Evaluating Different Methods for Semantic Reasoning Over Ontologies

Fernando Zhapa-Camacho[1], Robert Hoehndorf[1]

[1]*Computational Bioscience Research Center, Computer, Electrical & Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, 4700 KAUST, 23955 Thuwal, Saudi Arabia*

Abstract

Reasoning over knowledge bases such as Semantic Web ontologies enables the discovery of new facts from existing knowledge. Knowledge-enhanced machine learning has motivated the development of neuro-symbolic reasoners, which enable faster but approximate computation of new facts or entailments. Neuro-symbolic methods generate vector representations (embeddings) of entities in a knowledge base. We analyze some ontology embedding methods, by implementing them as neuro-symbolic reasoners and evaluating their predictive performance on the datasets and tasks provided by the Semantic Reasoning Evaluation Challenge 2023. We explore two types of embedding methods: graph-based and model-theoretic. Regarding graph-based embeddings, we evaluated the impact of different combinations of graph representation of ontologies with knowledge graph embedding methods. For model-theoretic embeddings, which create models for theories, we evaluate the impact of using several models, enabling approximate semantic entailment.

Keywords

approximate entailment, neuro-symbolic AI, ontology embedding

1. Introduction

Symbolic reasoning over knowledge bases such as ontologies is a computationally expensive task. In recent years, alternative neuro-symbolic approaches have emerged to predict plausible facts leveraging existing background knowledge [1, 2] or generate *models* for a theory [3, 4, 5, 6]. Knowledge-enhanced machine learning uses background knowledge to enhance the capabilities of machine learning algorithms, motivating the development of neuro-symbolic reasoners.

We implemented several neuro-symbolic reasoners based on ontology embedding methods and evaluated their predictive performance of ontology axioms over the datasets of Task 2 of the Semantic Reasoning Evaluation Challenge (SemREC) 2023. We evaluated graph-based ontology embedding methods [2, 7] and geometric embedding methods [3, 4, 5] with respect to their performance in generating entailments. We relied on the implementations found in the mOWL library [8] and include our recent work CatE[7], which is a graph-based method to generate ontology embeddings.

Our contributions are the following:

SemREC'23: Semantic Reasoning Evaluation Challenge, ISWC'23, 06 – 10, Athens, Greece
fernando.zhapacamacho@kaust.edu.sa (F. Zhapa-Camacho); robert.hoehndorf@kaust.edu.sa (R. Hoehndorf)
https://ferzcam.github.io/ (F. Zhapa-Camacho); https://leechuck.de/ (R. Hoehndorf)
0000-0002-0710-2259 (F. Zhapa-Camacho); 0000-0001-8149-5890 (R. Hoehndorf)

- We evaluate the difference between graph-based and geometric model-based embedding methods.
- We analyze the impact of total and injective graph-embedding methods on axiom prediction.
- We analyze the effect of implementing multiple models to achieve approximate semantic entailment.

2. Preliminaries

A Description Logic (DL) language consists of a signature $\Sigma = (\mathbf{C}, \mathbf{R}, \mathbf{I})$, where $\mathbf{C}$ is a set of concept names, $\mathbf{R}$ is a set of role names and $\mathbf{I}$ is a set of individual names. We focus on two DL languages: $\mathcal{ALC}$ and $\mathcal{EL}$. In the language $\mathcal{ALC}$, concept descriptions can be constructed inductively; all concept names are concept descriptions. If A and B are concept descriptions and r a role name, then $A \sqcap B$, $A \sqcup B$, $\neg A$, $\exists r.A$, and $\forall r.A$ are concept descriptions. If A and B are concept descriptions, a and b individual names, r a role name, then $A \sqsubseteq B$, $r(a, b)$, and $B(a)$ are axioms in $\mathcal{ALC}$. Axioms $A \sqsubseteq B$ are called *general concept inclusions* and when A, B are concept names, we will refer to them as *subsumption axioms*. Axioms $B(a)$ are called *class assertion axioms*, and when B is a concept name, we will refer to them as *membership axioms*. A set of axioms is an $\mathcal{ALC}$ theory.

An interpretation $\mathcal{I}$ of an $\mathcal{ALC}$ theory consists of an *interpretation domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$. For every concept name $C \in \mathbf{C}$, $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$; individual name $a \in \mathbf{I}$, $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$; role name $r \in \mathbf{R}$, $r^{\mathcal{I}} \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$; and, inductively:

$$\begin{aligned} \bot^{\mathcal{I}} = \emptyset \qquad \top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad (\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash A^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}} \quad (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}} \\ (\exists R.C)^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \mid \exists b.((a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}) \right\} \\ (\forall R.C)^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \mid \forall b.((a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}) \right\} \end{aligned}$$

Given an interpretation $\mathcal{I}$, an axiom $A \sqsubseteq B$ holds if and only if $A^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Similarly, axioms $A(a)$ and $r(a, b)$ hold if and only if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, respectively. If every axiom holds under $\mathcal{I}$, then $\mathcal{I}$ is a *model* for the theory.

In the language $\mathcal{EL}$, all axioms can be rewritten equivalently using four normal forms [9]:

$$A \sqsubseteq B, \quad A \sqcap B \sqsubseteq C, \quad A \sqsubseteq \exists r.B, \quad \exists r.A \sqsubseteq B$$

where A, B, C are concept names and the concept $\bot$ can only occur in the right side of normal forms 1,3,4. An $\mathcal{EL}$ theory is a subset of a $\mathcal{ALC}$ theory.

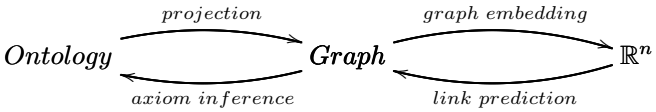

Figure 1: Ontology embedding (left to right) and inference (right to left) processes.

3. Methods

3.1. Graph-based embeddings

A graph-based embedding method is a two-step process [10] (Figure 1). The first step is a *graph projection* function from ontologies into graphs. The second step is performed by a Knowledge Graph Embedding (KGE) method that maps nodes and edge labels to vectors in $\mathbb{R}^n$.

