Learning Surrogate Models for Scoring OWL Axioms: Why it works and how to make it scalable

Andrea G. B. Tettamanzi

Université Côte d'Azur, CNRS, Inria, I3S, France

LIST, Luxembourg, September 6, 2024

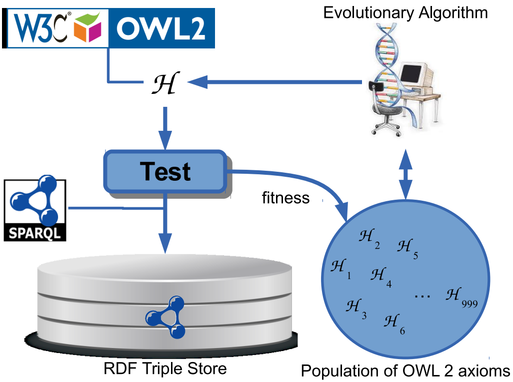
Research Question and Contribution

- Candidate axiom scoring is key to automatic axiom discovery.
- We deal with the problem of learning a surrogate model of a very accurate but computationally heavy scoring heuristics
- We got good results using a kernel-based representation of formulas

$\Rightarrow$ How can we obtain scalability for large ontologies?

- train a classifier against a set of scored axioms
- kernel-based representation of axioms
- ontological semantic similarity measure

Motivation: RDF Mining



Problem:

How to test OWL axioms under the open-world assumption?

Axiom Scoring

Axiom Testing

given a *hypothesis* about the relations holding among some entities of a domain, evaluate its credibility based on the available *evidence*

$$h \overset{?}{\Longleftarrow} e$$

OWL | RDF

Fundamental problem in epistemology, with ramifications in statistical inference, data mining, inductive reasoning, medical diagnosis, judicial decision making, and even the philosophy of science.

Confirmation is central to this problem

Extended hypothetico-deductivism:

- e confirms h if $h \models e$
- e disconfirms h if $e \models \neg h$

Content of an Axiom

Definition (Content of Axiom ϕ)

Given an RDF datset $\mathcal{K}$, $content_{\mathcal{K}}(\phi)$, is defined as the set of all the basic statements of $D_{\mathcal{K}}(\phi)$, the development of ϕ.

E.g., $\phi = \texttt{dbo:LaunchPad} \sqsubseteq \texttt{dbo:Infrastructure}$

Let us assume $\mathcal{K} =$ DBpedia; then

$$
\begin{aligned}
t(\phi; x, y) &= \forall x(\neg \texttt{dbo:LaunchPad}(x) \vee \texttt{dbo:Infrastructure}(x)) \\
D_{\mathcal{K}}(\phi) &= \bigwedge_{c \in I(\mathcal{K})} (\neg \texttt{dbo:LaunchPad}(c) \vee \texttt{dbo:Infrastructure}(c)) \\
content(\phi) &= \{\neg \texttt{dbo:LaunchPad}(c) \vee \texttt{dbo:Infrastructure}(c) : \\
&\qquad c \text{ is a resource occurring in DBPedia}\}
\end{aligned}
$$

By construction, for all $\psi \in content(\phi)$, $\phi \models \psi$.

Confirmation and Counterexample of an Axiom

Given $\psi \in \mathit{content}(\phi)$ and an RDF dataset $\mathcal{K}$, three cases:

1. $\mathcal{K} \models \psi$: $\longrightarrow \psi$ is a *confirmation* of ϕ;
2. $\mathcal{K} \models \neg\psi$: $\longrightarrow \psi$ is a *counterexample* of ϕ;
3. $\mathcal{K} \not\models \psi$ and $\mathcal{K} \not\models \neg\psi$: $\longrightarrow \psi$ is neither of the above

Selective confirmation: a ψ favoring ϕ rather than $\neg\phi$.
$\phi = \texttt{Raven} \sqsubseteq \texttt{Black} \longrightarrow \psi =$ a black raven (vs. a green apple)

Idea

Restrict $\mathit{content}(\phi)$ just to those ψ which can be counterexamples of ϕ. Leave out all ψ which would be trivial confirmations of ϕ.

