DELE: Deductive $\mathcal{EL}^{++}$ Embeddings for Knowledge Base Completion

NeSy 2024 Special Issue, paper # 51

Olga Mashkova [a], Fernando Zhapa-Camacho [a] and Robert Hoehndorf [a,b,c,*]

[a] *Computer Science Program, Computer, Electrical, and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*

[b] *Center of Excellence for Smart Health, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*

[c] *Center of Excellence for Generative AI, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia*

E-mail: robert.hoehndorf@kaust.edu.sa

Abstract.
Ontology embeddings map classes, relations, and individuals in ontologies into $\mathbb{R}^n$, and within $\mathbb{R}^n$ similarity between entities can be computed or new axioms inferred. For ontologies in the Description Logic $\mathcal{EL}^{++}$, several embedding methods have been developed that explicitly generate models of an ontology. However, these methods suffer from some limitations; they do not distinguish between statements that are unprovable and provably false, and therefore they may use entailed statements as negatives. Furthermore, they do not utilize the deductive closure of an ontology to identify statements that are inferred but not asserted. We evaluated a set of embedding methods for $\mathcal{EL}^{++}$ ontologies, incorporating several modifications that aim to make use of the ontology deductive closure. In particular, we designed novel negative losses that account both for the deductive closure and different types of negatives and formulated evaluation methods for knowledge base completion. We demonstrate that our embedding methods improve over the baseline ontology embedding in the task of knowledge base or ontology completion.

Keywords: Ontology Embedding, Knowledge Base Completion, Description Logic $\mathcal{EL}^{++}$

1. Introduction

Several methods have been developed to embed Description Logic theories or ontologies in vector spaces [10, 11, 21, 29, 40, 42, 44, 55]. These embedding methods preserve some aspects of the semantics in the vector space, and may enable the computation of semantic similarity, inferring axioms that are entailed, and predicting axioms that are not entailed but may be added to the theory. For the lightweight Description Logic $\mathcal{EL}^{++}$, several geometric embedding methods have been developed [21, 29, 40, 42, 55]. They can be proven to "faithfully" approximate a model in the sense that, if a certain optimization objective is reached (usually, a loss function reduced to 0), the embedding method has constructed a model of the $\mathcal{EL}^{++}$theory. Geometric model construction enables the execution of various tasks. These tasks include knowledge base completion and subsumption prediction via either

*Corresponding author. E-mail: robert.hoehndorf@kaust.edu.sa.

testing the truth of a statement under consideration in a single (approximate) model or aggregating truth values over multiple models.

Advances on different geometric embedding methods have usually focused on the expressiveness of the embedding methods; originally, hyperballs [29] where used to represent the interpretation of concept symbols, yet hyperballs are not closed under intersection. Therefore, axis-aligned boxes were introduced [21, 44, 55]. Furthermore, $\mathcal{EL}^{++}$allows for axioms pertaining to relations, and several methods have extended the way in which relations are modeled [21, 29, 55]. However, there are several aspects of geometric embeddings that have not yet been investigated. In particular, for $\mathcal{EL}^{++}$, there are sound and complete reasoners with efficient implementations that scale to very large knowledge bases [26]; it may therefore be possible to utilize a deductive reasoner together with the embedding process to improve generation of embeddings that represent geometric models.

We evaluate geometric embedding methods and incorporate deductive inference into the training process. We use the *ELEmbeddings* [29], *ELBE* [44], and *Box*^{2}*EL* [21] models for our experiments; however, our results also apply to other geometric embedding methods for $\mathcal{EL}^{++}$.

Our main contributions are as follows:

- We propose loss functions that incorporate negative samples in all normal forms and account for deductive closure during training.
- We introduce a fast approximate algorithm for computing the deductive closure of an $\mathcal{EL}^{++}$theory and use it to improve negative sampling during model training.
- We formulate evaluation methods for knowledge base completion that account for the deductive closure during evaluation.

This is an extended version of our previous work [35]. We now include a more comprehensive treatment of computing the deductive closure and using the deductive closure with $\mathcal{EL}^{++}$embedding methods. We make our code and data available at https://github.com/bio-ontology-research-group/DELE.

2. Preliminaries

2.1. Description Logic $\mathcal{EL}^{++}$

Let $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$ be a signature with set $\mathbf{C}$ of concept names, $\mathbf{R}$ of role names, and $\mathbf{I}$ of individual names. Given $A, B \in \mathbf{C}$, $r \in \mathbf{R}$, and $a, b \in \mathbf{I}$, $\mathcal{EL}^{++}$concept descriptions are constructed with the grammar $\bot \mid \top \mid A \sqcap B \mid \exists r.A \mid \{a\}$. ABox axioms are of the form $A(a)$ and $r(a,b)$, TBox axioms are of the form $A \sqsubseteq B$, and RBox axioms are of the form $r_1 \circ r_2 \circ \cdots \circ r_n \sqsubseteq r$. $\mathcal{EL}^{++}$*generalized concept inclusions* (GCIs) and *role inclusions* (RIs) can be normalized to follow one of these forms [3]: $C \sqsubseteq D$ (GCI0), $C \sqcap D \sqsubseteq E$ (GCI1), $C \sqsubseteq \exists R.D$ (GCI2), $\exists R.C \sqsubseteq D$ (GCI3), $C \sqsubseteq \bot$ (GCI0-BOT), $C \sqcap D \sqsubseteq \bot$ (GCI1-BOT), $\exists R.C \sqsubseteq \bot$ (GCI3-BOT) and $r \sqsubseteq s$ (RI0), $r_1 \circ r_2 \sqsubseteq s$ (RI1), respectively.

To define the semantics of an $\mathcal{EL}^{++}$ theory, we use [3] an *interpretation domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$. For every concept $A \in \mathbf{C}$, $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$; individual $a \in \mathbf{I}$, $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; role $r \in \mathbf{R}$, $r^{\mathcal{I}} \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Furthermore, the semantics for other $\mathcal{EL}^{++}$ constructs are the following (omitting concrete domains and role inclusions):

$$
\begin{aligned}
\bot^{\mathcal{I}} &= \emptyset \\
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}}, \\
(A \sqcap B)^{\mathcal{I}} &= A^{\mathcal{I}} \cap B^{\mathcal{I}}, \\
(\exists r.A)^{\mathcal{I}} &= \left\{ a \in \Delta^{\mathcal{I}} \mid \exists b : ((a,b) \in r^{\mathcal{I}} \land b \in A^{\mathcal{I}}) \right\}, \\
(a)^{\mathcal{I}} &= \{a\}
\end{aligned}
$$

An interpretation $\mathcal{I}$ is a model for an axiom $C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, for an axiom $B(a)$ if and only if $a^{\mathcal{I}} \in B^{\mathcal{I}}$; and for an axiom $r(a,b)$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ [4].

2.2. Knowledge Base Completion

The task of knowledge base completion is the addition (or prediction) of axioms to a knowledge base that are not explicitly represented. We call the task “ontology completion” when exclusively TBox axioms are predicted. The task of knowledge base completion may encompass both deductive [24, 48] and inductive [7, 15] inference processes and give rise to two subtly different tasks: adding only “novel” axioms to a knowledge base that are *not* in the deductive closure of the knowledge base, and adding axioms that are in the deductive closure as well as some “novel” axioms that are not deductively inferred; both tasks are related but differ in how they are evaluated.

Inductive inference, analogously to knowledge graph completion [12], predicts axioms based on patterns and regularities within the knowledge base. Knowledge base completion, or ontology completion, can be further distinguished based on the information that is used to predict “novel” axioms. We distinguish between two approaches to knowledge base completion: (1) knowledge base completion which relies solely on (formalized) information within the knowledge base to predict new axioms, and (2) knowledge base completion which incorporates side information, such as text, to enhance the prediction of new axioms. Here, we mainly consider the first case.

3. Related Work

3.1. Graph-Based Ontology Embeddings

Graph-based ontology embeddings rely on a construction (projection) of graphs from ontology axioms mapping ontology classes, individuals and roles to nodes and labeled edges [57]. Embeddings for nodes and edge labels are optimized following two strategies: by generating random walks and using a sequence learning method such as Word2Vec [39]or by using Knowledge Graph Embedding (KGE) methods [54]. These type of methods have been shown effective on knowledge base and ontology completion [10] and have been applied to domain-specific tasks such as protein–protein interaction prediction [10] or gene–disease association prediction [1, 11]. Graph-based methods rely on adjacency information of the ontology structure but cannot easily handle logical operators and do not approximate ontology models. Therefore, graph-based methods are not “faithful”, i.e., do not approximate models, do not allow determining whether statements are “true” in these models, and therefore cannot be used to perform semantic entailment.

3.2. Geometric-Based Ontology Embeddings

Multiple methods have been developed for the geometric construction of models for the $\mathcal{EL}^{++}$ language. ELEmbeddings [29] constructs an interpretation of concept names as sets of points lying within an open n-dimensional ball and generates an interpretation of role names as the set of pairs of points that are separated by a vector in $\mathbb{R}^n$, i.e., by the embedding of the role name. EmEL++ [40] extends ELEmbeddings with more expressive constructs such as role chains and role inclusions. ELBE [44] and BoxEL [55] use n-dimensional axis-aligned boxes to represent concepts, which has an advantage over balls because the intersection of two axis-aligned boxes is a box whereas the intersection of two n-balls is not an n-ball. BoxEL additionally preserves ABox facilitating a more accurate representation of knowledge base’s logical structure by ensuring, e.g., that an entity has the minimal volume. Box^{2}EL [21] represents ontology roles more expressively with two boxes encoding the semantics of the domain and codomain of roles. Box^{2}EL enables the expression of one-to-many relations as opposed to other methods. Axis-aligned cone-shaped geometric model introduced in [42] deals with $\mathcal{ALC}$ ontologies and allows for full negation of concepts and existential quantification by construction of convex sets in $\mathbb{R}^n$. This work has not yet been implemented or evaluated in an application.