As graph projection methods we chose OWL2Vec* [2] and CatE [7]. CatE was designed to encode the $\mathcal{ALC}$ language and OWL2Vec* can encode more complex DL languages than $\mathcal{ALC}$. OWL2Vec* defines a set of axiom patterns to generate graph edges. The set of patterns might not cover all the axioms existing in the ontology and therefore the projection might not be a total function. On the other hand, CatE represents concepts and axioms as diagrams using their category-theoretical representation. The categorical representation ensures the projection of any concept description and therefore it is a total function. In OWL2Vec*, subsumption axioms $A \sqsubseteq B$ are mapped to the triple $(A, subclassof, B)$ and membership axioms $A(a)$ are mapped into triples $(a, instanceof, A)$. The CatE projection does not only generate edges from axioms but also from complex concept descriptions. However, class assertion axioms are not originally supported. To extend CatE to assertion axioms, we use the same representation that OWL2Vec* utilizes.

Once the graph is generated as a set of triples (h, r, t) where h, t are nodes and t are edge labels standing for relations, we use several KGE methods to embed graph nodes and relations into $\mathbb{R}^n$. KGE methods can be grouped, depending on their scoring method, into distance-based (TransE [11], TransD [12]), similarity-based (DistMult [13]) and neural-network-based (ConvE [14]). For CatE, we also used another KGE method that supports the encoding of hierarchical structures (OrderE [15]). For example, in TransE, the scoring function is:

$$s(h, r, t) = -\|v(h) + v(r) - v(t)\| \tag{1}$$

where $v(x)$ is the vector representation of x.

3.2. Model-theoretic embeddings

Model-theoretic embeddings generate models for a DL theory. An axiom (i.e., $A \sqsubseteq B$ or $A(a)$) is semantically entailed by a DL theory $\mathcal{T}$ if it is true in all models of $\mathcal{T}$. We used three methods to generate model-theoretic embeddings: ELEmbeddings [3], ELBoxEmbeddings [4] and Box^{2}EL [5]. These methods are designed to generate models for theories in the $\mathcal{EL}$ language and they differ by the geometric shapes used for generating models. ELEmbeddings represent

concept descriptions as n-dimensional balls, and axioms $A \sqsubseteq B$ are enforced by the loss function:

$$loss_{A \sqsubseteq B}(A, B) = \max(0, \|c(A) - c(B)\| + rad(A) - rad(B) - \gamma) \tag{2}$$

where $c(\cdot)$ represent the center of a ball and $rad(\cdot)$ is its radius. Equation 2 enforces the ball representing A to be inside the ball representing B, with margin parameter γ.

In ELBoxEmbeddings, concepts are represented as n-dimensional boxes, and axioms $A \sqsubseteq B$ are encoded in the loss function:

$$loss_{A \sqsubseteq B}(A, B) = \max(0, \|c(A) - c(B)\| + offset(A) - offset(B) - \gamma) \tag{3}$$

where $offset(\cdot)$ is the offset of a box.

Box^{2}EL, which also utilizes boxes to encode concepts, implements a generic loss function for axioms $A \sqsubseteq B$:

$$loss_{A \sqsubseteq B}(A, B) = \begin{cases} \max(0, \|c(A) - c(B)\| + offset(A) - offset(B) - \gamma) & \text{if } B \neq \bot \\ \max(0, offset(A) + 1) & \text{otherwise} \end{cases} \tag{4}$$

we call Equations 2, 3, 4 *inclusion* losses.

Class assertion axioms $A(a)$ are formulated as $\{a\} \sqsubseteq A$. Thus, to encode class assertions, we could still use Equations 2, 3, 4. However, we use the following formulations:

$$loss_{A(a)}(a, A) = \sigma(c(a) \cdot c(A)^T + rad(A)) \tag{5}$$

$$loss_{A(a)}(a, A) = \sigma(c(a) \cdot c(A)^T + mean(offset(A))) \tag{6}$$

where Equation 5 is used in ELEmbeddings and Equation 6 is used in ELBoxEmbeddings and Box^{2}EL, where the offset of a box is a vector in $\mathbb{R}^n$. The formulations in Equations 5, 6 obtained better performance and are more computationally efficient than using inclusion losses. We call Equations5, 6 *similarity losses*.

3.3. Computing entailment of axioms

In the case of graph-based generated embeddings, we formulated the entailment computation of axioms $A \sqsubseteq B$ and $A(a)$ as link prediction of edges $(A, subclassof, B)$ and $(a, instanceof, C)$, as shown in Figure 1 (right to left direction). A scoring function $s(A, subclassof, B)$ or $s(a, instanceof, A)$ is used to compute the plausibility of an edge belonging to the graph. The scoring function is given by the KGE method.

For model-theoretic methods, we use the inclusion and similarity losses as scoring functions for subsumption and membership axioms, respectively. To enable approximate semantic entailment, we generate multiple models and aggregate the scores produced by all models. We analyze different aggregator functions such as minimum, maximum, arithmetic mean and median.

3.4. Experimental setup

We evaluated five ontology embedding methods over the datasets provided for the SemREC2023 challenge: ORE1, ORE2, ORE3, OWL2Bench1, OWL2Bench2 and CaLiGraph10e4. Due to computational requirements (memory and time required), we were unable to evaluate CaLiGraph10e5.

For all methods, we used an embedding size of 256 and a batch size of 128. We applied cyclic learning rate [16] with a maximum learning rate of 0.001 and minimum learning rate of 0.0001.

For all the experiments we report Mean Reciprocal Rank (MRR), Mean Rank (MR), Median Rank (MedR), Hits@$\{1, 3, 10, 100\}$ and ROCAUC (AUC). The best results are presented in boldface and the second best results are underlined. All the experiments were performed using a NVIDIA TITAN X (12GB). Due to time constraints and the large number of methods and datasets, we were unable to tune hyperparameters.

3.5. Implementation

We relied on mOWL [8] for the implementation of the methods. mOWL is a Python library that implements many ontology embedding methods and functionalities regarding neural reasoning algorithms. For graph-based methods, we used mOWL functionalities such as the OWL2Vec* graph projection. We used the original CatE implementation, which is not part of mOWL yet but it wraps its functionalities. Furthermore, we used PyKEEN [17] implementations of several knowledge graph embedding methods. To implement model-theoretic embedding methods, we used mOWL implementations of ELEmbeddings, ELBoxEmbeddings and Box^{2}EL. Currently, mOWL only allows generating one model structure. Therefore, we wrapped mOWL functionalities to enable the generation of multiple models with different aggregation strategies.

All the source code and data is available at https://github.com/bio-ontology-research-group/semrec2023.