Support, Confirmation, and Counterexample of an Axiom

Definition

Given axiom ϕ, let us define

$$\begin{aligned} u_\phi &= \|content(\phi)\| \text{ (a.k.a. the } support \text{ of } \phi) \\ u_\phi^+ &= \text{the number of confirmations of } \phi \\ u_\phi^- &= \text{the number of counterexamples of } \phi \end{aligned}$$

Some properties:

- $u_\phi^+ + u_\phi^- \leq u_\phi$ (there may be ψ s.t. $\mathcal{K} \not\models \psi$ and $\mathcal{K} \not\models \neg\psi$)
- $u_\phi^+ = u_{\neg\phi}^-$ (confirmations of ϕ are counterexamples of $\neg\phi$)
- $u_\phi^- = u_{\neg\phi}^+$ (counterexamples of ϕ are confirmations of $\neg\phi$)
- $u_\phi = u_{\neg\phi}$ (ϕ and $\neg\phi$ have the same support)

Possibility Theory

Definition (Possibility Distribution)

$$\pi : \Omega \to [0, 1]$$

Definition (Possibility and Necessity Measures)

$$\begin{aligned} \Pi(A) &= \max_{\omega \in A} \pi(\omega); \\ N(A) &= 1 - \Pi(\bar{A}) = \min_{\omega \in \bar{A}} \{1 - \pi(\omega)\}. \end{aligned}$$

For all subsets $A \subseteq \Omega$,

1. $\Pi(\emptyset) = N(\emptyset) = 0, \quad \Pi(\Omega) = N(\Omega) = 1;$
2. $\Pi(A) = 1 - N(\bar{A})$ (duality);
3. $N(A) > 0$ implies $\Pi(A) = 1, \quad \Pi(A) < 1$ implies $N(A) = 0$.

In case of complete ignorance on A, $\Pi(A) = \Pi(\bar{A}) = 1$.

Possibility and Necessity of an Axiom with conjunctive development

If $u_\phi > 0$ and $D(\phi)$ is conjunctive,

$$\Pi(\phi) = 1 - \sqrt{1 - \left(\frac{u_\phi - u_\phi^-}{u_\phi}\right)^2}; \quad (1)$$

$$N(\phi) = \begin{cases} \sqrt{1 - \left(\frac{u_\phi - u_\phi^+}{u_\phi}\right)^2}, & \text{if } u_\phi^- = 0, \\ 0, & \text{if } u_\phi^- > 0; \end{cases} \quad (2)$$

Possibility and Necessity of an Axiom with disjunctive development

If $u_\phi > 0$ and $D(\phi)$ is disjunctive,

$$\Pi(\phi) = \begin{cases} 1 - \sqrt{1 - \left(\frac{u_\phi - u_\phi^-}{u_\phi}\right)^2}, & \text{if } u_\phi^+ = 0, \\ 1, & \text{if } u_\phi^+ > 0; \end{cases} \quad (3)$$

$$N(\phi) = \sqrt{1 - \left(\frac{u_\phi - u_\phi^+}{u_\phi}\right)^2}; \quad (4)$$

$$(5)$$

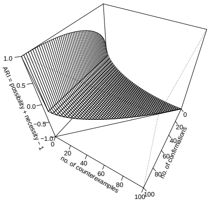
Acceptance/Rejection Index

Definition

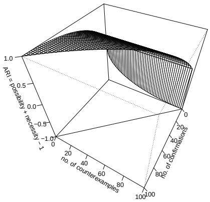
$$\mathrm{ARI}(\phi) = N(\phi) - N(\neg\phi) = N(\phi) + \Pi(\phi) - 1$$

- $-1 \leq \mathrm{ARI}(\phi) \leq 1$ for all axiom ϕ
- $\mathrm{ARI}(\phi) < 0$ suggests rejection of ϕ ($\Pi(\phi) < 1$)
- $\mathrm{ARI}(\phi) > 0$ suggests acceptance of ϕ ($N(\phi) > 0$)
- $\mathrm{ARI}(\phi) \approx 0$ reflects ignorance about the status of ϕ

CNF:



DNF:



Surrogate Model

Idea:

- if we can train a model to predict $\Pi(\phi)$ and $\Pi(\neg\phi)$
- then we have enough information to estimate $\mathrm{ARI}(\phi)$ henceforth
- without having to perform computations involving $\mathcal{K}$!