3.3. Knowledge Base Completion Task

Several recent advancements in the knowledge base completion rely on side information as included in Large Language Models (LLMs). [23] explores how pretrained language models can be utilized for incorporating one ontology into another, with the main focus on inconsistency handling and ontology coherence. HalTon [9] addresses the task of event ontology completion via simultaneous event clustering, hierarchy expansion and type naming utilizing BERT [13] for instance encoding. [33] formulates knowledge base completion task as a Natural Language Inference (NLI) problem and examines how this approach may be combined with concept embeddings for identifying missing knowledge in ontologies. As for other approaches, [38] proposes a method that converts an ontology into a graph to recommend missing edges using structure-only link analysis methods, [51] constructs matrix-based ontology embeddings which capture the global and local information for subsumption prediction. All these methods use side information from LLMs and would not be applicable, for example, in the case where a knowledge base is private or consists of only identifiers; we do not consider methods based on pre-trained LLMs here as baselines.

3.4. Approximate Semantic Entailment

We follow [19] to state that when a model $\mathcal{M}$ of a theory $\mathcal{T}$ is also a model of an axiom $C \sqsubseteq D$ defined over $\mathcal{T}$, we call it *entailment* and denote it as $\mathcal{T} \models C \sqsubseteq D$. In this sense, semantic entailment can be understood as entailment over all the models of $\mathcal{T}$, which is expressed as $Mod(\mathcal{T}) \subseteq Mod(C \sqsubseteq D)$. Geometric-based ontology embedding methods construct geometric models for $\mathcal{EL}^{++}$theories. However, since the collection $Mod(\mathcal{T})$ is a class, it is not possible to construct all the possible geometric models. Therefore, we refer as *approximate semantic entailment* to the construction of a finite set of geometric models for a $\mathcal{EL}^{++}$theory.

In the context of bioinformatics, methods such as DeepGOZero [28] formulate the prediction of protein functions as an entailment problem, relying on ELEmbeddings to generate a model for the Gene Ontology. Subsequently, the extension to approximate semantic entailment is implemented in [30], where it is effectively showed that the generation of multiple models improves predictive performance of protein functions.

4. Methods

4.1. Datasets

4.1.1. Gene Ontology & STRING Data

Following previous works [21, 29, 44] we use common benchmarks for knowledge-base completion, in particular a task that predicts protein–protein interactions (PPIs) based on the functions of proteins. We also use the same data for the task of protein function prediction. For these tasks we use two datasets, each of them consists of the Gene Ontology (GO) [59] with all its axioms, protein–protein interactions (PPIs) and protein function axioms extracted from the STRING database [37]; each dataset focuses on only yeast proteins. GO is formalized using OWL 2 EL [17].

For the PPI yeast network we use the built-in dataset `PPIYeastDataset` available in the mOWL [58] Python library (release 0.2.1) where axioms of interest are split randomly into train, validation and test datasets in ratio 90:5:5 keeping pairs of symmetric PPI axioms within the same dataset, and other axioms are placed into the training part; validation and test sets are made up of TBox axioms of type $\{P_1\} \sqsubseteq \exists interacts\_with.\{P_2\}$ where P_1, P_2 are protein names. The GO version released on 2021-10-20 and the STRING database version 11.5 were used. Alongside with the yeast *interacts_with* dataset we collected the yeast $has\_function$ dataset organized in the same manner with validation and test parts containing TBox axioms of type $\{P\} \sqsubseteq \exists has\_function.\{GO\}$. Based on the information in the STRING database, in PPI yeast, the *interacts_with* relation is symmetric and the dataset is closed against symmetric interactions. We normalize each train ontology using the updated implementation of the jcel [36] reasoner [1] where we take into consideration newly generated concept and role names. Although role inclusion axioms may be utilized within the Box^2EL framework we ignore them since neither *ELEmbeddings* nor *ELBE* incorporate these types of axioms. Table in the appendix A shows the number of GCIs of each type in the datasets and the number of concepts and roles after normalization. For more precise evaluation of novel knowledge prediction we remove entailed axioms from the test set for function prediction task based on the precomputed deductive closure of the train ontology (see Section 5.2.1).

[1] https://github.com/julianmendez/jcel/pull/12

4.1.2. Food ontology

Food Ontology [14] contains structured information about foods formalized in $\mathcal{SRIQ}$ DL expressivity [10] involving terms from UBERON [41], NCBITaxon [16], Plant Ontology [22] etc. The data for subsumption prediction was extracted from the case studies used to evaluate OWL2Vec* [10][2]; the train part of the ontology was restricted to $\mathcal{EL}$ fragment and normalized using the jcel [36] reasoner. Since the normalization procedure splits each complex axiom into a set of shorter axioms including subsumptions between atomic concepts from the signature, it may result in adding axioms represented in the validation or test part of the ontology to the train part; to avoid this, we filtered out such axioms from the original validation and test datasets after the train ontology for subsumption prediction was normalized. Additionally, as described in Section 4.1.1, we remove entailed axioms from the test dataset. Statistics about the number of axioms of each GCI type, relations and classes can be found in Appendix B.

4.2. Evaluation Scores and Metrics

For GO & STRING data, we predict GCI2 axioms of type $\{P_1\} \sqsubseteq \exists interacts\_with.\{P_2\}$ or $\{P\} \sqsubseteq \exists has\_function.\{GO\}$ depending on the dataset. On Food Ontology, we predict GCI0 axioms of type $C \sqsubseteq D$, C and D are arbitrary classes from the signature. For each axiom type, we use the corresponding loss expressions to score axioms. This is justified by the fact that objective functions are measures of truth for each axiom within constructed models.

The predictive performance is measured by the Hits@n metrics for $n = 1, 10, 100$, macro and micro mean rank, and the area under the ROC curve (AUC ROC). For rank-based metrics, we calculate the score of $C \sqsubseteq \exists R.D$ or $C \sqsubseteq D$ for every class C from the test set and for every D from the set $\mathbf{C}$ of all classes (or subclasses of a certain type, such as proteins or functions for domain-specific cases) and determine the rank of a test axiom $C \sqsubseteq \exists R.D$. For macro mean rank and AUC ROC, we consider all axioms from the test set; for micro metrics, we compute corresponding class-specific metrics averaging them over all classes in the signature:

$$micro\_MR_{C \sqsubseteq \exists R.D} = Mean(MR_C(\{C \sqsubseteq \exists R.D,\ D \in \mathbf{C}\})) \tag{1}$$

$$micro\_MR_{C \sqsubseteq D} = Mean(MR_C(\{C \sqsubseteq D,\ D \in \mathbf{C}\})) \tag{2}$$

$$micro\_AUC\_ROC_{C \sqsubseteq \exists R.D} = Mean(AUC\_ROC_C(\{C \sqsubseteq \exists R.D,\ D \in \mathbf{C}\})) \tag{3}$$

$$micro\_AUC\_ROC_{C \sqsubseteq D} = Mean(AUC\_ROC_C(\{C \sqsubseteq D,\ D \in \mathbf{C}\})) \tag{4}$$

Additionally, we remove axioms represented in the train set or deductive closures (see Section 5.2.1) to obtain corresponding filtered metrics (FHits@n, FMR, FAUC).

4.3. Training Procedure

All models are optimized with respect to the sum of individual GCI losses (here we define the loss in most general case using all positive and all negative losses):

$$\begin{aligned}\mathcal{L} = {} & l_{C \sqsubseteq D} + l_{C \sqcap D \sqsubseteq E} + l_{C \sqsubseteq \exists R.D} + l_{\exists R.C \sqsubseteq D} + l_{C \sqsubseteq \bot} + l_{C \sqcap D \sqsubseteq \bot} + l_{\exists R.C \sqsubseteq \bot} + \\ & + l_{C \not\sqsubseteq D} + l_{C \sqcap D \not\sqsubseteq E} + l_{C \not\sqsubseteq \exists R.D} + l_{\exists R.C \not\sqsubseteq D} + l_{C \not\sqsubseteq \bot} + l_{C \sqcap D \not\sqsubseteq \bot} + l_{\exists R.C \not\sqsubseteq \bot}\end{aligned} \tag{5}$$

[2]https://github.com/KRR-Oxford/OWL2Vec-Star/tree/master/case_studies

All model architectures are built using mOWL [58] library on top of mOWL's base models. All models were trained using the same fixed random seed.

All models are trained for 2,000 epochs for STRING & GO datasets and 800 epochs for Food Ontology dataset with batch size of 32,768. Training and optimization is performed using Pytorch with Adam optimizer [27] and ReduceLROnPlateau scheduler with patience parameter 10. We apply early stopping if validation loss does not improve for 20 epochs. For *ELEmbeddings*, hyperparameters are tuned using grid search over the following set: margin $\gamma \in \{-0.1, -0.01, 0, 0.01, 0.1\}$, embedding dimension $\{50, 100, 200, 400\}$, learning rate $\{0.01, 0.001, 0.0001\}$; since none of our datasets contains unsatisfiable classes, we do not tune the parameter ε appearing in GCI0-BOT and GCI3-BOT negative losses. For *ELBE*, grid search is performed over 60 randomly chosen subsets of the following hyperparameters: embedding dimension $\{25, 50, 100, 200\}$, margin $\{-0.1, -0.01, 0, 0.01, 0.1\}$, $\varepsilon \in \{0.1, 0.01, 0.001\}$ (for experiments with all negative losses involved), learning rate $\{0.01, 0.001, 0.0001\}$. The same strategy is applied to Box^2EL models for embedding dimension $\{25, 50, 100, 200\}$, margin $\gamma \in \{-0.1, -0.01, 0, 0.01, 0.1\}$, $\delta \in \{1, 2, 4\}$, $\varepsilon \in \{0.1, 0.01, 0.001\}$ (similarly, for experiments with all negative losses involved), regularization factor $\lambda \in \{0, 0.05, 0.1, 0.2\}$, and learning rate $\{0.01, 0.001, 0.0001\}$. For experiments with negatives filtration during training we use the same set of hyperparameters for random and filtered mode of negative sampling. See Appendix C for details on optimal hyperparameters used.