4. Results

4.1. Graph-based vs. Model-theoretic embeddings (RQ1)

Table 1 shows the results of the prediction of subsumption axioms and Table 2 shows the results of the prediction of membership axioms. In terms of subsumption axiom prediction (Table 1), the performance seems to be dataset-dependent, but there are noted tendencies. Graph-based methods dominate in the ORE datasets. For smaller datasets (OWL2Bench) and larger ones (CaLiGraph), results are not conclusive.

In terms of membership axiom prediction (Table 2), metrics such as ROCAUC and Mean Rank, are dominated by graph-based methods, whereas Hits@k and Median Rank are mostly dominated by model-theoretic methods, except for the smaller datasets (OWL2Bench1, OWL2Bench2). This suggests that although model-theoretic methods can rank positives at high positions, those methods also suffer from outlier predictions (i.e., ranking positive samples very low).

Dataset	Method	MRR	MR	MedR	H@1	H@3	H@10	H@100	AUC
OWL2Bench1	OWL2Vec*	**0.1868**	11.4375	7.0	0.0312	0.1875	**0.6875**	**1.0000**	0.9370
	CatE	0.1599	22.0312	8.5	0.0312	**0.2188**	0.5938	0.9688	0.8765
	ELEm	0.1385	11.5294	9.0	0.0000	0.1176	0.5588	**1.0000**	0.9316
	ELBox	0.1650	15.0588	13.0	**0.0588**	0.1471	0.4412	**1.0000**	0.9080
	Box2EL	0.1657	**9.7647**	**6.0**	0.0000	0.1471	0.6765	**1.0000**	**0.9431**
OWL2Bench2	OWL2Vec*	**0.2440**	13.7188	6.5	**0.0625**	**0.3125**	0.5938	**1.0000**	0.9210
	CatE	0.0906	20.0000	14.5	0.0000	0.0000	0.4688	**1.0000**	0.8800
	ELEm	0.1524	**11.1765**	**6.0**	0.0000	0.0882	0.6176	**1.0000**	**0.9359**
	ELBox	0.1706	15.5294	9.5	0.0588	0.1176	0.5000	**1.0000**	0.9071
	Box2EL	0.1631	13.6471	**6.0**	0.0000	0.1471	**0.7353**	0.9706	0.9282
ORE1	OWL2Vec*	**0.0686**	**1180.6032**	**267.5**	0.0265	**0.0721**	**0.1523**	**0.3784**	**0.8487**
	CatE	0.0091	1271.2363	453.5	0.0017	0.0064	0.0115	0.1497	0.8371
	ELEm	0.0304	1416.5081	531.0	0.0128	0.0264	0.0584	0.2191	0.8184
	ELBox	0.0608	1394.5226	534.0	**0.0294**	0.0708	0.1057	0.2651	0.8213
	Box2EL	0.0369	1387.8662	535.0	0.0000	0.0558	0.0929	0.2553	0.8221
ORE2	OWL2Vec*	**0.0665**	**1275.5541**	**282.5**	**0.0256**	0.0673	**0.1505**	**0.3700**	**0.8368**
	CatE	0.0102	1347.1505	484.5	0.0038	0.0064	0.0115	0.1449	0.8276
	ELEm	0.0297	1401.0730	548.0	0.0119	0.0255	0.0628	0.2076	0.8207
	ELBox	0.0583	1404.2699	582.0	0.0250	**0.0734**	0.1087	0.2606	0.8203
	Box2EL	0.0357	1413.7046	574.5	0.0000	0.0484	0.0900	0.2483	0.8191
ORE3	OWL2Vec*	**0.0770**	**1121.2250**	**241.5**	**0.0316**	**0.0820**	**0.1644**	**0.3907**	**0.8562**
	CatE	0.0082	1290.9701	467.0	0.0021	0.0043	0.0077	0.1533	0.8344
	ELEm	0.0332	1425.1348	527.5	0.0119	0.0327	0.0616	0.2419	0.8171
	ELBox	0.0610	1400.8202	525.0	0.0315	0.0676	0.1076	0.2713	0.8203
	Box2EL	0.0352	1413.5727	515.0	0.0000	0.0438	0.0991	0.2547	0.8186
CaLiGraphi10e4	OWL2Vec*	0.0787	265.0687	14.0	0.0000	0.0126	0.2695	**0.9497**	0.9756
	CatE	0.0652	1371.4366	17.0	**0.0002**	0.0089	0.2032	0.8293	0.8734
	ELEm	0.1047	**102.4787**	**12.0**	0.0000	0.0695	**0.4256**	0.8739	**0.9906**
	ELBox	**0.1119**	294.0854	19.0	0.0001	**0.1250**	0.3441	0.8547	0.9729
	Box2EL	0.0951	263.9739	21.0	0.0000	0.0892	0.2941	0.8551	0.9757

Table 1
Results of subsumption prediction of different ontology embedding methods.

4.2. Analysis of different graph-based methods (RQ2)

In Table 3 we show the results of subsumption prediction, where OWL2Vec* dominates over CatE. More specifically, the combination OWL2Vec*+DistMult is the best setting for all datasets. On the other hand, when predicting membership axioms (Table 4), the results are dataset-dependent and for the smaller datasets such as OWL2Bench1 and OWL2Bench2 OWL2Vec* performs better than CatE in most metrics. However, for larger datasets, CatE outperforms OWL2Vec* in ROCAUC and Mean Rank, indicating can concentrate the predictions of positive samples in high positions. Furthermore, except for ORE2, CatE also dominates Hits@k metrics, and Median Rank, suggesting that CatE is more robust to outlier predictions in these datasets.

4.3. Approximate semantic entailment (RQ3)

Semantic entailment is defined as a relation between models: ϕ is semantically entailed by T, $T \models \phi$, if and only if all models of T are also models of ϕ: $Mod(T) \subseteq Mod(\{\phi\})$. We can approximate semantic entailment by generating finite subsets of $Mod(T)$ using model-generating embedding methods. To evaluate the impact of applying approximate semantic

