Using Grammar-based Genetic Programming for Mining Disjointness Axioms Involving Complex Class Expressions

Thu Huong Nguyen   Andrea G. B. Tettamanzi

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

Symposium MaDICS 2020, July 6, 2020

Motivation

- Linked Open Data (LOD) has made a huge number of interconnected RDF triples freely available for sharing and reuse
- Shared schemas and ontologies are needed to support reasoning
- Manual acquisition of axioms:
    - is exceedingly expensive and time-consuming
    - depends on the availability of domain specialists and knowledge engineers

Introduction: Ontology Learning

Top-down construction of ontologies has limitations

- aprioristic and dogmatic
- does not scale well
- does not lend itself to a collaborative effort

Bottom-up, *grass-roots* approach to ontology and KB creation

- start from RDF facts and learn OWL 2 axioms

Recent contributions towards OWL 2 ontology learning

- FOIL-like algorithms for learning concept definitions
- statistical schema induction via association rule mining
- light-weight schema enrichment (DL-Learner framework)

All these methods apply and extend ILP techniques.

Class Disjointness Axiom Learning

Problem:

We focus on learning OWL class disjointness axioms involving existential quantification ($\exists r.C$) and value restriction ($\forall r.C$)

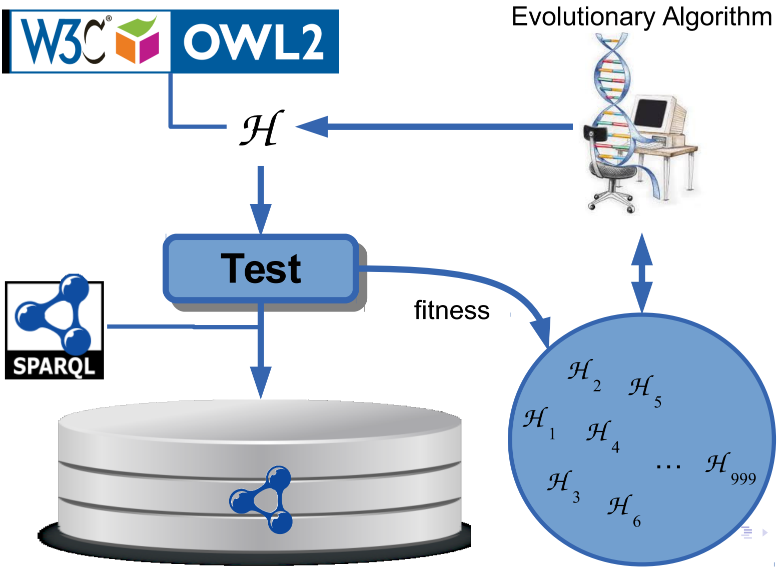
Class disjointness axioms are important

- to check the correctness of a knowledge base
- to derive new knowledge

Method:

Grammar-Based Genetic Programming

Synopsis


W3C
OWL2
Evolutionary Algorithm
$\mathcal{H}$
Test
fitness
SPARQL
$\mathcal{H}_2$ $\mathcal{H}_5$
$\mathcal{H}_1$ $\mathcal{H}_4$
... $\mathcal{H}_{999}$
$\mathcal{H}_3$ $\mathcal{H}_6$

Grammatical Evolution

- A grammar-based form of Genetic Programming
- Search space distinguished from solution (= program) space
- Search space consists of variable-length bitstrings
- Bitstrings are mapped into programs thanks to a BNF *grammar*

In the mapping process, codons are used consecutively to choose production rules in the BNF grammar according to the function:

$$\textit{production} = \textit{codon}\ \mathbf{modulo} \left[\text{Number of productions for the current non-terminal}\right]$$

Static Part of the Grammar

Axiom := ClassAxiom

ClassAxiom := DisjointClasses

DisjointClasses := 'DisjointClasses' '(' ClassExpression1 ' 'ClassExpression2 ')'

ClassExpression1 := Class
| ObjectSomeValuesFrom
| ObjectAllValuesFrom
| ObjectIntersection

ClassExpression2 := ObjectSomeValuesFrom
| ObjectAllValuesFrom

ObjectIntersectionOf := 'ObjectIntersectionOf' '(' Class ' ' Class ')'

ObjectSomeValuesFrom := 'ObjectSomeValuesFrom' '(' ObjectPropertyOf ' ' Class ')'

ObjectAllValuesFrom := 'ObjectAllValuesFrom' '(' ObjectPropertyOf ' ' Class ')'