5. Results

5.1. Negative sampling and objective functions

Ontology embedding methods select negatives by replacing one of the classes with a randomly chosen one; e.g., for axioms of type $C \sqsubseteq D$ represented within the ontology $C \sqsubseteq D'$ for some arbitrary or semantically valid concept D'. *ELEmbeddings*, *ELBE* and Box^2EL use a single loss for "negatives", i.e., axioms that are not included in the knowledge base; the loss is used only for axioms of the form $C \sqsubseteq \exists R.D$ (GCI2) which are randomly sampled; negatives are not sampled for other normal forms. Correspondingly, the embedding methods were primarily evaluated on predicting GCI2 axioms (Box^2EL was also evaluated on subsumption prediction); this evaluation procedure might have introduced biases towards axioms of type GCI2, and influenced the ability of geometric models to predict axioms of other types.

Consequently, we also sample negatives for other normal forms and add "negative" losses (i.e., losses for the sampled "negatives") for all other normal forms. We test the effect of the expanded negative sampling and negative losses first on a small ontology that can be embedded and visualized in 2D space, and then on a larger application.

5.1.1. ELEmbeddings Negative Losses

For *ELEmbeddings*, we construct the following "negative" losses:

$$loss_{C \not\sqsubseteq D}(c, d) = \max(0, r_\eta(c) + r_\eta(d) - \|f_\eta(c) - f_\eta(d))\| + \gamma) + |\|f_\eta(c)\| - 1| + |\|f_\eta(d)\| - 1| \tag{6}$$

$$\begin{aligned} loss_{C \sqcap D \not\sqsubseteq E}(c, d, e) = \max(0, -r_\eta(c) - r_\eta(d) + \|f_\eta(c) - f_\eta(d))\| - \gamma) + \\ + \max(0, r_\eta(c) - \|f_\eta(c) - f_\eta(e))\| + \gamma) + \max(0, r_\eta(d) - \|f_\eta(d) - f_\eta(e))\| + \gamma) + \\ + |\|f_\eta(c)\| - 1| + |\|f_\eta(d)\| - 1| + |\|f_\eta(e)\| - 1| \end{aligned} \tag{7}$$

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

$$loss_{C \not\sqsubseteq \perp}(c) = \max(0, \varepsilon - r_\eta(c)) \tag{9}$$

$$loss_{C \sqcap D \not\sqsubseteq \perp}(c, d) = \max(0, -r_\eta(c) - r_\eta(d) + \|f_\eta(c) - f_\eta(d))\| - \gamma) + |\|f_\eta(c)\| - 1| + |\|f_\eta(d)\| - 1| \tag{10}$$

$$loss_{\exists R.C \not\sqsubseteq \perp}(r, c) = \max(0, \varepsilon - r_\eta(c)) \tag{11}$$

Here, γ stands for a margin parameter, and ε is a small positive number. We employ notations from the *ELEmbeddings* method where $r_\eta(c)$, $r_\eta(d)$, $r_\eta(e)$ and $f_\eta(c)$, $f_\eta(d)$, $f_\eta(e)$ denote the radius and the ball center associated with classes c, d, e, respectively; and $f_\eta(r)$ denotes the embedding vector associated with relation r. There is a geometrical part as well as a regularization part for each new negative loss forcing class centers to lie on a unit ℓ_2-sphere.

As reflected in Eq. 6, we use the original GCI1-BOT loss for disjoint classes; although non-containment of ball corresponding to C within the ball corresponding to D is not equivalent to their disjointness, the loss aims to minimize the classes' overlap for better optimization. The same logic applies for the negative loss in Eq. 8 where we minimize overlap between the translated ball corresponding to class C and the ball representing D.

Negative loss 7 is constructed similarly to the $C \sqcap D \sqsubseteq E$ loss: the first part penalizes non-overlap of the classes C and D (we do not consider the disjointness case since, for every class X, we have $\perp \sqsubseteq X$); furthermore, for negative sampling of axioms of this type, we vary only the E part of GCI1 axioms from the ontology, so the intersection of C and D is non-empty by assumption. The second and the third part force the center corresponding to E not to lie in the intersection of balls associated with C and D. Here we do not consider constraints on radius of the ball for E class and focus only on relative positions of C, D and E class centers and overlapping of n-balls representing C and D. Since the first part of the loss encourages classes to have a non-empty intersection, we use it as a negative loss for GCI1-BOT axioms (see Eq. 10).

In the original method losses for axioms of type GCI0-BOT and GCI3-BOT force radii of unsatisfiable classes to become 0. For the correspondent negative losses (see Eq. 9 and Eq. 11) we use the interpretation for satisfiable classes as balls with non-zero radius, i.e., with radius which equals to or greater than some small positive number ε.

5.1.2. ELBE Negative Losses

ELBE is a model that relies on boxes instead of balls. The negative losses for *ELBE* have the following form:

$$loss_{C \not\sqsubseteq D}(c, d) = \|\max(zeros, -|e_c(c) - e_c(d)| + e_o(c) + e_o(d) + margin)\| \tag{12}$$

$$loss_{C \sqcap D \not\sqsubseteq E}(c, d, e) = \|\max(zeros, -|e_c(new) - e_c(e)| + e_o(new) + e_o(e) + margin)\| \tag{13}$$

$$loss_{\exists R.C \not\sqsubseteq D}(r, c, d) = \|max(zeros, -|e_c(c) - e_c(r) - e_c(d))| + e_o(c) + e_o(d) + margin)\| \tag{14}$$

$$loss_{C \not\sqsubseteq \perp}(c) = \max(0, \varepsilon - \|e_o(c)\|) \tag{15}$$

$$loss_{C \sqcap D \not\sqsubseteq \perp}(c, d) = \max(0, \varepsilon - \|e_o(new)\|) \tag{16}$$

$$loss_{\exists R.C \not\sqsubseteq \perp}(r, c) = \max(0, \varepsilon - \|e_o(c)\|) \tag{17}$$

Here, similarly, ε is a small positive number, $e_c(c)$, $e_c(d)$ and $e_o(c)$, $e_o(d)$ denote the box center and the box offset associated with classes c, d, respectively, $e_c(r)$ denotes the embedding vector associated with relation r, and $e_c(new)$, $e_o(new)$ correspond to the center and the offset of the box which is the result of intersection of boxes associated with concepts c and d.

Following the same method of negative loss construction for *ELEmbeddings*, we use GCI1-BOT loss as a negative loss for $C \sqsubseteq D$ axioms (see Eq. 12). Since axis-aligned hyperrectangles are closed under intersection, we also use GCI1-BOT for the intersection of boxes representing C and D concepts and the E box. This property also allows us to interpret each negative sample for $C \sqcap D \sqsubseteq \perp$ axioms as a box intersection with nonzero offset (see Eq. 16).

5.1.3. Box^2EL Negative Losses

Box^2EL is also based on boxes but uses a different relation model compared to ELBE. The corresponding negative losses are designed as follows:

$$loss_{C \not\sqsubseteq D}(c, d) = \| \max(\mathbf{0}, -(\boldsymbol{d}(Box(C), Box(D)) + \gamma))\| \tag{18}$$

$$loss_{C \sqcap D \not\sqsubseteq E}(c, d, e) = \| \max(\mathbf{0}, -(\boldsymbol{d}(Box(C) \cap Box(D), Box(E)) + \gamma))\| \tag{19}$$

$$loss_{\exists R.C \not\sqsubseteq D}(r, c, d) = (\delta - \mu(Head(r) - Bump(C), Box(D)))^2 \tag{20}$$

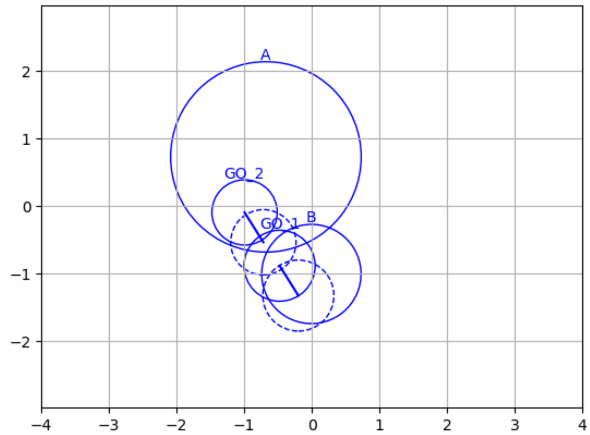
$$loss_{C \not\sqsubseteq \perp}(c) = \max(0, \varepsilon - \|o(C)\|) \tag{21}$$

$$loss_{C \sqcap D \not\sqsubseteq \perp}(c, d) = \max(0, \varepsilon - \|o(Box(C) \cap Box(D))\|) \tag{22}$$

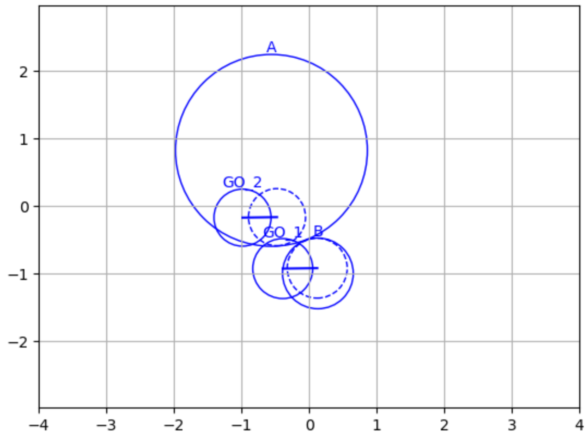
$$loss_{\exists R.C \not\sqsubseteq \perp}(r, c) = \max(0, \varepsilon - \|o(C)\|) \tag{23}$$

Additionally making use of the notations from Box^2EL [21], ε is a small positive number, $Box(C)$, $Box(D)$, $Box(E)$ are boxes associated with classes c, d, e, respectively, γ denotes a margin parameter, δ is a parameter from the GCI2 negative loss, $Head(r)$ represents the head box of relation r interpretation, and $Bump(C)$ corresponds to a bump vector associated with concept C.