Dataset	Method	MRR	MR	MedR	H@1	H@3	H@10	H@100	AUC
OWL2Bench1	OWL2Vec*	**0.7750**	**1.0499**	**0.0**	**0.6202**	**0.9356**	**0.9772**	**0.9996**	**0.9946**
	CatE	0.7048	1.8279	**0.0**	0.5650	0.8046	0.9658	**0.9996**	0.9897
	ELEm	0.6979	6.1625	**0.0**	0.5988	0.7727	0.8204	0.9926	0.9619
	ELBox	0.7445	2.7459	**0.0**	0.6018	0.8769	0.9527	0.9930	0.9838
	Box2EL	0.7366	2.9426	**0.0**	0.6045	0.8410	0.9566	0.9930	0.9828
OWL2Bench2	OWL2Vec*	0.6944	**1.1591**	1.0	0.4724	**0.9473**	**0.9838**	**0.9998**	**0.9943**
	CatE	0.5454	2.6888	1.0	0.3351	0.7244	0.9497	**0.9998**	0.9849
	ELEm	0.5353	5.0340	1.0	0.2928	0.7690	0.9053	0.9917	0.9700
	ELBox	**0.7534**	2.2989	**0.0**	**0.6112**	0.8873	0.9716	0.9941	0.9866
	Box2EL	0.7422	2.7902	**0.0**	0.6060	0.8627	0.9567	0.9939	0.9835
ORE1	OWL2Vec*	0.2275	63.0833	16.0	0.1300	0.2621	0.3944	0.8783	0.9920
	CatE	0.2500	**59.2896**	10.0	0.1446	0.2596	0.4845	0.9121	**0.9925**
	ELEm	0.2089	315.0340	33.0	0.1589	0.2031	0.3140	0.5915	0.9596
	ELBox	**0.3979**	78.1807	**3.0**	**0.2460**	**0.4556**	**0.7571**	0.9380	0.9900
	Box2EL	0.3770	64.7830	**3.0**	0.2146	0.4416	0.7570	**0.9425**	0.9918
ORE2	OWL2Vec*	0.0901	62.9938	23.0	0.0008	0.0433	0.3317	**0.9668**	0.9920
	CatE	0.1476	**51.5951**	17.0	0.0215	0.1895	0.3619	0.9467	**0.9935**
	ELEm	0.2110	326.8189	38.0	0.1579	0.2074	0.3295	0.5917	0.9582
	ELBox	**0.3950**	71.6050	**3.0**	**0.2436**	**0.4526**	0.7493	0.9351	0.9909
	Box2EL	0.3803	57.7614	**3.0**	0.2184	0.4460	**0.7545**	0.9436	0.9927
ORE3	OWL2Vec*	0.0929	59.2915	23.0	0.0009	0.0458	0.3368	**0.9708**	0.9925
	CatE	0.1335	**50.1269**	18.0	0.0144	0.1524	0.3602	0.9513	**0.9937**
	ELEm	0.2186	323.0865	52.0	0.1628	0.2324	0.3130	0.5836	0.9586
	ELBox	**0.3930**	72.2575	**3.0**	**0.2389**	0.4523	0.7522	0.9359	0.9908
	Box2EL	0.3887	57.4257	**3.0**	0.2268	**0.4544**	**0.7635**	0.9440	0.9927
CaLiGraphi10e4	OWL2Vec*	0.1025	**42.7590**	13.0	0.0040	0.0512	0.3573	**0.9644**	**0.9961**
	CatE	0.1072	53.1599	12.0	0.0037	0.0610	0.3898	0.9497	0.9952
	ELEm	**0.2786**	180.2772	**5.0**	**0.1289**	**0.3325**	**0.6146**	0.8876	0.9834
	ELBox	0.1052	307.7107	30.5	0.0000	0.1398	0.3064	0.7232	0.9717
	Box2EL	0.1201	229.3583	20.0	0.0000	0.1792	0.3803	0.7827	0.9789

Table 2
Results of membership prediction of different ontology embedding methods.

entailment, we evaluated model-theoretic methods with one model and compared those methods with ten models. For both tasks, subsumption (Table 5) and membership (Table 6) prediction, generating multiple models improves the predictive performance in all metrics. Furthermore, box-based methods (ELBox [4] and Box^{2}EL [5]) outperform ELEmbeddings [3], which is a ball-based geometric method. When generating multiple models, one question to solve is how to aggregate the predictions of all the models to generate a single score. Experiments suggest that applying the arithmetic mean is the most stable option to obtain better performance. However, in small datasets such as OWL2Bench1 and OWL2Bench2, other aggregator operations such as maximum, minimum or median obtain the best performance. In theory, if $T \models \phi$, the minimum degree of truth over all models should be the appropriate aggregator as only this ensures that ϕ is true in all models $Mod(T)$ (or the subset that has been sampled). In practice, however, the model-generating algorithms do not always generate accurate or "good" models where statements that should be true are actually true; therefore, other aggregators achieve better results as they can effectively ignore models that are degenerate.