We can use a set of candidate axioms whose ARI is known to construct a training set consisting of axioms and their negations, labeled with their Π, which may be regarded formally as a label to be predicted through regression.

A Step Back—Why Do Surrogate Models Work?

- Can an agent guess if a formula is true or false?
    - given the current knowledge,
    - *in the current state of the world* (not necessarily in general)
- No reasoning!

$\Rightarrow$ Classification problem

- train a classifier against the knowledge base
- kernel-based representation of formulas
- model-theoretic semantic similarity measure

Problem Statement

Let Φ be a set of formulas in a logical language $\mathcal{L}$
Let $\mathcal{I}$ be an interpretation

$$\begin{bmatrix} \phi_1, & \phi_1^{\mathcal{I}} \\ \phi_2, & \phi_2^{\mathcal{I}} \\ \vdots & \vdots \end{bmatrix},$$

$\phi_i \in \Phi \subset \mathcal{L}$, for $i = 1, 2, \ldots$.

The open world hypothesis holds!

$\psi^{\mathcal{I}} = ?$ $\psi \notin \Phi$

Semantic Similarity

Definition (Language)

Let $\mathcal{A}$ be a *finite* set of atomic propositions and let $\mathcal{L}$ be the propositional language such that $\mathcal{A} \cup \{\top, \bot\} \subseteq \mathcal{L}$, and, $\forall \phi, \psi \in \mathcal{L}$, $\neg\phi \in \mathcal{L}$, $\phi \wedge \psi \in \mathcal{L}$, $\phi \vee \psi \in \mathcal{L}$.

Universe: $\Omega = \{0, 1\}^{\mathcal{A}}$

The semantics of a formula $\phi \in \mathcal{L}$ is the set of its models, $[\phi]$.

Semantic Similarity

Definition

$$\mathrm{sim}(\phi, \psi) = \frac{1}{\|\Omega\|} \sum_{\mathcal{I} \in \Omega} [\phi^{\mathcal{I}} = \psi^{\mathcal{I}}] = \frac{1}{\|\Omega_{\phi,\psi}\|} \sum_{\mathcal{I} \in \Omega_{\phi,\psi}} [\phi^{\mathcal{I}} = \psi^{\mathcal{I}}].$$

Properties:

1. $\mathrm{sim}(\phi, \phi) = 1$
2. $\mathrm{sim}(\phi, \neg\phi) = 0$
3. if $\mathcal{A}_\phi \cap \mathcal{A}_\psi = \emptyset$, $\mathrm{sim}(\phi, \psi) = \frac{1}{2}$
4. $\mathrm{sim}(\phi, \psi) = 1 - \mathrm{sim}(\neg\phi, \psi)$

Approximating Similarity

1. Randomly sample n interpretations from $\Omega_{\phi,\psi}$
2. Count for how many of them $\phi^{\mathcal{I}} = \psi^{\mathcal{I}}$.

$(1-\alpha)$ confidence interval for $\mathrm{sim}(\phi,\psi)$:

$$\hat{s}_{\phi,\psi} \pm z_{\alpha/2}\sqrt{\hat{s}_{\phi,\psi}(1-\hat{s}_{\phi,\psi})/n}, \qquad (6)$$

z_c is the $1-c$ quantile of $\mathcal{N}(0,1)$.

Example (99% confidence, worst case $\hat{s}_{\phi,\psi} = 0.5$)

$n = 30, \Rightarrow \hat{s}_{\phi,\psi} \pm 0.2351 \quad (= 2.576\sqrt{1/120})$
$n = 100, \Rightarrow \hat{s}_{\phi,\psi} \pm 0.1288$
$n = 1000 \Rightarrow \hat{s}_{\phi,\psi} \pm 0.0407$

A precise computation of the similarity between formulas is not really required for the proposed approach to work!