Equations 18 and 19 are constructed in a similar fashion as for *ELBE* based on the GCI1-BOT loss which penalizes the element-wise distance $\boldsymbol{d}$ between axis-aligned boxes; negative losses 21–23 encourage boxes to be non-empty. The GCI3 negative loss reflects the structure of the original GCI3 loss, and the negative loss for GCI2 axioms forces the minimal distance μ between the "bumped" box representing class C and box D to be at least δ.



(a) GCI2 negative loss



(b) All negative losses

Fig. 1. *ELEmbeddings* example. Dashed circles represent translated classes by relational vector corresponding to *has_function* relation.

5.1.4. Experiments

We evaluate whether adding negative losses for all normal forms will allow for the construction of a better model and improve the performance in the task of knowledge base completion. We formulate and add negative losses for all normal forms given by equations 6–23.

First, we investigate a simple example corresponding to the task of protein function prediction using the *ELEmbeddings* model. Let us consider an ontology consisting of two axioms stating that there are two disjoint functions $\{GO_1\}$ and $\{GO_2\}$, and proteins having these functions are also disjoint: $\{GO_1\} \sqcap \{GO_2\} \sqsubseteq \bot$, $\exists has\_function.\{GO_1\} \sqcap has\_function.\{GO_2\} \sqsubseteq \bot$. After normalization, the last axiom is substituted by the following three axioms: $A \sqcap B \sqsubseteq \bot$, $\exists has\_function.\{GO_1\} \sqsubseteq B$, $\exists has\_function.\{GO_2\} \sqsubseteq A$ where A, B are new concept names. To visualize the results, we embed these axioms in 2D space. Figure 1(a) shows the embedding generated with the original *ELEmbeddings* model. Since there are no axioms of type GCI2 represented within the knowledge base, the model learns without any negative examples and demonstrates poor performance compared to the model with incorporated negative losses for all normal forms as demonstrated in Figure 1(b).

Since we are interested in predicting not only axioms of type $C \sqsubseteq \exists R.D$ for which negative sampling is used in the original *ELEmbeddings*, *ELBE* and *Box*^{2}*EL*, we also examine the effect of all negative losses utilization during training on Food Ontology for subsumption prediction (see Table 3). We find that the *ELEmbeddings* model does not improve on the Food Ontology subsumption prediction task, but *ELBE* with additional losses improves over the original model; *Box*^{2}*EL* with additional losses surpasses its version with just GCI2 negative loss in Hits@n metrics.

Additionally, we evaluate the performance on a standard benchmark set for protein–protein interaction (PPI) prediction (see Table 2). For this task, the test axioms are of the type GCI2. We observe that *ELEmbeddings* and *ELBE* with negative losses for all normal forms integrated demonstrate superior performance compared to their initial configurations in terms of Hits@n metrics; it also allows *Box*^{2}*EL* to lower ranks of test axioms. Generally, for the task of PPI prediction, additional negative sampling improves performance.

5.2. Negative sampling

In the case of knowledge base completion where the deductive closure contains potentially many non-trivial entailed axioms, the random sampling approach for negatives may lead to suboptimal learning since some of the axioms treated as negatives may be entailed (and should therefore be true in any model, in particular the one constructed by the geometric embedding method). As an example, let us consider the simple ontology consisting of two axioms: $A \sqcap B \sqsubseteq C$ and $D \sqsubseteq B$. For the $A \sqcap B \sqsubseteq C$ axiom, random negative sampling will sample $A \sqcap B \sqsubseteq C'$ where C' is one of A, B, C, D. Since the knowledge base makes the axioms $A \sqcap B \sqsubseteq A$, $A \sqcap B \sqsubseteq B$, and $A \sqcap B \sqsubseteq C$ true, in 75% of cases we will sample a negative for this axiom that is actually true in each model.

We suggest to filter selected negatives during training based on the deductive closure of the knowledge base: for each randomly generated axiom to be used as negative, we check whether it is present in the deductive closure and, if it is, we delete it.

ALGORITHM 1

An algorithm for computation of axioms in the deductive closure using inference rules; axioms in bold correspond to subclass/superclass axioms derived using ELK reasoner (here we use the transitive closure of the ELK inferences); plain axioms come from the knowledge base.

---

for all $C \sqcap D \sqsubseteq E$ in the knowledge base **do**

$$\frac{C \sqcap D \sqsubseteq E \quad \boldsymbol{C' \sqsubseteq C} \quad \boldsymbol{D' \sqsubseteq D} \quad \boldsymbol{E \sqsubseteq E'}}{C' \sqcap D' \sqsubseteq E'}$$

end for

for all $C \sqsubseteq \exists R.D$ in the knowledge base **do**

$$\frac{C \sqsubseteq \exists R.D \quad \boldsymbol{C' \sqsubseteq C} \quad \boldsymbol{D \sqsubseteq D'} \quad R \sqsubseteq R'}{C' \sqsubseteq \exists R'.D'} \qquad \frac{C \sqsubseteq \exists R.D \quad D \sqsubseteq \exists R'.E \quad R \circ R' \sqsubseteq S}{C \sqsubseteq \exists S.E}$$

end for

for all $\exists R.C \sqsubseteq D$ in the knowledge base **do**

$$\frac{\exists R.C \sqsubseteq D \quad \boldsymbol{C' \sqsubseteq C} \quad \boldsymbol{D \sqsubseteq D'} \quad R' \sqsubseteq R}{\exists R'.C' \sqsubseteq D'}$$

end for

for all $C \sqcap D \sqsubseteq \bot$ in the knowledge base **do**

$$\frac{C \sqcap D \sqsubseteq \bot \quad \boldsymbol{C' \sqsubseteq C} \quad \boldsymbol{D' \sqsubseteq D}}{C' \sqcap D' \sqsubseteq \bot} \qquad \frac{C \sqcap D \sqsubseteq \bot}{C \sqcap D \sqsubseteq E}$$

end for

for all $\exists R.C \sqsubseteq \bot$ in the knowledge base **do**

$$\frac{\exists R.C \sqsubseteq \bot \quad \boldsymbol{C' \sqsubseteq C} \quad R' \sqsubseteq R}{\exists R'.C' \sqsubseteq \bot}$$

end for

5.2.1. Deductive Closure

The *deductive closure* of a theory T refers to the smallest set containing all statements which can be inferred by deductive reasoning over T; for a given deductive relation $\vdash$, we call $T^{\vdash} = \{\phi \mid T \vdash \phi\}$ the deductive closure of T. In knowledge bases, the deductive closure is usually not identical to the asserted axioms in the knowledge base; it is also usually infinite. Representing the deductive closure is challenging since it is infinite, but, in $\mathcal{EL}^{++}$, any knowledge base can be normalized to one of the seven normal forms; therefore, we can compute the deductive closure with respect to these normal forms, and this set will be finite (as long as the concept and role names are finite). However, $\mathcal{EL}^{++}$reasoners such as ELK [26] compute subsumption hierarchies, i.e., all axioms of the form

$C \sqsubseteq D$ in the deductive closure, but not entailed axioms for the other normal forms. We use the inferences computed by ELK (of the form $C \sqsubseteq D$ where C and D are concept names) to design an algorithm that computes (a part of) the deductive closure with respect to the $\mathcal{EL}^{++}$normal forms; the algorithm implements a sound but possibly incomplete set of inference rules. Algorithm 1 contains inference rules for deriving entailed axioms of type GCI1, GCI2, GCI3, GCI1-BOT and GCI3-BOT from axioms explicitly represented within a knowledge base; GCI0 and GCI0-BOT axioms are precomputed by ELK. Algorithm 2 provides a set of additional rules depending on arbitrary classes and relations represented within a knowledge base after inferred axioms from Algorithm 1 are computed. Although we can use ELK or similar reasoners to query for arbitrary entailed axioms, the algorithms we propose have an advantage over this method since it does not require the addition of a new concept to an ontology and recomputing the concept hierarchy.

ALGORITHM 2
Additional entailed axioms

for all concepts C, D, E, E' in the signature **do**

$$\frac{}{C \sqcap \bot \sqsubseteq E} \quad \frac{D \sqsubseteq \bot}{C \sqcap D \sqsubseteq E} \quad \frac{E \sqsubseteq E'}{C \sqcap E \sqsubseteq E'} \quad \frac{C \sqcap D \sqsubseteq \bot}{C \sqcap D \sqsubseteq E}$$

$$\frac{C \sqsubseteq E \quad D \sqsubseteq E \quad C' \sqsubseteq C \quad D' \sqsubseteq D \quad E \sqsubseteq E'}{C' \sqcap D' \sqsubseteq E'} \quad \frac{C \sqsubseteq C'}{C \sqcap \top \sqsubseteq C'}$$

end for
for all relations R and all concepts $D \neq \bot$ in the signature **do**

$$\frac{}{\bot \sqsubseteq \exists R.D} \quad \frac{C \sqsubseteq \bot}{C \sqsubseteq \exists R.D}$$

end for
for all relations R and all concepts $C \neq \bot$ in the signature **do**

$$\frac{}{\exists R.C \sqsubseteq \top}$$

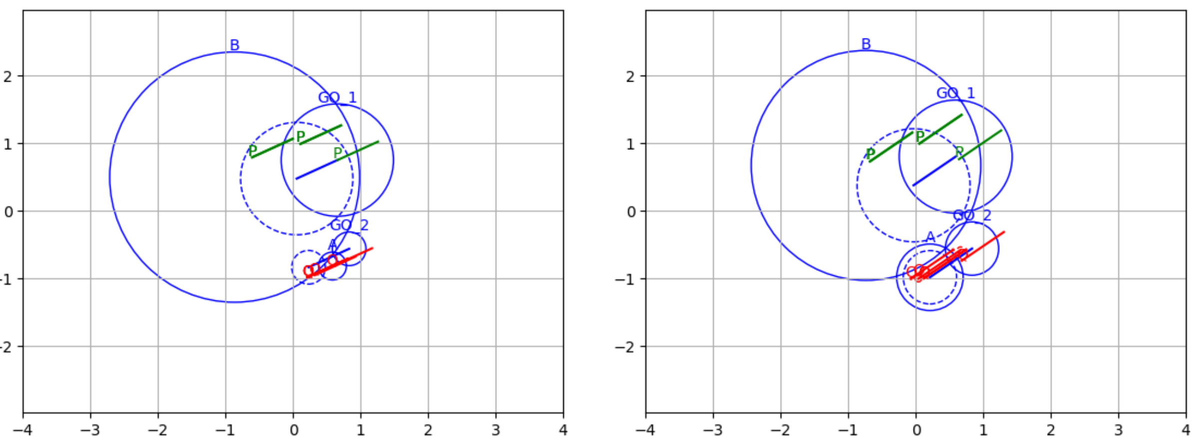
end for

We show a detailed example of the algorithm works in Appendix D based on the simple ontology example introduced in Section 5.1.4.

