Optimizing the Computation of a Possibilistic Heuristic to Test OWL SubClassOf Axioms Against RDF Data

Evolutionary Axiom Discovery from Knowledge Graphs

Rémi FELIN; Olivier CORBY; Catherine FARON; Andrea G. B. TETTAMANZI

Université Côte d'Azur, Inria, CNRS, I3S
WIMMICS Team

- A research area focused on **Ontology Enrichment** of the Semantic Web.
- An evolutionary approach for OWL axiom extraction based on **grammatical evolution** combined with **possibility theory**.
- We use W3C standards such as **RDF**, **OWL** and **SPARQL** to extract meaningful **axioms**.

- A research area focused on **Ontology Enrichment** of the Semantic Web.
- An evolutionary approach for OWL axiom extraction based on **grammatical evolution** combined with **possibility theory**.
- We use W3C standards such as **RDF**, **OWL** and **SPARQL** to extract meaningful **axioms**.

- A research area focused on **Ontology Enrichment** of the Semantic Web.
- An evolutionary approach for OWL axiom extraction based on **grammatical evolution** combined with **possibility theory**.
- We use W3C standards such as **RDF**, **OWL** and **SPARQL** to extract meaningful **axioms**.

- We focus on **2 research questions**:
  - *How to automatically discover new OWL axioms over a knowledge graph ?*
  - *How to assess OWL axioms against a given RDF graph ?*
- We focus on the assessment of OWL Subsumption axioms.
  - `SubClassOf(<C> <D>)` where C and D are concepts from a given ontology.
  - $C \sqsubseteq D$ highlight the perfect inclusion of the instances of C within the instances of D, i.e. the instances of C are also instances of D
  - verify the compatibility of a given subsumption axiom with the known facts: a possibilistic heuristic to test it!

- We focus on **2 research questions**:
  - *How to automatically discover new OWL axioms over a knowledge graph ?*
  - *How to assess OWL axioms against a given RDF graph ?*
- We focus on the assessment of OWL Subsumption axioms.
  - `SubClassOf(<C> <D>)` where C and D are concepts from a given ontology.
  - $C \sqsubseteq D$ highlight the perfect inclusion of the instances of C within the instances of D, i.e. the instances of C are also instances of D
  - verify the compatibility of a given subsumption axiom with the known facts: a possibilistic heuristic to test it!

- We focus on **2 research questions**:
  - *How to automatically discover new OWL axioms over a knowledge graph ?*
  - *How to assess OWL axioms against a given RDF graph ?*
- We focus on the assessment of OWL Subsumption axioms.
  - `SubClassOf(<C> <D>)` where `C` and `D` are concepts from a given ontology.
  - $C \sqsubseteq D$ highlight the perfect inclusion of the instances of C within the instances of D, i.e. the instances of C are also instances of D
  - verify the compatibility of a given subsumption axiom with the known facts: a possibilistic heuristic to test it!

Possibility theory

Possibility theory is a mathematical theory of epistemic uncertainty which considers events denoted ω of a universe of discourse Ω ($\omega \in \Omega$) where each ω has a degree of possibility $\pi(\omega) \in [0,1]$. The theory includes a measure of possibility denoted by Π and a measure of necessity denoted by N such that:

$$\Pi(A) = \max_{\omega \in A} \pi(\omega),$$
$$N(A) = 1 - \Pi(\bar{A}) = \min_{\omega \in \bar{A}}\{1 - \pi(\omega)\},$$

where $A \sqsubseteq \Omega$ or $A = \{\omega : \omega \models \phi\}$.

Possibility theory applied on subsumption axioms evaluation [1]

Let us consider v_ϕ^+ the *confirmations* and v_ϕ^- the *exceptions* observed among the elements of v_ϕ, the **support** for an axiom ϕ. We define the **possibility** $\Pi(\phi)$ and **necessity** $N(\phi)$ of an axiom as follows:

$$\Pi(\phi) = 1 - \sqrt{1 - \left(\frac{v_\phi - v_\phi^-}{v_\phi}\right)^2}, N(\phi) = \begin{cases} \sqrt{1 - \left(\frac{v_\phi - v_\phi^+}{v_\phi}\right)^2}, & if \, \Pi(\phi) = 1 \\ 0 & otherwise \end{cases}$$

[1] Tettamanzi, A., Faron Zucker, C., and Gandon, F. - International Journal of Approximate Reasoning (2017).