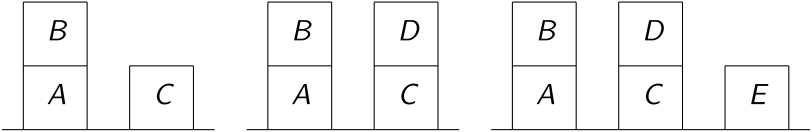
An Example from the Block World

Individual constants: A, B, C, $Table$
Unary predicate: covered$(\cdot)$,
Binary predicate on$(\cdot,\cdot)$.

$$\mathcal{A}_{12} = \{ \begin{array}{llll} \text{covered}(A), & \text{on}(A,B), & \text{on}(A,C), & \text{on}(A,Table), \\ \text{covered}(B), & \text{on}(B,A), & \text{on}(B,C), & \text{on}(B,Table), \\ \text{covered}(C), & \text{on}(C,A), & \text{on}(C,B), & \text{on}(C,Table) \end{array} \}.$$

$$\|\Omega_{12}\| = 2^{12} = 4,096$$
$$\|\Omega_{20}\| = 2^{20} = 1,048,576$$
$$\|\Omega_{30}\| = 2^{30} = 1,073,741,824$$

Reference Interpretations



$\mathcal{I}^*_{12}$ $\mathcal{I}^*_{20}$ $\mathcal{I}^*_{30}$

$$\mathcal{I}^*_{12} = \{\text{on}(A, \mathit{Table}), \text{on}(C, \mathit{Table}), \text{on}(B, A), \text{covered}(A)\}$$

etc.

Baseline Experiment

1. Generate 500 random formulas for each universe
2. Label each formula with its truth, based on the reference interpretation of its universe
3. Compute the similarity matrix (exact)

Truth	*Formula*	ϕ_0	ϕ_1	$\ldots$	ϕ_m
$Truth_0$	ϕ_0	1	$S_{0,1}$	$\ldots$	$S_{0,m}$
$Truth_1$	ϕ_1	$S_{1,0}$	1	$\ldots$	$S_{1,m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$Truth_{m-1}$	ϕ_{m-1}	$S_{m-1,0}$	$S_{m-1,1}$	$\ldots$	$S_{m-1,m}$
$Truth_m$	ϕ_m	$S_{m,0}$	$S_{m,1}$	$\ldots$	1

4. Train and test an SVM classifier (20-fold x-validation)

Generating Random Formulas

```
function RANDOM_FORMULA(𝒜, level)
                                  ▷ Recursive; the nesting level is initially = 0
    if RANDRANGE(level + 4) then              ▷ (level+3)/(level+4) probability of stopping
        return CHOICE(𝒜)
    end if
    if RANDRANGE(3) = 0 then                  ▷ with probability 1/3
        return ¬RANDOM_FORMULA(𝒜, level + 1)
    end if
    return RANDOM_FORMULA(𝒜, level + 1) · CHOICE(∧, ∨)·
        RANDOM_FORMULA(𝒜, level + 1)
end function
```

Sample training set made of 10 formulas from universe Ω_{12}

Formula	Label
$(\text{on}(C,B) \wedge \text{on}(B,C)) \vee (\neg\neg\neg\neg\text{on}(A,B) \vee \text{on}(B,A))$	True
$\neg(\neg(\text{on}(A,B) \wedge ((\text{on}(C,Tbl) \vee \neg\neg\text{on}(C,B)) \wedge \text{covered}(A))) \vee (\text{on}(C,A) \vee \text{on}(B,C)))$	False
$\neg(((\text{covered}(B) \vee \neg\text{on}(B,A)) \vee (\text{on}(B,Tbl) \wedge (\text{on}(B,A) \wedge \text{on}(C,A)))) \wedge \text{covered}(B))$	True
$\neg\neg\neg\text{on}(A,C)$	True
$(\text{on}(B,C) \vee \text{covered}(A)) \vee \text{covered}(A)$	True
$\neg(\neg\text{on}(A,C) \wedge \neg\text{covered}(C))$	False
$(\text{on}(A,C) \vee \text{on}(C,A)) \wedge \neg(\text{covered}(C) \wedge \text{on}(A,Tbl))$	False
$\text{on}(C,Tbl) \wedge \text{on}(C,B)$	False
$\text{on}(C,B) \vee \text{on}(B,C)$	False
$\neg\text{on}(C,B) \wedge (\neg\neg\neg\text{on}(C,B) \wedge \text{covered}(B))$	False