5.2.2. *Experiments*

Using the example introduced in Section 5.1.4 and the *ELEmbeddings* embedding model, we demonstrate that negatives filtration may be beneficial for constructing a model of a theory. Apart from axioms mentioned earlier, i.e., $\{GO_1\} \sqcap \{GO_2\} \sqsubseteq \bot$, $A \sqcap B \sqsubseteq \bot$, $\exists has\_function.\{GO_1\} \sqsubseteq B$ and $\exists has\_function.\{GO_2\} \sqsubseteq A$, we add 10 more axioms about 5 proteins $\{P_1\}, \dots, \{P_5\}$ having function $\{GO_1\}$ (i.e., $\{P_i\} \sqsubseteq \exists has\_function.\{GO_1\},\ i = 1, \dots, 5$), and 5 proteins $\{Q_1\}, \dots, \{Q_5\}$ having function $\{GO_2\}$ (i.e., $\{Q_i\} \sqsubseteq \exists has\_function.\{GO_2\},\ i = 1, \dots, 5$). Figure 2 shows the constructed models with and without negatives filtering. We observe that the model with filtered negatives provides faithful representation of GCI3 axiom $\exists has\_function.\{GO_2\} \sqsubseteq A$ and axioms introducing proteins having function $\{GO_2\}$ as opposed to its counterpart with random negatives.

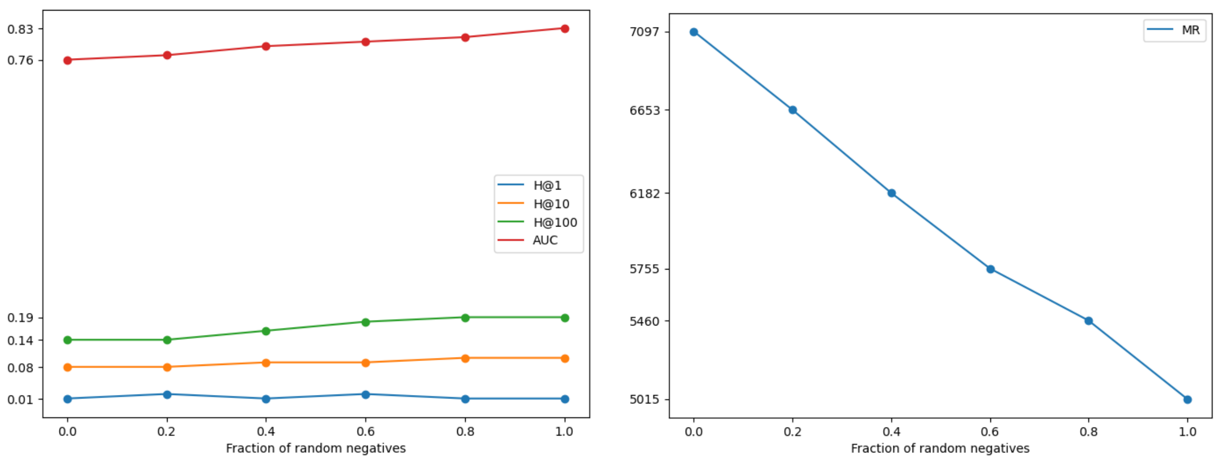
Tables 2–3 show results in the tasks of protein–protein interaction and subsumption prediction. We find that excluding axioms in the deductive closure for negative selection slightly improves or yields similar results. One



(a) With random negatives (b) With filtered negatives

Fig. 2. *ELEmbeddings* example. Dashed circles represent translated classes by relational vector corresponding to *has_function* relation. 'Red' classes represent proteins $\{Q_1\}, \ldots, \{Q_5\}$, 'green' classes represent proteins $\{P_1\}, \ldots, \{P_5\}$.

possible reason is that a randomly chosen axiom is very unlikely to be entailed since very few axioms are entailed compared to all possible axioms to choose from.



(a) (b)

Fig. 3.

Because the chance of selecting an entailed axiom as a negative depends on the knowledge base on which the embedding method is applied, we perform additional experiments on Food Ontology with *ELEmbeddings* model where we bias the selection of negatives; we chose between 100% negatives to 0% negatives from the entailed axioms. We find that reducing the number of entailed axioms from the negatives has an effect to improve performance and the effect increases the more axioms would be chosen from the entailed ones (see Figure 3).

5.3. Evaluation Strategies

In the task of knowledge base completion with many non-trivial entailed axioms, the deductive closure can also be used to modify the evaluation metrics, or define novel evaluation metrics that distinguish between entailed and non-entailed axioms. So far, ontology embedding methods that have been applied to the task of knowledge base completion have used evaluation measures that are taken from the task of knowledge graph completion; in particular, they only evaluate knowledge base completion using axioms that are "novel" and not entailed. However, any entailed axiom will be true in all models of the knowledge base, and therefore also in the geometric model that is constructed by the embedding method.

We suggest to filter entailed axioms from training or test sets when the aim is to predict "novel" (i.e., non-entailed) knowledge. The geometric embedding methods generate models making all entailed axioms true in all models. It is expected that methods explicitly constructing models preferentially make entailed axioms true and rank them higher than non-entailed axioms. If the evaluation is based solely on non-entailed axioms, it will consider all similar inferred axioms false, and to avoid this, we may filter such axioms from the ranking list. The more axioms are filtered, the more entailed axioms are predicted by a model.

We compute filtered metrics for the protein function and subsumption prediction tasks. Both of them account for entailed axioms prediction since if, e.g., $C \sqsubseteq D$ is being predicted then first models may predict axioms of type $C \sqsubseteq D'$ where D' is any superclass of D; the same is true for function prediction axioms $\{P\} \sqsubseteq \exists has\_function.\{GO\}$ and all superclasses $\{GO'\}$ of $\{GO\}$ class. Note that the protein–protein interaction prediction task is not tailored for evaluation using deductive closures of the train or test set: for each protein $\{P\}$ its subclasses include only $\bot$ and superclasses include only $\top$. As a result, the only inferred axioms will be of type $\bot \sqsubseteq \exists interacts\_with.\{P\}$, $\{P_1\} \sqsubseteq \exists interacts\_with.\{P_2\}$ or $\{P\} \sqsubseteq \exists interacts\_with.\top$, and filtered metrics may be computed only with respect to the train part of the ontology.

For function prediction and subsuption prediction, we employ filtration of metrics based on the deductive closure of the train set and of the test set. Tables 3 and 1 contain results for subsumption prediction on Food Ontology and function prediction on GO, respectively.

Our findings suggest that the baseline *ELEmbeddings* predicts primarily entailed axioms of GCI2 type, yet for GCI0 the model predicts "novel" knowledge first whereas the model modifications with additional negative losses and negatives filtration derive entailed knowledge in the first place. Losses for all normal forms and negatives filtering during training aid *ELBE* and *Box*^{2}*EL* to construct model-generated embeddings which first predict logically inferred knowledge and then non-entailed axioms of type GCI2 or GCI0, respectively. The results indicate that models with all types of valid negatives in most cases explicitly construct models.

6. Discussion

We evaluated properties of *ELEmbeddings*, *ELBE* and *Box*^{2}*EL*, ontology embedding methods that aims to generate a model of an $\mathcal{EL}^{++}$theory; the properties we evaluate hold similarly for other ontology embedding methods that construct models of $\mathcal{EL}^{++}$theories. While we demonstrate several improvements over the original model, we can also draw some general conclusions about ontology embedding methods and their evaluation. Knowledge base completion is the task of predicting axioms that should be added to a knowledge base; this task is adapted from knowledge graph completion where triples are added to a knowledge graph. The way both tasks are evaluated is by removing some statements (axioms or triples) from the knowledge base, and evaluating whether these axioms or triples can be recovered by the embedding method. This evaluation approach is adequate for knowledge graphs which do not give rise to many entailments. However, knowledge bases give rise to potentially many non-trivial entailments that need to be considered in the evaluation. In particular, embedding methods that aim to generate a model of a knowledge base will first generate entailed axioms (because entailed axioms are true in all models); these methods perform knowledge base completion as a generalization of generating the model where either other statements may be true, or they may be approximately true in the generated structure. This has two consequences: the evaluation procedure needs to account for this; and the model needs to be sufficiently rich to allow useful predictions.