ARI (Acceptance/Rejection Index) [2]

In order to decide about a given axiom ϕ, we define an ARI value using $\Pi(\phi)$ and $N(\phi)$:

$$\mathrm{ARI}(\phi) = \mathrm{N}(\phi) + \Pi(\phi) - 1 \in [-1,1].$$

[2] Tettamanzi, A., Faron Zucker, C., and Gandon, F. - International Journal of Approximate Reasoning (2017).

SPARQL queries are built to extract the **number of instances** implied by a given axiom (v_ϕ), the **number of confirmations** (v_ϕ^+) and the **number of exceptions** (v_ϕ^-).

Let's consider the given axiom: SubClassOf(<C> <D>)

SPARQL Query to extract v_ϕ

```
SELECT (count(distinct ?x) as ?u_phi) WHERE {
    ?x a <C> .
}
```

SPARQL queries are built to extract the **number of instances** implied by a given axiom (v_ϕ), the **number of confirmations** (v_ϕ^+) and the **number of exceptions** (v_ϕ^-).

Let's consider the given axiom: `SubClassOf(<C> <D>)`

SPARQL Query to extract v_ϕ

```
SELECT (count(distinct ?x) as ?u_phi) WHERE {
       ?x a <C> .
}
```

SPARQL Query to extract υ_{ϕ}^{+}

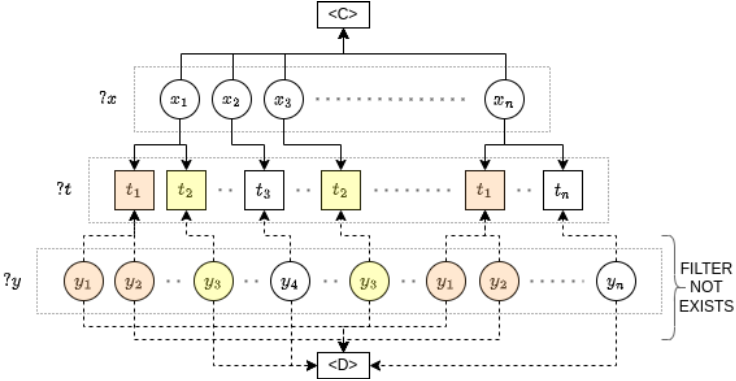
```
SELECT (count(distinct ?x) as ?conf) WHERE {
    ?x a <C> , <D> .
}
```

SPARQL Query to extract υ_{ϕ}^{-}: a first heuristic

```
SELECT (count(distinct ?x) AS ?exc) WHERE {
    ?x a <C> , ?T .
    FILTER NOT EXISTS {
        ?y a ?T , <D> .
    }
}
```

- a second heuristic: adding a **time cap** for the computation of v_{ϕ}^{-}: [3]
  - if the limit is reached, a quick but approximate computation of exceptions is adopted.

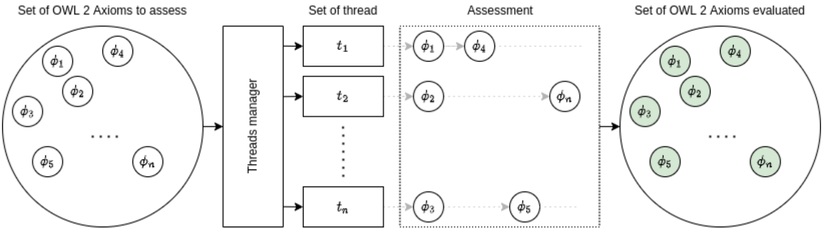
[3] Tettamanzi, A., Faron Zucker, C., and Gandon, F. - K-CAP (2015)


<C>
?x
x_1
x_2
x_3
x_n
?t
t_1
t_2
t_3
t_2
t_1
t_n
?y
y_1
y_2
y_3
y_4
y_3
y_1
y_2
y_n
FILTER NOT EXISTS
<D>

