, r, d) = -\max(0, &\|f_\eta(c) + f_\eta(r) - f_\eta(d)\| \\ &-r_\eta(c) - r_\eta(d) - \gamma) \end{aligned} \tag{21}$$

We evaluate the predictive performance based on recall at rank 10, rank 100, mean rank and area under the ROC curve using our validation set. In our experiments, we perform an extensive search for optimal parameters for TransE and our EL Embeddings, testing embeddings sizes of 50, 100, 200, and 400. We also evaluate the performance with different margin parameters γ, using -0.1, -0.01, 0, 0.01, and 0.1. The optimal set of parameters for EL Embeddings are $embedding\_size = 50$ and $\gamma = -0.1$. For TransE (plain) $embedding\_size = 50$ (human) and 100 (yeast), for TransE (RDF) $embedding\_size = 400$ (human) and 200 (yeast)[2].

[2] Detailed results are available at https://www.dropbox.com/s/wresfh9fkfah4ei/supplement.pdf?dl=0.

Method	Raw Hits@10	Filtered Hits@10	Raw Hits@100	Filtered Hits@100	Raw Mean Rank	Filtered Mean Rank	Raw AUC	Filtered AUC
TransE (RDF)	0.03	0.05	0.22	0.27	855	809	0.84	0.85
TransE (plain)	0.06	0.13	0.41	0.54	378	330	0.93	0.94
SimResnik	0.08	0.18	0.38	0.49	713	663	0.87	0.88
SimLin	0.08	0.17	0.34	0.45	807	756	0.85	0.86
EL Embeddings	**0.10**	**0.23**	**0.50**	**0.75**	**247**	**187**	**0.96**	**0.97**

Table 2: Prediction performance for yeast protein–protein interactions.

Method	Raw Hits@10	Filtered Hits@10	Raw Hits@100	Filtered Hits@100	Raw Mean Rank	Filtered Mean Rank	Raw AUC	Filtered AUC
TransE (RDF)	0.02	0.03	0.12	0.16	2262	2189	0.85	0.85
TransE (plain)	0.05	0.11	0.32	0.44	809	737	0.95	0.95
SimResnik	0.05	0.10	0.23	0.28	2549	2475	0.83	0.83
SimLin	0.04	0.08	0.19	0.22	2818	2743	0.81	0.82
EL Embeddings	**0.09**	**0.22**	**0.43**	**0.70**	**658**	**572**	**0.96**	**0.96**

Table 3: Prediction performance for human protein–protein interactions.

We report results on our testing set in Table 2 for the yeast PPI dataset and in Table 3 for the human PPI dataset.

For a query interaction $interacts(P_1, P_2)$ we predict interactions of P_1 to all proteins from our training set and identify the rank of P_2. Then we compute the mean of ranks for all interactions in our testing set. We refer to this result as *raw mean rank*. Since the interactions from our training and validation set will rank higher than those in our testing set, we perform this evaluation excluding training and validation interactions and report them as *Filtered*. We further report the area under the ROC curve (AUC) which is more commonly used for evaluating PPI predictions [Kulmanov and Hoehndorf, 2017; Alshahrani *et al.*, 2017].

5 Discussion

Knowledge graph embeddings and other forms of relational learning methods are increasingly applied in scientific tasks such as prediction of protein–protein interactions, gene–disease associations, or drug targets. Our work on EL Embeddings is motivated by the need to incorporate background knowledge into machine learning tasks. This need exists in particular in scientific domains in which large formal knowledge bases have been created that can be utilized to constrain optimization or improve search. Our embeddings are based on the Description Logic $\mathcal{EL}^{++}$which is widely used in life science ontologies [Smith *et al.*, 2007] and combined with biological knowledge graphs [Alshahrani *et al.*, 2017]. EL Embeddings generate models for $\mathcal{EL}^{++}$theories and do not rely on precomputing deductive closures of graphs, and account for the semantics of conjunction, existential quantifiers, and the bottom concept (and therefore basic disjointness between classes).

While EL embeddings do not require negatives, we implement a form of negative sampling by randomly changing classes in axioms, similarly to how TransE and other knowledge graph embeddings generate negatives [Wang *et al.*, 2017]. In future work, we intend to explore more of the negatives that arise from the $\mathcal{EL}^{++}$theories directly. For example, we can encode the unique names assumption by asserting $\{a\} \sqcap \{b\} \sqsubseteq \bot$ for all instances $a, b \in I$, and infer further negatives by exploring disjointness.