Dataset	Method	MRR	MR	MedR	H@1	H@3	H@10	H@100	AUC
OWL2Bench1	OWL2Vec* + TransE	0.3263	14.1250	3.0	0.1250	0.4688	0.6562	0.9688	0.9241
	OWL2Vec* + TransD	0.1868	**11.4375**	7.0	0.0312	0.1875	0.6875	**1.0000**	**0.9370**
	OWL2Vec* + DistMult	**0.5230**	19.1875	**1.0**	**0.4062**	**0.6250**	**0.7188**	0.9688	0.8924
	OWL2Vec* + ConvE	0.2812	18.8438	13.0	0.2188	0.2500	0.3438	0.9688	0.8918
	CatE + TransE	0.2603	24.4062	18.5	0.2188	0.2188	0.3125	0.9688	0.8542
	CatE + TransD	0.0714	39.9688	13.0	0.0000	0.0000	0.4375	0.8125	0.7559
	CatE + DistMult	0.1491	24.0312	6.5	0.0000	0.2188	0.6250	0.9062	0.8615
	CatE + ConvE	0.1029	26.1562	10.0	0.0000	0.0625	0.4688	0.9375	0.8460
	CatE + OrderE	0.1599	22.0312	8.5	0.0312	0.2188	0.5938	0.9688	0.8765
OWL2Bench2	OWL2Vec* + TransE	0.3187	18.8750	5.5	0.1250	0.4688	0.5938	**1.0000**	0.8883
	OWL2Vec* + TransD	0.2440	**13.7188**	6.5	0.0625	0.3125	0.5938	**1.0000**	**0.9210**
	OWL2Vec* + DistMult	**0.4689**	27.3438	**2.5**	**0.3750**	**0.5000**	**0.6250**	0.9062	0.8398
	OWL2Vec* + ConvE	0.2623	19.9375	9.5	0.1875	0.2188	0.5000	0.9375	0.8931
	CatE + TransE	0.1295	31.0938	24.5	0.0625	0.1250	0.2188	0.9375	0.8153
	CatE + TransD	0.0335	47.9062	37.0	0.0000	0.0000	0.0312	0.9062	0.7031
	CatE + DistMult	0.1899	23.1875	10.0	0.0000	0.2812	0.4688	0.9375	0.8661
	CatE + ConvE	0.1008	30.6875	24.5	0.0000	0.0938	0.2500	0.9375	0.8187
	CatE + OrderE	0.0906	20.0000	14.5	0.0000	0.0000	0.4688	**1.0000**	0.8800
ORE1	OWL2Vec* + TransE	0.0236	1493.0755	625.0	0.0085	0.0209	0.0461	0.1890	0.8086
	OWL2Vec* + TransD	0.0399	1357.4646	585.5	0.0188	0.0363	0.0806	0.2436	0.8260
	OWL2Vec* + DistMult	**0.0686**	**1180.6032**	**267.5**	**0.0265**	**0.0721**	**0.1523**	**0.3784**	**0.8487**
	OWL2Vec* + ConvE	0.0199	1422.8311	665.5	0.0026	0.0154	0.0444	0.2026	0.8176
	CatE + TransE	0.0131	1389.5064	590.0	0.0030	0.0047	0.0128	0.1924	0.8219
	CatE + TransD	0.0091	1271.2363	453.5	0.0017	0.0064	0.0115	0.1497	0.8371
	CatE + DistMult	0.0141	1864.6156	839.5	0.0000	0.0055	0.0346	0.2278	0.7609
	CatE + ConvE	0.0066	1744.1984	851.0	0.0000	0.0034	0.0077	0.1246	0.7763
	CatE + OrderE	0.0072	1345.3161	554.0	0.0013	0.0026	0.0090	0.0943	0.8276
ORE2	OWL2Vec* + TransE	0.0250	1544.5610	665.0	0.0090	0.0200	0.0490	0.2084	0.8023
	OWL2Vec* + TransD	0.0466	1411.4284	540.5	0.0213	0.0541	0.0776	0.2413	0.8194
	OWL2Vec* + DistMult	**0.0665**	**1275.5541**	**282.5**	**0.0256**	**0.0673**	**0.1505**	**0.3700**	0.**8368**
	OWL2Vec* + ConvE	0.0311	1448.4062	662.5	0.0136	0.0298	0.0563	0.2118	0.8147
	CatE + TransE	0.0252	1490.2349	604.0	0.0132	0.0222	0.0388	0.1594	0.8093
	CatE + TransD	0.0102	1347.1505	484.5	0.0038	0.0064	0.0115	0.1449	0.8276
	CatE + DistMult	0.0140	1951.5132	920.0	0.0000	0.0060	0.0269	0.2340	0.7502
	CatE + ConvE	0.0082	1735.8444	854.0	0.0000	0.0030	0.0171	0.1368	0.7778
	CatE + OrderE	0.0070	1392.9885	544.0	0.0004	0.0021	0.0111	0.1228	0.8218
ORE3	OWL2Vec* + TransE	0.0275	1507.8412	672.0	0.0090	0.0256	0.0581	0.2054	0.8065
	OWL2Vec* + TransD	0.0533	1392.0944	603.5	0.0286	0.0598	0.0897	0.2472	0.8214
	OWL2Vec* + DistMult	**0.0770**	**1121.2250**	**241.5**	**0.0316**	**0.0820**	**0.1644**	**0.3907**	**0.8562**
	OWL2Vec* + ConvE	0.0391	1414.6985	649.5	0.0243	0.0342	0.0581	0.2208	0.8185
	CatE + TransE	0.0133	1411.1435	566.0	0.0030	0.0038	0.0149	0.2054	0.8189
	CatE + TransD	0.0082	1290.9701	467.0	0.0021	0.0043	0.0077	0.1533	0.8344
	CatE + DistMult	0.0146	1921.1264	915.5	0.0000	0.0060	0.0384	0.2412	0.7534
	CatE + ConvE	0.0063	1697.5436	834.5	0.0000	0.0021	0.0068	0.1243	0.7821
	CatE + OrderE	0.0064	1341.4317	522.5	0.0000	0.0021	0.0094	0.1127	0.8279
CaLiGraphi10e4	OWL2Vec* + TransE	0.2037	499.8468	**13.0**	0.0276	**0.3443**	**0.4707**	0.7878	0.9539
	OWL2Vec* + TransD	**0.2092**	456.6912	26.0	**0.1324**	0.2168	0.3631	0.7970	0.9579
	Owl2vec* + DistMult	0.0787	**265.0687**	14.0	0.0000	0.0126	0.2695	**0.9497**	**0.9756**
	OWL2Vec* + ConvE	0.0655	499.5322	25.0	0.0000	0.0231	0.2391	0.7569	0.9539
	CatE + TransE	0.1517	1680.7490	56.0	0.0979	0.1594	0.2443	0.5897	0.8449
	CatE + TransD	0.0245	1749.0114	81.0	0.0000	0.0000	0.0276	0.5486	0.8386
	CatE + DistMult	0.0652	1371.4366	17.0	0.0002	0.0089	0.2032	0.8293	0.8734
	CatE + ConvE	0.0245	1749.0114	81.0	0.0000	0.0000	0.0276	0.5486	0.8386
	CatE + OrderE	0.0951	1634.8890	34.0	0.0424	0.0649	0.2303	0.7071	0.8491

Table 3
Results of subsumption prediction of different graph-based embedding methods.