Sampling Experiment

1. We compute 3 new similarity matrices for each set of formulas, with $n = 30$, $n = 100$, and $n = 1000$
2. We study how the sample size n affects classification performance

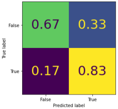
Results

Universe	Training set Size	sample size	Accuracy score	MCC
Ω_{12}	10	no sampling	0.77	0.56
		30	0.76	0.54
		100	0.77	0.56
		1000	0.77	0.55
Ω_{20}	20	no sampling	0.81	0.63
		30	0.81	0.62
		100	0.82	0.63
		1000	0.82	0.65
Ω_{30}	30	no sampling	0.83	0.66
		30	0.79	0.56
		100	0.82	0.62
		1000	0.83	0.64

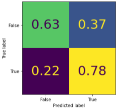
Sample of test set from universe Ω_{12}.

Formula	Actual	Predicted
$\neg(\neg(\neg(on(B,A) \wedge covered(B)) \vee ((covered(C) \vee on(A,)B) \wedge on(C,Tbl))) \vee \neg((on(B,A) \wedge on(A,Tbl)) \wedge \neg covered(A)))$	False	False
$(on(A,Tbl) \wedge on(B,A)) \vee \neg\neg((on(A,C) \wedge ((on(A,Tbl) \vee covered(A)) \wedge covered(A))) \vee on(C,B))$	True	True
$covered(B) \wedge on(A,Tbl)$	False	True
$((covered(B) \vee (\neg on(A,B) \wedge on(C,Tbl))) \wedge (((on(B,A) \wedge (on(A,C) \wedge on(C,Tbl))) \vee (on(C,A) \vee on(A,Tbl))) \wedge on(A,Tbl))) \vee on(C,A)$	True	True
$((covered(A) \wedge covered(B)) \vee (covered(A) \vee \neg(((covered(A) \vee on(B,C)) \vee (on(B,C) \wedge on(B,Tbl))) \vee on(C,A)))) \vee ((on(B,A) \vee on(A,C)) \wedge (((on(C,Tbl) \wedge on(B,C)) \wedge on(A,C)) \wedge covered(A)))$	True	True
$covered(B) \wedge (((on(C,B) \vee ((on(C,B) \wedge on(B,Tbl)) \wedge ((\neg covered(B) \vee (on(A,B) \wedge \neg on(A,C))) \vee (\neg\neg on(B,C) \wedge (covered(C) \vee on(A,B)))))) \vee covered(C)) \wedge (\neg(on(A,C) \vee on(B,C)) \wedge ((covered(A) \vee on(C,B)) \vee (on(A,Tbl) \wedge (on(A,Tbl) \vee on(C,B))))))$	False	False

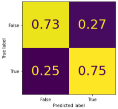
Confusion Matrices



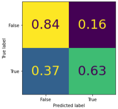
(a) Ω_{12}^{base}



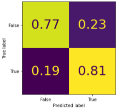
(b) Ω_{12}^{30}



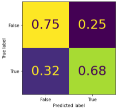
(c) Ω_{12}^{100}



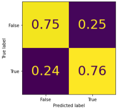
(d) Ω_{12}^{1000}



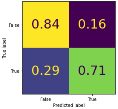
(e) Ω_{20}^{base}



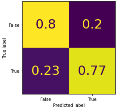
(f) Ω_{20}^{30}



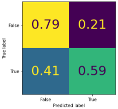
(g) Ω_{20}^{100}



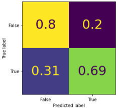
(h) Ω_{20}^{1000}



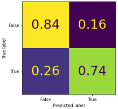
(i) Ω_{30}^{base}



(j) Ω_{30}^{30}



(k) Ω_{30}^{100}



(l) Ω_{30}^{1000}

Wrap-Up on Truth Guessing (i.e., Surrogate Models)

Framework to train a classification model that predicts the truth-value of a formula.

- Semantic similarity between formulas
- Tested an implementation of this framework using SVM
- Accuracy around 80%, even when similarity is approximated
- $\Rightarrow$ Approach is tractable!
- No guarantee that all the predictions made by a model be altogether consistent.
- When knowledge is incomplete, any prediction would be acceptable for some formulas

Now we can apply this idea to OWL axioms!