Another limitation of our method is the use of TransE as relational model which does not allow us to capture role inclusion axioms or model relations that are not one-to-one relations. Most of the EL Embedding loss functions require no or little changes when using a different relation model; however, as a consequence of using TransE as relation model, for example the loss function for $\exists R.C \sqsubseteq \bot$ is degenerate and will need to be modified. Extending the relation model is not the only extension possible to our model; in the future, we also intend to explore improvements towards covering more expressive logics than $\mathcal{EL}^{++}$.

References

[Albrecht and Schlicker, 2007] Mario Albrecht and Andreas Schlicker. FunSimMat: a comprehensive functional similarity database. *Nucleic Acids Research*, 36(suppl1):D434–D439, 10 2007.

[Alshahrani *et al.*, 2017] Mona Alshahrani, et al. Neurosymbolic representation learning on biological knowledge graphs. *Bioinformatics*, 33(17):2723–2730, 2017.

[Ashburner *et al.*, 2000] Michael Ashburner, et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.

[Baader *et al.*, 2005] Franz Baader, et al. Pushing the el envelope. LTCS-Report LTCS-05-01, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2005.

[Baader, 2003] Franz Baader. *The Description Logic Handbook : Theory, Implementation and Applications*. Cambridge University Press, January 2003.

[Barwise and Etchemendy, 2002] Jon Barwise and John Etchemendy. *Language, Proof and Logic*. Center for the Study of Language and Inf, April 2002.

[Bordes *et al.*, 2013] Antoine Bordes, et al. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795. 2013.

[Golbreich and Horrocks, 2007] Christine Golbreich and Ian Horrocks. The OBO to OWL mapping, GO to OWL 1.1! In Christine Golbreich, et al., editors, *Proceedings of OWL: Experiences and Directions 2007 (OWLED-2007)*. CEUR-WS.org, 2007.

[Grau *et al.*, 2008] Bernardo Grau, et al. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, November 2008.

[Harispe *et al.*, 2015] Sebastien Harispe, et al. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

[Hoehndorf *et al.*, 2011] Robert Hoehndorf, et al. A common layer of interoperability for biomedical ontologies based on OWL EL. *Bioinformatics*, 27(7):1001–1008, April 2011.

[Jabeen *et al.*, 2019] Hajira Jabeen, et al. BioKEEN: A library for learning and evaluating biological knowledge graph embeddings. *Bioinformatics*, 02 2019.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kulmanov and Hoehndorf, 2017] Maxat Kulmanov and Robert Hoehndorf. Evaluating the effect of annotation size on measures of semantic similarity. *Journal of Biomedical Semantics*, 8(1):7, 2017.

[Mendez, 2012] Julian Mendez. jcel: A modular rule-based reasoner. In *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012*, 2012.

[Mikolov *et al.*, 2013] Tomas Mikolov, et al. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119, 2013.

[Motik *et al.*, 2009] Boris Motik, et al. Owl 2 web ontology language: Profiles. Recommendation, World Wide Web Consortium (W3C), 2009.

[Nickel *et al.*, 2016] Maximilian Nickel, et al. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, Jan 2016.

[Pesquita *et al.*, 2009] Catia Pesquita, et al. Semantic similarity in biomedical ontologies. *PLoS Comput Biol*, 5(7):e1000443, 07 2009.

[Resnik, 1995] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, pages 448–453, 1995.

[Roth *et al.*, 2016] Alexander Roth, et al. The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research*, 45(D1):D362–D368, 10 2016.

[Schulz *et al.*, 2009] Stefan Schulz, et al. Snomed reaching its adolescence: Ontologists' and logicians' health check. *International Journal of Medical Informatics*, 78:S86 – S94, 2009. MedInfo 2007.

[Smaili *et al.*, 2018] Fatima Zohra Smaili, et al. OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics*, 11 2018.

[Smith *et al.*, 2007] Barry Smith, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotech*, 25(11):1251–1255, 2007.

[Subramanian *et al.*, 2005] Aravind Subramanian, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.

[Wang *et al.*, 2017] Quan Wang, et al. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, Dec 2017.

[Zemojtel *et al.*, 2016] Tomasz Zemojtel, et al. The Human Phenotype Ontology in 2017. *Nucleic Acids Research*, 45(D1):D865–D876, 11 2016.