Dataset	Method	MRR	MR	MedR	H@1	H@3	H@10	H@100	AUC
OWL2Bench1	OWL2Vec* + TransE	**0.7750**	**1.0499**	**0.0**	**0.6202**	**0.9356**	**0.9772**	0.9996	**0.9946**
	OWL2Vec* + TransD	0.7445	1.3268	**0.0**	0.5913	0.8809	0.9693	**1.0000**	0.9928
	OWL2Vec* + DistMult	0.3314	4.8852	2.0	0.0215	0.6141	0.9636	0.9860	0.9718
	OWL2Vec* + ConvE	0.3535	3.2873	2.0	0.0377	0.7105	0.9698	0.9969	0.9821
	CatE + TransE	0.7048	1.8279	**0.0**	0.5650	0.8046	0.9658	0.9996	0.9897
	CatE + TransD	0.5808	1.9619	1.0	0.3530	0.7906	0.9750	0.9996	0.9895
	CatE + DistMult	0.2974	19.5164	2.0	0.0000	0.6583	0.8673	0.8901	0.8769
	CatE + ConvE	0.3115	3.3184	3.0	0.0307	0.4967	0.9715	0.9996	0.9818
	CatE + OrderE	0.4498	4.0758	2.0	0.2011	0.6746	0.8861	0.9996	0.9764
OWL2Bench2	OWL2Vec* + TransE	0.6944	**1.1591**	1.0	0.4724	**0.9473**	**0.9838**	0.9998	**0.9943**
	OWL2Vec* + TransD	**0.7572**	1.1877	**0.0**	**0.6135**	0.8756	0.9781	0.9998	0.9937
	OWL2Vec* + DistMult	0.3310	4.2995	2.0	0.0165	0.6288	0.9675	0.9883	0.9755
	OWL2Vec* + ConvE	0.3552	2.8420	2.0	0.0300	0.7259	0.9752	0.9977	0.9849
	CatE + TransE	0.5454	2.6888	1.0	0.3351	0.7244	0.9497	0.9998	0.9849
	CatE + TransD	0.3190	3.3538	3.0	0.0714	0.3626	0.9781	**1.0000**	0.9814
	CatE + DistMult	0.3042	15.5441	2.0	0.0000	0.6606	0.8954	0.9141	0.9024
	CatE + ConvE	0.3051	3.2765	3.0	0.0241	0.4855	0.9752	**1.0000**	0.9821
	CatE + OrderE	0.3109	5.3626	3.0	0.0929	0.3428	0.9094	0.9998	0.9684
ORE1	OWL2Vec* + TransE	0.2275	63.0833	16.0	0.1300	**0.2621**	0.3944	0.8783	0.9920
	OWL2Vec* + TransD	0.2269	70.5858	18.0	**0.1510**	0.2248	0.3631	0.8651	0.9910
	OWL2Vec* + DistMult	0.0920	76.5117	23.0	0.0009	0.0461	0.3370	**0.9657**	0.9903
	OWL2Vec* + ConvE	0.0705	89.3093	39.0	0.0033	0.0435	0.2368	0.8186	0.9886
	CatE + TransE	**0.2500**	**59.2896**	**10.0**	0.1446	0.2596	**0.4845**	0.9121	**0.9925**
	CatE + TransD	0.0682	95.0810	40.0	0.0120	0.0532	0.1524	0.8110	0.9879
	CatE + DistMult	0.0955	116.4503	23.0	0.0001	0.0544	0.3447	0.9538	0.9851
	CatE + ConvE	0.0728	104.6214	40.0	0.0048	0.0604	0.2173	0.8033	0.9867
	CatE + OrderE	0.1438	59.7547	17.0	0.0288	0.1618	0.3610	0.9397	0.9924
ORE2	OWL2Vec* + TransE	**0.2443**	63.7438	**16.0**	0.1555	**0.2598**	**0.3944**	0.8758	0.9919
	OWL2Vec* + TransD	0.2408	71.5020	20.0	**0.1700**	0.2429	0.3532	0.8654	0.9909
	OWL2Vec* + DistMult	0.0901	62.9938	23.0	0.0008	0.0433	0.3317	**0.9668**	0.9920
	OWL2Vec* + ConvE	0.0557	91.7927	43.0	0.0020	0.0227	0.1838	0.7980	0.9883
	CatE + TransE	0.1482	86.7392	39.0	0.1085	0.1302	0.1964	0.7807	0.9890
	CatE + TransD	0.0486	96.1510	43.0	0.0105	0.0313	0.0689	0.7895	0.9878
	CatE + DistMult	0.0941	106.9611	23.0	0.0001	0.0531	0.3358	0.9554	0.9864
	CatE + ConvE	0.0513	118.9734	49.0	0.0042	0.0323	0.1358	0.7546	0.9848
	CatE + OrderE	0.1476	**51.5951**	17.0	0.0215	0.1895	0.3619	0.9467	**0.9935**
ORE3	OWL2Vec* + TransE	0.2432	63.4592	17.0	0.1545	0.2579	0.3978	0.8794	0.9919
	OWL2Vec* + TransD	0.2338	69.4029	19.0	**0.1578**	0.2371	0.3655	0.8618	0.9912
	OWL2Vec* + DistMult	0.0929	59.2915	23.0	0.0009	0.0458	0.3368	**0.9708**	0.9925
	OWL2Vec* + ConvE	0.0577	85.9952	41.0	0.0013	0.0213	0.2068	0.8151	0.9890
	CatE + TransE	**0.2589**	53.4928	**9.0**	0.1464	**0.2773**	**0.5074**	0.9127	0.9932
	CatE + TransD	0.0246	131.6056	94.0	0.0011	0.0061	0.0433	0.5271	0.9832
	CatE + DistMult	0.0929	99.5963	23.0	0.0000	0.0488	0.3360	0.9567	0.9873
	CatE + ConvE	0.0585	104.0308	44.0	0.0024	0.0364	0.1769	0.7933	0.9867
	CatE + OrderE	0.1335	**50.1269**	18.0	0.0144	0.1524	0.3602	0.9513	**0.9937**
CaLiGraphi10e4	OWL2Vec* + TransE	0.1624	298.3805	19.0	0.0000	0.2784	0.3908	0.7729	0.9725
	OWL2Vec* + TransD	0.0338	500.3494	35.0	0.0000	0.0000	0.0682	0.6996	0.9539
	OWL2Vec* + DistMult	0.1025	**42.7590**	13.0	0.0040	0.0512	0.3573	**0.9644**	**0.9961**
	OWL2Vec* + ConvE	0.0472	459.0266	44.0	0.0000	0.0022	0.1854	0.6057	0.9577
	CatE + TransE	**0.2008**	169.5685	13.0	**0.0701**	**0.3120**	**0.4527**	0.8461	0.9844
	CatE + TransD	0.0453	291.8673	43.0	0.0000	0.0226	0.1218	0.7578	0.9731
	CatE + DistMult	0.1072	53.1599	**12.0**	0.0037	0.0610	0.3898	0.9497	0.9952
	CatE + ConvE	0.0453	291.8673	43.0	0.0000	0.0226	0.1218	0.7578	0.9731
	CatE + OrderE	0.1102	219.6210	25.0	0.0500	0.0826	0.2429	0.8174	0.9798

Table 4
Results of membership prediction of different graph-based embedding methods.