We have introduced a method to compute the deductive closure of $\mathcal{EL}^{++}$knowledge bases; this method relies on an automated reasoner and is sound. We use all the axioms in the deductive closure as positive axioms to be predicted when evaluating knowledge base completion, to account for methods that treat knowledge base completion as a generalization of constructing a model and testing for truth in this model. We find that some models (e.g., modified box-based models using valid negatives of all types) can predict entailed axioms well, some (e.g., the original *Box*^{2}*EL* model) preferentially predict "novel", non-entailed axioms; these methods solve subtly different problems (either generalizing construction of a model, or specifically predicting novel non-entailed axioms). We also modify the evaluation procedure to account for the inclusion of entailed axioms as positives; however, the evaluation measures are still based on ranking individual axioms and do not account for semantic similarity. For example, if during testing, the correct axiom to predict is $C \sqsubseteq \exists R.D$ but the predicted axiom is $C \sqsubseteq \exists R.E$, the prediction may be considered to be "more correct" if $D \sqsubseteq E$ was in the knowledge base than if $D \sqcap E \sqsubseteq \bot$ was in the knowledge base. Novel evaluation metrics need to be designed to account for this phenomenon, similarly to ontology-based evaluation measures used in life sciences [46]. It is also important to expand the set of benchmark sets for knowledge base completion.

Use of the deductive closure is not only useful in evaluation but also when selecting negatives. In formal knowledge bases, there are at least two ways in which negatives for axioms can be chosen: they are either non-entailed axioms, or they are axioms whose negation is entailed. However, in no case should entailed axioms be considered as negatives; we demonstrate that filtering entailed axioms from selected negatives during training improves the performance of the embedding method consistently in knowledge base completion (and, obviously, more so when entailed axioms are considered as positives during evaluation).

While we only report our experiments with *ELEmbeddings*, *ELBE* and *Box*^{2}*EL*, our findings, in particular about the evaluation and use of deductive closure, are applicable to other geometric ontology embedding methods. As ontology embedding methods are increasingly applied in knowledge-enhanced learning and other tasks that utilize some form of approximate computation of entailments, our results can also serve to improve the applications of ontology embeddings.

Table 1

Protein function prediction experiments on yeast proteins. 'l' corresponds to all negative losses, 'l+n' means a model was trained using all negative losses and negatives filtering. For each model we report non-filtered metrics (NF) and filtered metrics with respect to the deductive closure of the train and the test set combined together (F). Values in **bold** indicate best metrics; <u>underlined</u> values highlight best filtered metrics.

Model	H@1		H@10		H@100		macro_MR		micro_MR		macro_AUC		micro_AUC	
	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F
ELEm	0.00	0.00	**0.01**	0.01	0.03	0.03	21198	<u>21150</u>	21165	<u>21118</u>	0.62	0.62	0.63	0.63
ELEm+l	0.00	0.00	0.00	0.00	0.03	0.03	9603	9575	9449	9423	**0.83**	0.83	**0.84**	0.84
ELEm+l+n	0.00	0.00	0.00	0.00	0.03	0.03	**9488**	9460	**9334**	9307	**0.83**	0.83	**0.84**	0.84
ELBE	0.00	0.00	**0.03**	0.03	**0.24**	0.24	**4229**	4209	**4156**	4137	**0.92**	0.92	**0.93**	0.93
ELBE+l	0.00	0.00	0.00	0.00	0.01	0.01	12920	<u>12865</u>	12797	12745	0.77	0.77	0.78	0.78
ELBE+l+n	0.00	0.00	0.00	0.00	0.01	0.01	12900	<u>12845</u>	12772	<u>12719</u>	0.77	0.77	0.78	0.78
Box^2*EL*	**0.05**	0.09	**0.28**	0.31	**0.55**	0.55	**1988**	1979	**1988**	1980	0.96	0.96	**0.97**	0.97
Box^2*EL*+l	0.04	0.06	0.24	0.27	0.54	0.55	2129	2120	2099	2091	0.96	0.96	**0.97**	0.97
Box^2*EL*+l+n	**0.05**	0.06	0.24	0.27	0.54	0.55	2161	2152	2147	2139	0.96	0.96	0.96	0.96

Table 2

Protein–protein interaction prediction experiments on yeast proteins. 'l' corresponds to all negative losses, 'l+n' means a model was trained using all negative losses and negatives filtering. Non-filtered metrics are reported. Values in **bold** indicate best non-filtered metrics.

Model	H@1	H@10	H@100	macro_MR	micro_MR	macro_AUC	micro_AUC
ELEm	0.00	0.05	0.31	599.21	701.57	0.90	0.90
ELEm+l	0.00	**0.06**	0.35	532.93	681.02	**0.91**	0.90
ELEm+l+n	0.00	**0.06**	**0.37**	**519.62**	671.19	**0.91**	**0.91**
ELBE	0.00	0.07	0.37	**829.86**	**1123.47**	**0.91**	**0.89**
ELBE+l	0.00	**0.08**	**0.40**	984.92	1259.54	0.84	0.82
ELBE+l+n	0.00	**0.08**	**0.40**	984.18	1281.20	0.84	0.82
Box^2EL	0.00	0.05	0.57	215.07	287.16	0.96	0.96
Box^2EL+l	0.00	0.05	0.57	200.85	**250.17**	**0.97**	0.96
Box^2EL+l+n	0.00	0.05	**0.58**	**197.73**	250.47	**0.97**	0.96

Table 3

Subsumption prediction experiments on Food Ontology. 'l' corresponds to all negative losses, 'l+n' means a model was trained using all negative losses and negatives filtering. For each model we report non-filtered metrics (NF) and filtered metrics with respect to the deductive closure of the train and the test set combined together (F). Values in **bold** indicate best metrics; <u>underlined</u> values highlight best filtered metrics.

Model	H@1		H@10		H@100		macro_MR		micro_MR		macro_AUC		micro_AUC	
	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F	NF	F
ELEm	0.01	0.02	**0.12**	0.12	**0.21**	0.21	**4659**	<u>4656</u>	**4662**	4659	**0.84**	0.84	**0.84**	0.84
ELEm+l	0.01	0.02	0.10	<u>0.11</u>	0.19	0.19	5015	5013	5020	5017	0.83	0.83	0.83	0.83
ELEm+l+n	0.01	0.02	0.10	<u>0.11</u>	0.19	0.19	5022	<u>5019</u>	5027	5024	0.83	0.83	0.83	0.83
ELBE	0.00	0.00	0.01	0.01	0.09	0.09	6695	<u>6692</u>	6688	6686	0.77	0.77	0.77	0.77
ELBE+l	0.00	0.00	**0.04**	0.04	**0.14**	0.14	5428	5426	5412	<u>5409</u>	**0.81**	0.81	**0.82**	0.82
ELBE+l+n	0.00	0.00	**0.04**	0.04	**0.14**	0.14	**5427**	<u>5424</u>	**5410**	5408	**0.81**	0.81	**0.82**	0.82
Box^2EL	0.00	0.00	0.01	0.01	0.10	0.10	**3900**	3898	**3877**	<u>3874</u>	**0.87**	0.87	**0.87**	0.87
Box^2EL+l	0.00	0.00	0.04	0.04	0.13	0.13	7550	<u>7547</u>	7555	7553	0.74	0.74	0.74	0.74
Box^2EL+l+n	0.00	0.00	**0.05**	0.05	**0.14**	0.14	6865	<u>6862</u>	6869	<u>6866</u>	0.76	0.76	0.77	0.77

Acknowledgements

This work has been supported by funding from King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. URF/1/4355-01-01, URF/1/4675-01-01, URF/1/4697-01-01, URF/1/5041-01-01, and REI/1/5334-01-01. This work was supported by the SDAIA–KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA–KAUST AI), by funding from King Abdullah University of Science and Technology (KAUST) – KAUST Center of Excellence for Smart Health (KCSH) under award number 5932, and by funding from King Abdullah University of Science and Technology (KAUST) – KAUST Center of Excellence for Generative AI under award number 5940. We acknowledge support from the KAUST Supercomputing Laboratory.

References

[1] A. Althagafi, F. Zhapa-Camacho and R. Hoehndorf, Prioritizing genomic variants through neuro-symbolic, knowledge-enhanced learning, *Bioinformatics* (2024), btae301. doi:10.1093/bioinformatics/btae301.

[2] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin and G. Sherlock, Gene Ontology: tool for the unification of biology, *Nature Genetics* **25**(1) (2000), 25–29. doi:10.1038/75556.

[3] F. Baader, S. Brandt and C. Lutz, Pushing the $\mathcal{EL}$ Envelope, in: *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence IJCAI-05*, Morgan-Kaufmann Publishers, Edinburgh, UK, 2005.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P.F. Patel-Schneider (eds), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2003.

[5] 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, Curran Associates, Inc., 2013.

[6] 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, Curran Associates, Inc., 2013.

[7] Z. Bouraoui, S. Jameel and S. Schockaert, Inductive Reasoning about Ontologies Using Conceptual Spaces, *Proceedings of the AAAI Conference on Artificial Intelligence* **31**(1) (2017). doi:10.1609/aaai.v31i1.11162.

[8] J. Cao, J. Fang, Z. Meng and S. Liang, Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces, *ACM Computing Surveys* **56**(6) (2024), 1–42–. doi:10.1145/3643806.

[9] P. Cao, Y. Hao, Y. Chen, K. Liu, J. Xu, H. Li, X. Jiang and J. Zhao, Event Ontology Completion with Hierarchical Structure Evolution Networks, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino and K. Bali, eds, Association for Computational Linguistics, Singapore, 2023, pp. 306–320. doi:10.18653/v1/2023.emnlp-main.21. https://aclanthology.org/2023.emnlp-main.21.

[10] J. Chen, P. Hu, E. Jimenez-Ruiz, O.M. Holter, D. Antonyrajah and I. Horrocks, OWL2Vec*: embedding of OWL ontologies, *Machine Learning* (2021). doi:10.1007/s10994-021-05997-6.

[11] J. Chen, A. Althagafi and R. Hoehndorf, Predicting candidate genes from phenotypes, functions and anatomical site of expression, *Bioinformatics* **37**(6) (2020), 853–860. doi:10.1093/bioinformatics/btaa879.