Axiom Similarity

Our kernel-based representation requires a kernel function which, for our purposes, is the similarity between two candidate axioms.

Based on the *ontological distance*

$$\forall (t_1, t_2) \in H^2,$$

$$D_H(t_1, t_2) = min_t(l_H(\langle t_1, t\rangle) + l_H(\langle t_2, t\rangle))$$

$$= min_t \left(\sum_{\{x \in <t_1, t>, x \neq t_1\}} 1/2^{d_H(x)} + \sum_{\{x \in <t_2, t>, x \neq t_2\}} 1/2^{d_H(x)} \right)$$

i.e., the minimum of the sum of the lengths of the subsumption paths between t_1 and t_2 and a common super type in hierarchy H.

Concept Similarity Matrix

Concepts	C_0	C_1	$\dots$	C_n
C_0	1	$S_{0,1}$	$\dots$	$S_{0,n}$
C_1	$S_{1,0}$	1	$\dots$	$S_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
C_{n-1}	$S_{n-1,0}$	$S_{n-1,1}$	$\dots$	$S_{n-1,n}$
C_n	$S_{n,0}$	$S_{n,1}$	$\dots$	1

From Concept to Axiom Similarity

1. Compare each axiom with all other axioms in the dataset.
2. When comparing two axioms, retrieve from the concept similarity matrix the similarity/distance between the concepts on the left side of the axiom.
3. Repeat the previous step for the right side.
4. In case of symmetric axiom types (disjointness/equivalence) repeat the comparison between the left concept from the first axiom and the right concept of the second axiom, and then between the right concept from the first axiom and the left concept from the second one. Keep the maximum of either.
5. Take the average of the two resulting values.
6. Store that value in the axiom similarity matrix (see next slide)

Axiom Similarity Matrix

Axioms	A_0	A_1	$\dots$	A_m
A_0	1	$S_{0,1}$	$\dots$	$S_{0,m}$
A_1	$S_{1,0}$	1	$\dots$	$S_{1,m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_{m-1}	$S_{m-1,0}$	$S_{m-1,1}$	$\dots$	$S_{m-1,m}$
A_m	$S_{m,0}$	$S_{m,1}$	$\dots$	1

Objectives

Our objective is to develop a scalable method to predict a score for atomic candidate OWL class axioms by learning from a set of previously scored axioms of the same type.

To this aim, we exploit the hierarchy of concepts formed by the subsumption rdfs:SubClassOf axioms.

To obtain scalability we perform feature selection.

N.B.: A separate model is required for each type of axiom addressed.

Overview of the Method

1. **Axiom extraction and scoring**: we create a set of scored axioms of a certain type.
2. **Axiom similarity calculation**, already explained.
3. **Kernel-based representation**: represent an axiom as the vector of similarities to a set of *base* axioms.
4. **Vector-space dimensionality reduction**, by feature selection.
5. **Candidate axiom encoding**: how a new candidate axiom is introduced into the vector-space, including the case where the axiom consists of concepts not available in the concept similarity matrix.
6. **Prediction**: we train a machine learning model with the dataset and predict the scores of new candidate axioms.

Axiom Extraction and Scoring

Two approaches:

1. Query $\mathcal{K}$ for existing axioms of one type (accepted) $+$ generate random axiom ϕ of same type s.t. $\mathcal{K} \not\models \phi$ (rejected)
2. Query $\mathcal{K}$ for existing subClassOf axioms (accepted for subClassOf and rejected fof disjointWith); then query $\mathcal{K}$ for existing disjointWith axioms (accepted for disjointWith and rejected fof subClassOf)

The latter approach looks more judicious.

Dimension Reduction

- We consider our dimensions (axioms) as features
- We apply a supervised filter-type feature selection method such as mutual information.
- This works by taking as input the axiom similarity matrix along with the scores of the axioms and returning a ranking of the dimensions from the most to the least impactful.
- We then keep z of these dimensions according to their ranks and discard the rest.
- The axiom similarity matrix becomes $m \times z$ (from $m \times m$)
- We also drop the unused concepts from the concept similarity matrix.