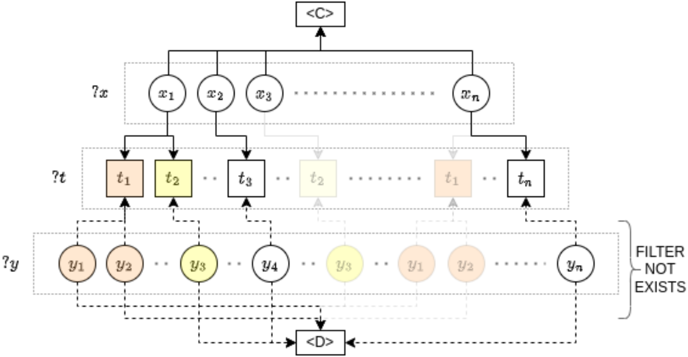
dbo concept	# instances
dbo:Agent	1,472,369
dbo:Person	1,124,388
dbo:Place	754,415
dbo:CareerStation	577,196
dbo:PopulatedPlace	531,228
dbo:Settlement	466,791
dbo:Work	409,594
dbo:Organisation	329,500
dbo:Athlete	313,730

```
SELECT ?class (count(distinct ?x) as ?n)
WHERE {
    ?x a ?class .
    FILTER(
        STRSTARTS(
            STR(?class),
            "http://dbpedia.org/ontology/"
        )
    )
}
GROUP BY ?class
ORDER BY DESC(?n)
LIMIT 10
```

A **multi-threading system** for the assessment of OWL axioms has been implemented.



- **Aim:** avoid the computation of types t already involved.



Recovering distinct types (i.e., classes) being evaluated as potentially containing exceptions.

Step 1: Recovering distinct types

```
SELECT distinct ?t WHERE {
    {
        SELECT ?t WHERE {
            {
                SELECT distinct ?t WHERE {
                    ?x a <C> , ?t .
                } ORDER BY ?t
            }
        } LIMIT $limit OFFSET $offset
    }
    FILTER NOT EXISTS {
        ?y a ?t , <D> .
    }
}
```

Retrieving the instances that belong to <C> and at least one of the types <t> just retrieved, which will count as exceptions.

Step 2: Retrieving the instances

```
SELECT distinct ?x WHERE {
    ?x a <C> .
    ?x a ?t values (?t) {
        (<t1>) (<t2>) ... (<tn>)
    }
} LIMIT $limit OFFSET $offset
```

- Adding a new operator as a URI parameter in federated query services.[4]
- Allows you to automatically iterate a service call.
- Implemented in the *Corese* semantic web factory.

[4] Corby, O., Faron, C., Gandon, F., Graux, D., and Michel, F - WEBIST (2021)

- Sub-processing of the pagination: more efficient and less code writing.

SPARQL Query of step.1

```
SELECT distinct ?t WHERE {
    {
        SELECT ?t WHERE {
            {
                SELECT distinct ?t WHERE {
                    ?x a <C> , ?t .
                } ORDER BY ?t
            }
        } LIMIT $limit OFFSET $offset
    }
    FILTER NOT EXISTS {
        ?z a ?t , <D> .
    }
}
```

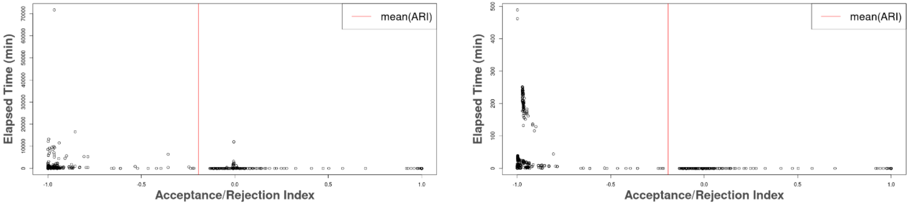
Revised SPARQL Query using loop

```
SELECT distinct ?t WHERE {
    SERVICE <$url/sparql?loop=true&limit=$limit> {
        SELECT distinct ?t WHERE {
            ?x a <C> , ?t .
        }
    }
    SERVICE <$url/sparql> {
        values ?t {undef}
        FILTER NOT EXISTS {
            ?z a <D> , ?t .
        }
    }
}
```

1. Benchmarking our approach with previous results.[5]
    - a set of 722 subsumption axioms to assess.
    - checking the equality of the ARIs between both approaches.
    - comparison of the computation times (CPU) for each axiom.
2. Assessing the usage of the `loop` operator.
    - *what benefit do we get from using this operator ?*
    - comparison of the computation times (CPU) for each axiom assessed in Experiment 1.