[12] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao and Z. Duan, Knowledge graph completion: A review, *Ieee Access* **8** (2020), 192435–192456.

[13] J. Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran and T. Solorio, eds, Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/V1/N19-1423. https://doi.org/10.18653/v1/n19-1423.

[14] D.M. Dooley, E.J. Griffiths, G.S. Gosal, P.L. Buttigieg, R. Hoehndorf, M.C. Lange, L.M. Schriml, F.S. Brinkman and W.W. Hsiao, FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration, *npj Science of Food* **2**(1) (2018), 23.

[15] C. d'Amato, N. Fanizzi, B. Fazzinga, G. Gottlob and T. Lukasiewicz, Ontology-based semantic search on the Web and its combination with the power of inductive reasoning, *Annals of Mathematics and Artificial Intelligence* **65**(2–3) (2012), 83–121–. doi:10.1007/s10472-012-9309-7.

[16] S. Federhen, The NCBI taxonomy database, *Nucleic acids research* **40**(D1) (2012), D136–D143.

[17] C. Golbreich and I. Horrocks, The OBO to OWL Mapping, GO to OWL 1.1!, in: *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, June 6-7, 2007*, C. Golbreich, A. Kalyanpur and B. Parsia, eds, CEUR Workshop Proceedings, Vol. 258, CEUR-WS.org, 2007. https://ceur-ws.org/Vol-258/paper35.pdf.

[18] T. Hinnerichs and R. Hoehndorf, DTI-Voodoo: machine learning over interaction networks and ontology-based background knowledge predicts drug-target interactions, *Bioinformatics* **37**(24) (2021), 4835–4843. doi:10.1093/bioinformatics/btab548.

[19] T. Hinnerichs, Z. Tang, X. Peng, X. Zhang and R. Hoehndorf, FALCON: Scalable Reasoning over Inconsistent ALC Ontologies, 2024. https://arxiv.org/abs/2208.07628.

[20] M. Horridge and S. Bechhofer, The OWL API: A Java API for OWL Ontologies, *Semant. Web* **2**(1) (2011), 11–21–.

[21] M. Jackermeier, J. Chen and I. Horrocks, Dual Box Embeddings for the Description Logic EL++, in: *Proceedings of the ACM Web Conference 2024*, WWW '24, 2024. doi:10.1145/3589334.3645648.

[22] P. Jaiswal, S. Avraham, K. Ilic, E.A. Kellogg, S. McCouch, A. Pujar, L. Reiser, S.Y. Rhee, M.M. Sachs, M. Schaeffer et al., Plant Ontology (PO): a controlled vocabulary of plant structures and growth stages, *Comparative and functional genomics* **6**(7–8) (2005), 388–397.

[23] Q. Ji, G. Qi, Y. Ye, J. Li, S. Li, J. Ren and S. Lu, Ontology Revision based on Pre-trained Language Models, 2023.

[24] X. Jiang, Y. Huang, M. Nickel and V. Tresp, *Combining Information Extraction, Deductive Reasoning and Machine Learning for Relation Prediction*, in: *The Semantic Web: Research and Applications*, Springer Berlin Heidelberg, 2012, pp. 164–178–. ISSN 1611-3349. ISBN 9783642302848. doi:10.1007/978-3-642-30284-8_18.

[25] Y. Jiang, T.R. Oron, W.T. Clark, A.R. Bankapur, D. D'Andrea, R. Lepore, C.S. Funk, I. Kahanda, K.M. Verspoor, A. Ben-Hur, D.C.E. Koo, D. Penfold-Brown, D. Shasha, N. Youngs, R. Bonneau, A. Lin, S.M.E. Sahraeian, P.L. Martelli, G. Profiti, R. Casadio, R. Cao, Z. Zhong, J. Cheng, A. Altenhoff, N. Skunca, C. Dessimoz, T. Dogan, K. Hakala, S. Kaewphan, F. Mehryary, T. Salakoski, F. Ginter, H. Fang, B. Smithers, M. Oates, J. Gough, P. Törönen, P. Koskinen, L. Holm, C.-T. Chen, W.-L. Hsu, K. Bryson, D. Cozzetto, F. Minneci, D.T. Jones, S. Chapman, D. BKC, I.K. Khan, D. Kihara, D. Ofer, N. Rappoport, A. Stern, E. Cibrian-Uhalte, P. Denny, R.E. Foulger, R. Hieta, D. Legge, R.C. Lovering, M. Magrane, A.N. Melidoni, P. Mutowo-Meullenet, K. Pichler, A. Shypitsyna, B. Li, P. Zakeri, S. ElShal, L.-C. Tranchevent, S. Das, N.L. Dawson, D. Lee, J.G. Lees, I. Sillitoe, P. Bhat, T. Nepusz, A.E. Romero, R. Sasidharan, H. Yang, A. Paccanaro, J. Gillis, A.E. Sedeño-Cortés, P. Pavlidis, S. Feng, J.M. Cejuela, T. Goldberg, T. Hamp, L. Richter, A. Salamov, T. Gabaldon, M. Marcet-Houben, F. Supek, Q. Gong, W. Ning, Y. Zhou, W. Tian, M. Falda, P. Fontana, E. Lavezzo, S. Toppo, C. Ferrari, M. Giollo, D. Piovesan, S.C.E. Tosatto, A. del Pozo, J.M. Fernández, P. Maietta, A. Valencia, M.L. Tress, A. Benso, S. Di Carlo, G. Politano, A. Savino, H.U. Rehman, M. Re, M. Mesiti, G. Valentini, J.W. Bargsten, A.D.J. van Dijk, B. Gemovic, S. Glisic, V. Perovic, V. Veljkovic, N. Veljkovic, D.C. Almeida-e-Silva, R.Z.N. Vencio, M. Sharan, J. Vogel, L. Kansakar, S. Zhang, S. Vucetic, Z. Wang, M.J.E. Sternberg, M.N. Wass, R.P. Huntley, M.J. Martin, C. O'Donovan, P.N. Robinson, Y. Moreau, A. Tramontano, P.C. Babbitt, S.E. Brenner, M. Linial,

C.A. Orengo, B. Rost, C.S. Greene, S.D. Mooney, I. Friedberg and P. Radivojac, An expanded evaluation of protein function prediction methods shows an improvement in accuracy, *Genome Biology* **17**(1) (2016). doi:10.1186/s13059-016-1037-6.

[26] Y. Kazakov, M. Krötzsch and F. Simančík, The Incredible ELK, *Journal of Automated Reasoning* **53**(1) (2013), 1–61–. doi:10.1007/s10817-013-9296-3.

[27] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015.

[28] M. Kulmanov and R. Hoehndorf, DeepGOZero: improving protein function prediction from sequence and zero-shot learning based on ontology axioms, *Bioinformatics* **38**(Supplement_1) (2022), i238–i245.

[29] M. Kulmanov, W. Liu-Wei, Y. Yan and R. Hoehndorf, EL Embeddings: Geometric construction of models for the Description Logic EL ++, in: *International Joint Conference on Artificial Intelligence*, 2019.

[30] M. Kulmanov, F.J. Guzmán-Vega, P. Duek Roggli, L. Lane, S.T. Arold and R. Hoehndorf, Protein function prediction as approximate semantic entailment, *Nature Machine Intelligence* **6**(2) (2024), 220–228–. doi:10.1038/s42256-024-00795-w.

[31] Y. Lan, G. He, J. Jiang, J. Jiang, W.X. Zhao and J.-R. Wen, Complex Knowledge Base Question Answering: A Survey, *IEEE Transactions on Knowledge and Data Engineering* **35**(11) (2023), 11196–11215–. doi:10.1109/tkde.2022.3223858. http://dx.doi.org/10.1109/TKDE.2022.3223858.

[32] L. Li, X. Zhang, Y. Ma, C. Gao, J. Wang, Y. Yu, Z. Yuan and Q. Ma, A knowledge graph completion model based on contrastive learning and relation enhancement method, *Knowledge-Based Systems* **256** (2022), 109889. doi:10.1016/j.knosys.2022.109889.

[33] N. Li, T. Bailleux, Z. Bouraoui and S. Schockaert, Ontology Completion with Natural Language Inference and Concept Embeddings: An Analysis, 2024.

[34] A.L. "Maas, A.Y. "Hannun and A.Y. "Ng, Rectifier nonlinearities improve neural network acoustic models, 2013.

[35] O. Mashkova, F. Zhapa-Camacho and R. Hoehndorf, Enhancing geometric ontology embeddings for EL++ with negative sampling and deductive closure filtering, in: *International Conference on Neural-Symbolic Learning and Reasoning*, Springer, 2024, pp. 331–354.

[36] J. 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*, I. Horrocks, M. Yatskevich and E. Jiménez-Ruiz, eds, CEUR Workshop Proceedings, Vol. 858, CEUR-WS.org, 2012. https://ceur-ws.org/Vol-858/ore2012_paper12.pdf.

[37] C.v. Mering, STRING: a database of predicted functional associations between proteins, *Nucleic Acids Research* **31**(1) (2003), 258–261–. doi:10.1093/nar/gkg034.

[38] S. Mežnar, M. Bevec, N. Lavrač and B. Škrlj, Ontology Completion with Graph-Based Machine Learning: A Comprehensive Evaluation, *Machine Learning and Knowledge Extraction* **4**(4) (2022), 1107–1123–. doi:10.3390/make4040056.

[39] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient Estimation of Word Representations in Vector Space, *CoRR* **abs/1301.3781** (2013). http://arxiv.org/abs/1301.3781.

[40] 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), Stanford University, Palo Alto, California, USA, March 22-24, 2021*, A. Martin, K. Hinkelmann, H. Fill, A. Gerber, D. Lenat, R. Stolle and F. van Harmelen, eds, CEUR Workshop Proceedings, Vol. 2846, CEUR-WS.org, 2021.