Benefits of Dimension Reduction

This reduction is beneficial in many ways:

- Reduction in the error rate due to the reduction in noise and redundancy.
- Reduction in the size of our vector-space and storage space for the axiom similarity matrix.
- Reduction in the size of our concept similarity matrix.
- Reduction in the computational complexity.
- A reduction in the look-up time when retrieving concept similarity.
- A better dataset for training our ML model.

Candidate Axiom Encoding

Two cases:

1. Candidate axiom comprises concepts already in the concept similarity matrix:
    - it goes straight through encoding, as we did with the training set.
2. Candidate axiom comprises concepts not found in the concept similarity matrix:
    - invoke a new similarity measure retrieval query;
    - this produces at most two new rows to be added to the concept similarity matrix;
    - then the candidate proceeds through encoding.

Prediction

- Now that we have our reduced vector space, we can apply any ML method.
- We chose to use k-NN to highlight the strength of our similarity measure.
- We compare it to more sophisticated methods such as random-forest regressors.
- This allows us to compare with previous literature.

Experimental Protocol

We use the following h/w configuration:

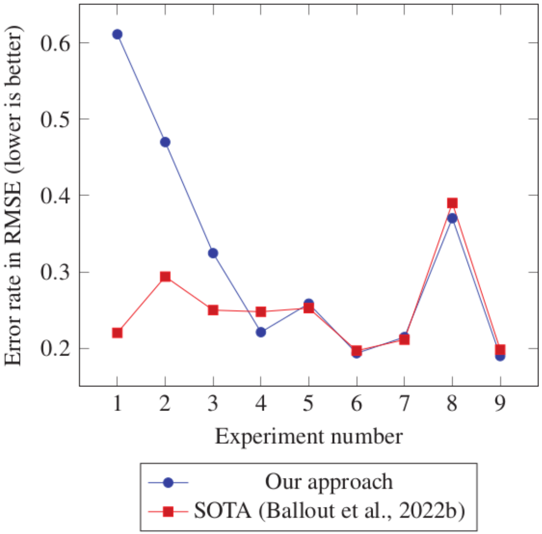
- CPU: Intel(R) Xeon(R) CPU W-11955M @ 2.60GHz base and 4.5 GHz all core boost. With 8 cores and 16 threads.
- A total of 128 GB of RAM memory with frequency 3200 MHZ.
- 1 TB of NVME SSD storage with read and write speeds of up to 2000 MB per second.

The code uses Python multiprocessing package.

We use the following ontologies of different sizes and domains:

- DBpedia, 762 concepts.
- Gene Ontology (GO), 29,575 concepts.
- Cell Ontology (CL), 78,835 concepts.

Results on DBpedia



Results on DBpedia (cont'd)

Comparison of computational cost in seconds as well as storage cost in number of values for CSM using using the DBpedia scored subClassOf dataset.

Method	# axioms processed	Initial CSM size	Concepts queried	Processing time	Encoding time
SOTA	722	580,644	762	13.72	0.019
Ours	722	**85,264**	**292**	**3.86**	**0.005**

Results on GO and CL

Comparison of computational cost in seconds as well as storage cost in MB for ASM and number of values for CSM using the GO disjointWith dataset. Time out error: TO.

Method	# axioms processed	ASM size	Initial CSM size	Concepts queried	Processing time	Encoding time
SOTA	600	62,000	6,214,957,225	78,835	TO	TO
Ours	600	**5.1**	**95,481**	**309**	**312.54**	**0.034**

Comparison of computational cost in seconds as well as storage cost in MB for ASM and number of values for CSM using the CL disjointWith dataset. Time out error: TO.

Method	# axioms processed	ASM size	Initial CSM size	Concepts queried	Processing time	Encoding time
SOTA	600	8,000	874,680,625	29,575	TO	TO
Ours	600	**2.05**	**216,225**	**465**	**103.7**	**0.015**

Conclusions and Future Work

Scalable method for the score prediction of atomic candidate *OWL* class axioms of different types. It relies on:

- an ontological semantic similarity measure
- feature selection for vector-space dimension reduction

Results support the effectiveness of the proposed method:

- scalable, consistent and stable w/ ontologies of different sizes
- suited to deal wuth large real-world ontologies.

Some research paths emerge, including:

- Coping with complex candidate axioms.
- Incorporating active learning.