Dataset	Method	MRR	MR	MedR	H@1	H@3	H@10	H@100	AUC
OWL2Bench1	ELEm 1	0.0834	25.0000	18.0	0.0000	0.0294	0.4118	0.9706	0.8528
	ELEm 10 (mean)	0.1385	11.5294	9.0	0.0000	0.1176	0.5588	**1.0000**	0.9316
	ELBox 1	0.1403	16.2059	16.0	0.0294	0.1176	0.4118	**1.0000**	0.9018
	ELBox 10 (max)	0.1650	15.0588	13.0	**0.0588**	**0.1471**	0.4412	**1.0000**	0.9080
	Box2EL 1	0.1419	10.7941	7.5	0.0000	0.1176	0.6471	**1.0000**	0.9369
	Box2EL 10 (min)	**0.1657**	**9.7647**	**6.0**	0.0000	**0.1471**	**0.6765**	**1.0000**	**0.9431**
OWL2Bench2	ELEm 1	**0.2087**	15.1176	7.0	**0.0882**	0.1176	0.5588	0.9706	0.9189
	ELEm 10 (median)	0.1524	**11.1765**	**6.0**	0.0000	0.0882	0.6176	**1.0000**	**0.9359**
	ELBox 1	0.1950	20.2059	9.0	0.0294	**0.2353**	0.5000	0.9412	0.8866
	ELBox 10 (max)	0.1706	15.5294	9.5	0.0588	0.1176	0.5000	**1.0000**	0.9071
	Box2EL 1	0.1843	14.8529	6.5	0.0000	0.2059	0.7059	0.9706	0.9204
	Box2EL 10 (mean)	0.1631	13.6471	**6.0**	0.0000	0.1471	**0.7353**	0.9706	0.9282
ORE1	ELEm 1	0.0226	1471.9015	**526.5**	0.0064	0.0183	0.0452	0.2187	0.8113
	ELEm 10 (mean)	0.0304	1416.5081	531.0	0.0128	0.0264	0.0584	0.2191	0.8184
	ELBox 1	0.0397	1435.4416	574.0	0.0145	0.0418	0.0793	0.2434	0.8160
	ELBox 10 (mean)	**0.0608**	1394.5226	534.0	**0.0294**	**0.0708**	**0.1057**	**0.2651**	0.8213
	Box2EL 1	0.0174	1452.5929	559.5	0.0000	0.0136	0.0405	0.2315	0.8138
	Box2EL 10 (mean)	0.0369	**1387.8662**	535.0	0.0000	0.0558	0.0929	0.2553	**0.8221**
ORE2	ELEm 1	0.0238	1469.2020	559.0	0.0076	0.0208	0.0420	0.2003	0.8120
	ELEm 10 (mean)	0.0297	**1401.0730**	**548.0**	0.0119	0.0255	0.0628	0.2076	**0.8207**
	ELBox 1	0.0416	1456.2347	621.0	0.0149	0.0454	0.0866	0.2508	0.8137
	ELBox 10 (mean)	**0.0583**	1404.2699	582.0	**0.0250**	**0.0734**	**0.1087**	**0.2606**	0.8203
	Box2EL 1	0.0179	1471.0136	569.5	0.0000	0.0166	0.0386	0.2203	0.8118
	Box2EL 10 (mean)	0.0357	1413.7046	574.5	0.0000	0.0484	0.0900	0.2483	0.8191
ORE2	ELEm 1	0.0238	1516.7747	518.0	0.0085	0.0179	0.0472	0.1913	0.8054
	ELEm 10 (mean)	0.0332	1425.1348	527.5	0.0119	0.0327	0.0616	0.2419	0.8171
	ELBox 1	0.0201	1459.2045	544.0	0.0051	0.0145	0.0366	0.2321	0.8128
	ELBox 10 (mean)	**0.0610**	**1400.8202**	525.0	**0.0315**	**0.0676**	**0.1076**	**0.2713**	**0.8203**
	Box2EL 1	0.0203	1473.7360	546.0	0.0000	0.0187	0.0519	0.2360	0.8109
	Box2EL 10 (mean)	0.0352	1413.5727	**515.0**	0.0000	0.0438	0.0991	0.2547	0.8186
CaLiGraphi10e4	ELEm 1	0.0817	148.5252	20.0	0.0000	0.0545	0.2917	0.8584	0.9864
	ELEm 10 (mean)	0.1047	**102.4787**	**12.0**	0.0000	0.0695	**0.4256**	**0.8739**	**0.9906**
	ELBox 1	0.0549	316.1944	55.0	**0.0131**	0.0363	0.1156	0.7145	0.9709
	ELBox 10 (mean)	**0.1119**	294.0854	19.0	0.0001	**0.1250**	0.3441	0.8547	0.9729
	Box2EL 1	0.0430	305.4137	60.0	0.0000	0.0258	0.1069	0.7021	0.9719
	Box2EL 10 (mean)	0.0951	263.9739	21.0	0.0000	0.0892	0.2941	0.8551	0.9757

Table 5
Results of membership prediction of different model-theoretic embedding methods.tab:subsumption

5. Conclusion

We have evaluated different ontology embedding methods using the datasets provided for the SemREC challenge. Our experiments show that different kinds of methods (graph-based, model-theoretic) have different properties when predicting subsumption or membership axioms. Furthermore, for graph-based methods, we evaluated different graph projections with several KGE embedding methods. We also analyzed the impact of generating multiple models for model-theoretic methods and the difference between different aggregation strategies.