[5] Tettamanzi, A., Faron Zucker, C., and Gandon, F. - K-CAP (2015)

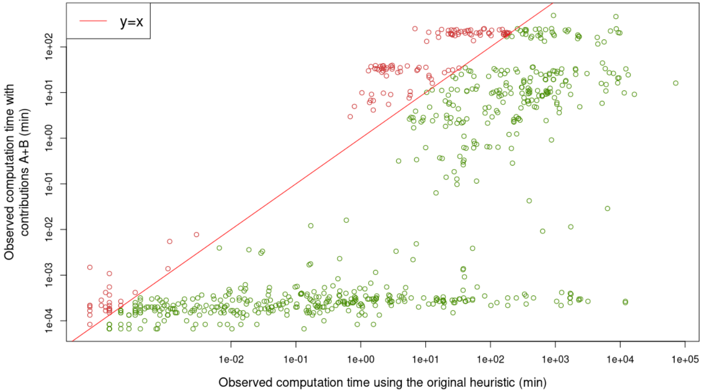
- The integrity of the ARI values is respected
  - $mean(ARI) \simeq$ -0.1936 in both settings.
- A significant reduction of the elapsed time
  - mean time spent to assess axioms ...
    - with the original heuristic: 577.9 minutes
    - with contributions A+B: **30.14** minutes
  - maximum computation time goes down from 71,699 to 489 minutes.



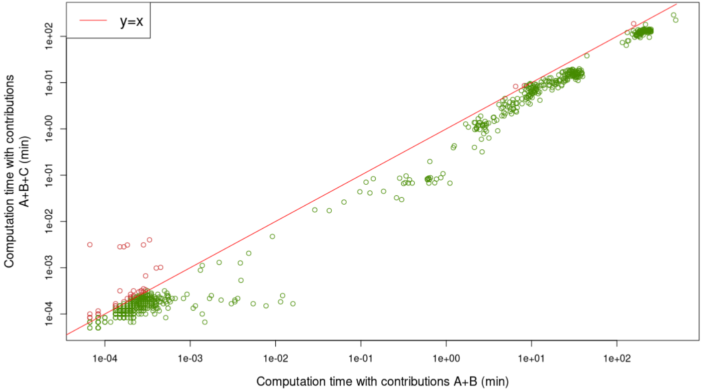
(a) Original heuristic

(b) Results obtained with contributions A+B

- 593 axioms (82%, in green) are faster to assess with contrib. A+B
  - average computation time saved: $\sim$ **679** minutes.
- 129 axioms (18%, in red) are slower to assess with contrib. A+B
  - average computation time increase: 57 minutes.
  - maximum loss: 244 minutes.



- a reduction of $\sim$ 12 minutes of the average computation time.
- 683 axioms (95%, in green) are faster to assess with contrib. A+B+C.
- 39 axioms (5%, in red) are slower to assess with contrib. A+B+C.



- A significant reduction of computation time (CPU) without any side effects.
- Makes it possible to lead huge experiments on OWL subsumption axioms mining in a reasonable time, without a time cap.
- Opens up new perspectives for the processing of the largest knowledge graphs in the Linked Open Data.

- A significant reduction of computation time (CPU) without any side effects.
- Makes it possible to lead huge experiments on OWL subsumption axioms mining in a reasonable time, without a time cap.
- Opens up new perspectives for the processing of the largest knowledge graphs in the Linked Open Data.

- A significant reduction of computation time (CPU) without any side effects.
- Makes it possible to lead huge experiments on OWL subsumption axioms mining in a reasonable time, without a time cap.
- Opens up new perspectives for the processing of the largest knowledge graphs in the Linked Open Data.

Thank you !