[41] C.J. Mungall, C. Torniai, G.V. Gkoutos, S.E. Lewis and M.A. Haendel, Uberon, an integrative multi-species anatomy ontology, *Genome biology* **13** (2012), 1–20.

[42] O.L. Özcep, M. Leemhuis and D. Wolter, Embedding Ontologies in the Description Logic ALC by Axis-Aligned Cones, *Journal of Artificial Intelligence Research* **78** (2023), 217–267–. doi:10.1613/jair.1.13939.

[43] S. Pei, L. Yu, R. Hoehndorf and X. Zhang, Semi-Supervised Entity Alignment via Knowledge Graph Embedding with Awareness of Degree Difference, in: *The World Wide Web Conference*, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 3130–3136–. ISBN 9781450366748. doi:10.1145/3308558.3313646.

[44] X. Peng, Z. Tang, M. Kulmanov, K. Niu and R. Hoehndorf, Description Logic EL++ Embeddings with Intersectional Closure, 2022.

[45] X. Qi, Y. Wei, X. Mei, R. Chellali and S. Yang, *Comparative Analysis of the Linear Regions in ReLU and LeakyReLU Networks*, in: *Neural Information Processing*, Springer Nature Singapore, 2023, pp. 528–539–. ISSN 1865-0937. ISBN 9789819981328. doi:10.1007/978-981-99-8132-8_40.

[46] P. Radivojac and W.T. Clark, Information-theoretic evaluation of predicted ontological annotations, *Bioinformatics* **29**(13) (2013), i53–i61. doi:10.1093/bioinformatics/btt228.

[47] M.Á. Rodríguez-García and R. Hoehndorf, Inferring ontology graph structures using OWL reasoning, *BMC Bioinformatics* **19**(1) (2018). doi:10.1186/s12859-017-1999-8.

[48] Y. Sato, G. Stapleton, M. Jamnik and Z. Shams, Deductive reasoning about expressive statements using external graphical representations, in: *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, Cognitive Science Society, 2018, pp. 0–0, CogSci 2018 ; Conference date: 25-07-2018 Through 28-07-2018.

[49] S. Sharma, S. Sharma and A. Athaiya, ACTIVATION FUNCTIONS IN NEURAL NETWORKS, *International Journal of Engineering Applied Sciences and Technology* **04**(12) (2020), 310–316–. doi:10.33564/ijeast.2020.v04i12.054.

[50] T. Shen, F. Zhang and J. Cheng, A comprehensive overview of knowledge graph completion, *Knowledge-Based Systems* **255** (2022), 109597. doi:10.1016/j.knosys.2022.109597.

[51] Y. Shiraishi and K. Kaneiwa, A Self-matching Training Method with Annotation Embedding Models for Ontology Subsumption Prediction, 2024.

[52] H. Shomer, W. Jin, W. Wang and J. Tang, Toward Degree Bias in Embedding-Based Knowledge Graph Completion, in: *Proceedings of the ACM Web Conference 2023*, WWW '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 705–715–. ISBN 9781450394161. doi:10.1145/3543507.3583544.
[53] R. Sivakumar and P. Arivoli, Ontology visualization PROTÉGÉ tools–a review, *International Journal of Advanced Information Technology (IJAIT) Vol* **1** (2011).
[54] 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.
[55] B. Xiong, N. Potyka, T.-K. Tran, M. Nayyeri and S. Staab, Faithful Embeddings for EL++ Knowledge Bases, in: *Proceedings of the 21st International Semantic Web Conference (ISWC2022)*, 2022, pp. 1–18.
[56] Y. Yang, C. Huang, L. Xia and C. Li, Knowledge Graph Contrastive Learning for Recommendation, in: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, ACM, 2022. doi:10.1145/3477495.3532009.
[57] F. Zhapa-Camacho and R. Hoehndorf, From axioms over graphs to vectors, and back again: evaluating the properties of graph-based ontology embeddings, 2023.
[58] F. Zhapa-Camacho, M. Kulmanov and R. Hoehndorf, mOWL: Python library for machine learning with biomedical ontologies, *Bioinformatics* (2022), btac811. doi:10.1093/bioinformatics/btac811.
[59] Gene Ontology Consortium: going forward, *Nucleic Acids Research* **43**(D1) (2014), D1049–D1056–. doi:10.1093/nar/gku1179.

Appendix A. GO & STRING data Statistics, Train Part

Dataset	GCI0	GCI1	GCI2	GCI3	GCI0_BOT	GCI1_BOT	GCI3_BOT	Classes	Relations	Test axioms
Yeast iw	81,068	11,825	269,567	11,823	0	31	0	61,846	16	12,040
Yeast hf	81,068	11,825	290,433	11,823	0	31	0	61,850	16	1,530

Appendix B. Food Ontology Statistics, Train Part

GCI0	GCI1	GCI2	GCI3	GCI0_BOT	GCI1_BOT	GCI3_BOT	Classes	Relations	Test axioms
21,795	1,267	10,719	897	0	495	0	24,969	43	5,752

Appendix C. Hyperparameters

Dataset	Model	dim	lr	γ	ϵ	δ	λ
Yeast iw	ELEm	100	0.0001	-0.10			
	ELEm+l	50	0.0001	0.00			
	ELBE	200	0.0001	0.00			
	ELBE+l	200	0.0100	0.00	0.001		
	Box^2EL	200	0.0010	0.01		1	0.05
	Box^2EL+l	200	0.0010	0.01	0.010	2	0.05
Yeast hf	ELEm	200	0.0001	0.01			
	ELEm+l	50	0.0001	-0.10			
	ELBE	200	0.0001	0.10			
	ELBE+l	200	0.0001	0.10	0.010		
	Box^2EL	200	0.0100	0.10		4	0.20
	Box^2EL+l	200	0.0100	0.10	0.010	4	0.05
FoodOn	ELEm	400	0.0010	-0.10			
	ELEm+l	400	0.0010	-0.10			
	ELBE	200	0.0100	0.10			
	ELBE+l	200	0.0100	-0.01	0.001		
	Box^2EL	100	0.0100	0.10		1	0.20
	Box^2EL+l	200	0.0010	0.10	0.01	4	0.10

Appendix D. Deductive Closure Computation Example

Let us add two more axioms to the simple ontology example from Section 5.1.4 about proteins $\{P\}$ and $\{Q\}$ having functions $\{GO_1\}$ and $\{GO_2\}$, respectively. ELK will infer the following class hierarchy:

C	Concepts D where $C \sqsubseteq D$
$\bot$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
$\{P\}$	$\{P\}$, B, $\top$
$\{Q\}$	$\{Q\}$, A, $\top$
A	A, $\top$
B	B, $\top$
$\{GO_1\}$	$\{GO_1\}$, $\top$
$\{GO_2\}$	$\{GO_2\}$, $\top$
$\top$	$\top$

In this small protein function prediction example there are two disjointness axioms: $A \sqcap B \sqsubseteq \bot$ and $\{GO_1\} \sqcap \{GO_2\} \sqsubseteq \bot$. Taking into consideration the concept hierarchy and inference rules from part 2 the algorithm will infer the following GCI1 and GCI1_BOT axioms:

C	D	Subsumptions E where $C \sqcap D \sqsubseteq E$
$\bot$	$\bot$	$\bot$
	$\{P\}$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\{Q\}$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	A	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	B	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\{GO_1\}$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\{GO_2\}$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\top$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
$\{P\}$	$\{P\}$	$\{P\}$, B, $\top$
	$\{Q\}$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	A	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	B	$\{P\}$, B, $\top$
	$\{GO_1\}$	$\{P\}$, $\{GO_1\}$, B, $\top$
	$\{GO_2\}$	$\{P\}$, $\{GO_2\}$, B, $\top$
	$\top$	$\{P\}$, B, $\top$
$\{Q\}$	$\{Q\}$	$\{Q\}$, A, $\top$
	A	$\{Q\}$, A, $\top$
	B	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\{GO_1\}$	$\{Q\}$, $\{GO_1\}$, A, $\top$
	$\{GO_2\}$	$\{Q\}$, $\{GO_2\}$, A, $\top$
	$\top$	$\{Q\}$, A, $\top$
A	A	A, $\top$
	B	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\{GO_1\}$	A, $\{GO_1\}$, $\top$
	$\{GO_2\}$	A, $\{GO_2\}$, $\top$
	$\top$	A, $\top$
B	B	B, $\top$
	$\{GO_1\}$	B, $\{GO_1\}$, $\top$
	$\{GO_2\}$	B, $\{GO_2\}$, $\top$
	$\top$	B, $\top$
$\{GO_1\}$	$\{GO_1\}$	$\{GO_1\}$, $\top$
	$\{GO_2\}$	$\bot$, $\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
	$\top$	$\{GO_1\}$, $\top$
$\{GO_2\}$	$\{GO_2\}$	$\{GO_2\}$, $\top$
	$\top$	$\{GO_2\}$, $\top$
$\top$	$\top$	$\top$

For GCI2 axioms $\{P\} \sqsubseteq \exists has\_function.\{GO_1\}$ and $\{Q\} \sqsubseteq \exists has\_function.\{GO_2\}$ the algorithm will output

C	Concepts $D \neq \bot$ where $C \sqsubseteq \exists has\_function.D$
$\bot$	$\{P\}$, $\{Q\}$, A, B, $\{GO_1\}$, $\{GO_2\}$, $\top$
$\{P\}$	$\{GO_1\}$, $\top$
$\{Q\}$	$\{GO_2\}$, $\top$

For GCI3 axioms $\exists has\_function.\{GO_1\} \sqsubseteq B$ and $\exists has\_function.\{GO_2\} \sqsubseteq A$ the algorithm will infer

$C \neq \bot$	Concepts D where $\exists has\_function.C \sqsubseteq D$
$\{P\}$	$\top$
$\{Q\}$	$\top$
A	$\top$
B	$\top$
$\{GO_1\}$	B, $\top$
$\{GO_2\}$	A, $\top$
$\top$	$\top$