Dataset	Method	MRR	MR	MedR	H@1	H@3	H@10	H@100	AUC
OWL2Bench1	ELEm 1	0.6366	6.1213	1.0	0.4801	0.7770	0.8546	0.9917	0.9625
	ELEm 10 (mean)	0.6979	6.1625	**0.0**	0.5988	0.7727	0.8204	0.9926	0.9619
	ELBox 1	0.6737	3.4074	**0.0**	0.5042	0.8235	0.9417	0.9926	0.9799
	ELBox 10 (mean)	**0.7445**	**2.7459**	**0.0**	0.6018	**0.8769**	0.9527	**0.9930**	**0.9838**
	Box2EL 1	0.6512	3.6925	1.0	0.4928	0.7731	0.9321	**0.9930**	0.9781
	Box2EL 10 (mean)	0.7366	2.9426	**0.0**	**0.6045**	0.8410	**0.9566**	**0.9930**	0.9828
OWL2Bench2	ELEm 1	0.5952	6.3944	1.0	0.4320	0.7381	0.8301	0.9910	0.9609
	ELEm 10 (max)	0.5353	5.0340	1.0	0.2928	0.7690	0.9053	0.9917	0.9700
	ELBox 1	0.6787	2.9211	**0.0**	0.5085	0.8242	0.9538	0.9939	0.9829
	ELBox 10 (mean)	**0.7534**	2.2989	**0.0**	**0.6112**	**0.8873**	**0.9716**	0.9941	**0.9866**
	Box2EL 1	0.6534	3.4571	1.0	0.4823	0.7938	0.9355	**0.9944**	0.9796
	Box2EL 10 (mean)	0.7422	**2.7902**	**0.0**	0.6060	0.8627	0.9567	0.9939	0.9835
ORE1	ELEm 1	0.1770	331.1202	65.0	0.1242	0.1810	0.2846	0.5543	0.9576
	ELEm 10 (mean)	0.2089	315.0340	33.0	0.1589	0.2031	0.3140	0.5915	0.9596
	ELBox 1	0.3237	106.8702	6.0	0.1989	0.3661	0.5905	0.8758	0.9864
	ELBox 10 (mean)	**0.3979**	78.1807	**3.0**	**0.2460**	**0.4556**	**0.7571**	0.9380	0.9900
	Box2EL 1	0.2704	91.3420	8.0	0.1517	0.2979	0.5343	0.8787	0.9884
	Box2EL 10 (mean)	0.3770	**64.7830**	**3.0**	0.2146	0.4416	0.7570	**0.9425**	**0.9918**
ORE2	ELEm 1	0.1544	359.6451	111.0	0.1131	0.1479	0.2461	0.4865	0.9540
	ELEm 10 (mean)	0.2110	326.8189	38.0	0.1579	0.2074	0.3295	0.5917	0.9582
	ELBox 1	0.3125	99.8090	7.0	0.1910	0.3529	0.5677	0.8747	0.9873
	ELBox 10 (mean)	**0.3950**	71.6050	**3.0**	**0.2436**	**0.4526**	0.7493	0.9351	0.9909
	Box2EL 1	0.2590	79.4612	8.0	0.1313	0.2893	0.5524	0.8857	0.9899
	Box2EL 10 (mean)	0.3803	**57.7614**	**3.0**	0.2184	0.4460	**0.7545**	**0.9436**	**0.9927**
ORE3	ELEm 1	0.1716	345.9059	61.0	0.1223	0.1704	0.2645	0.5573	0.9556
	ELEm 10 (mean)	0.2186	323.0865	52.0	0.1628	0.2324	0.3130	0.5836	0.9586
	ELBox 1	0.3117	99.5851	6.0	0.1833	0.3609	0.5799	0.8689	0.9873
	ELBox 10 (mean)	**0.3930**	72.2575	**3.0**	**0.2389**	0.4523	0.7522	0.9359	0.9908
	Box2EL 1	0.2469	78.0290	8.0	0.1187	0.2728	0.5452	0.8889	0.9901
	Box2EL 10 (mean)	0.3887	**57.4257**	**3.0**	0.2268	**0.4544**	**0.7635**	**0.9440**	**0.9927**
CaLiGraph10e4	ELEm 1	0.2161	187.7880	10.0	0.0976	0.2387	0.4809	0.8614	0.9827
	ELEm 10 (mean)	**0.2786**	**180.2772**	**5.0**	**0.1289**	**0.3325**	**0.6146**	**0.8876**	**0.9834**
	ELBox 1	0.0195	623.4732	183.0	0.0000	0.0062	0.0416	0.3600	0.9425
	ELBox 10 (mean)	0.1052	307.7107	30.5	0.0000	0.1398	0.3064	0.7232	0.9717
	Box2EL 1	0.0244	521.6279	140.0	0.0000	0.0109	0.0572	0.4177	0.9519
	Box2EL 10 (mean)	0.1201	229.3583	20.0	0.0000	0.1792	0.3803	0.7827	0.9789

Table 6
Results of membership prediction of different model-theoretic embedding methods.

References

[1] M. Kulmanov, F. Z. Smaili, X. Gao, R. Hoehndorf, Semantic similarity and machine learning with ontologies, Briefings in Bioinformatics 22 (2020). doi:10.1093/bib/bbaa199, bbaa199.

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

[3] M. Kulmanov, W. Liu-Wei, Y. Yan, R. Hoehndorf, El embeddings: Geometric construction of models for the description logic el ++, in: International Joint Conference on Artificial Intelligence, 2019.

[4] X. Peng, Z. Tang, M. Kulmanov, K. Niu, R. Hoehndorf, Description logic el++ embeddings

with intersectional closure, 2022. arXiv:2202.14018.
[5] M. Jackermeier, J. Chen, I. Horrocks, Box^{2}el: Concept and role box embeddings for the description logic el++, 2023. arXiv:2301.11118.
[6] Z. Tang, T. Hinnerichs, X. Peng, X. Zhang, R. Hoehndorf, FALCON: Faithful neural semantic entailment over alc ontologies, 2023. arXiv:2208.07628.
[7] F. Zhapa-Camacho, R. Hoehndorf, Cate: Embedding alc ontologies using category-theoretical semantics, 2023. arXiv:2305.07163.
[8] F. Zhapa-Camacho, M. Kulmanov, R. Hoehndorf, mOWL: Python library for machine learning with biomedical ontologies, Bioinformatics 39 (2022) btac811. URL: https://doi.org/10.1093/bioinformatics/btac811. doi:10.1093/bioinformatics/btac811.
[9] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[10] F. Zhapa-Camacho, R. Hoehndorf, From axioms over graphs to vectors, and back again: evaluating the properties of graph-based ontology embeddings, 2023. arXiv:2303.16519.
[11] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems, volume 26, Curran Associates, Inc., 2013.
[12] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Annual Meeting of the Association for Computational Linguistics, 2015. URL: https://api.semanticscholar.org/CorpusID:11202498.
[13] B. Yang, W. tau Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, 2015. arXiv:1412.6575.
[14] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, ArXiv abs/1707.01476 (2017). URL: https://api.semanticscholar.org/CorpusID:4328400.
[15] I. Vendrov, R. Kiros, S. Fidler, R. Urtasun, Order-embeddings of images and language (2016). ArXiv:1511.06361 [cs].
[16] L. N. Smith, Cyclical learning rates for training neural networks, 2017. arXiv:1506.01186.
[17] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, J. Lehmann, PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings, Journal of Machine Learning Research 22 (2021) 1–6. URL: http://jmlr.org/papers/v22/20-825.html.