Dynamic Part of the Grammar

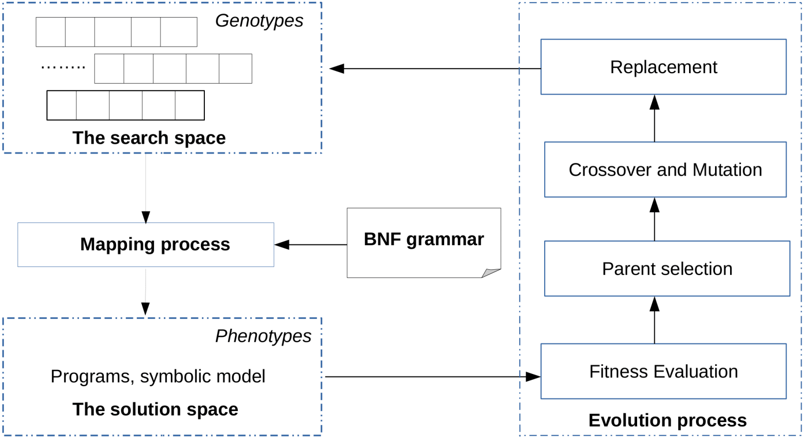
Class := ...

```
SELECT ?class
WHERE { ?instance rdf:type ?class . }
```

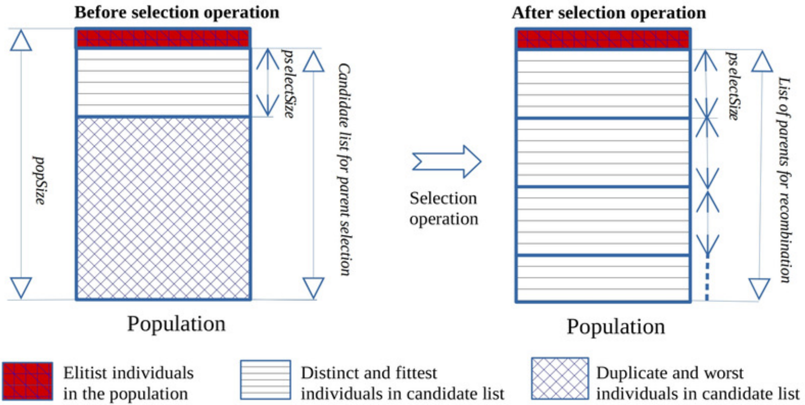
ObjectPropertyOf := ...

```
SELECT ?property
WHERE {
  ?subject ?property ?object .
  FILTER (isIRI(?object))
}
```

Grammatical Evolution Process



Parent Selection



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, $\Pi(A) = \Pi(\bar{A}) = 1$.

Content of an Axiom

Definition (Content of Axiom ϕ)

Given an RDF datset $\mathcal{K}$,

$$\mathrm{content}(\phi) = \{\psi : \phi \models_{\mathcal{K}} \psi\}$$

obtained through the instantiation of ψ to the vocabulary of $\mathcal{K}$.

Let $\phi = \mathsf{Dis}(C, D)$

$$\mathrm{content}(\phi) = \{\neg C(r) \vee \neg D(r) : \quad r \text{ is a resource in } \mathcal{K}\}$$

Confirmation and Counterexample of an Axiom

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

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

Definition

Given axiom ϕ, let us define

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

Axiom Evaluation

Definition (Generality)

$g_\phi = \min\{\|[C]\|, \|[D]\|\}$, where C, D are class expressions.

Definition (Possibility)

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

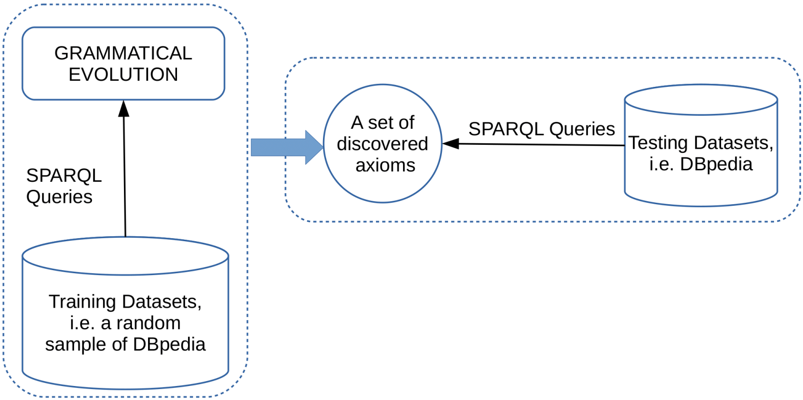
Definition (Fitness)

$$f(\phi) = g_\phi \cdot \Pi(\phi)$$

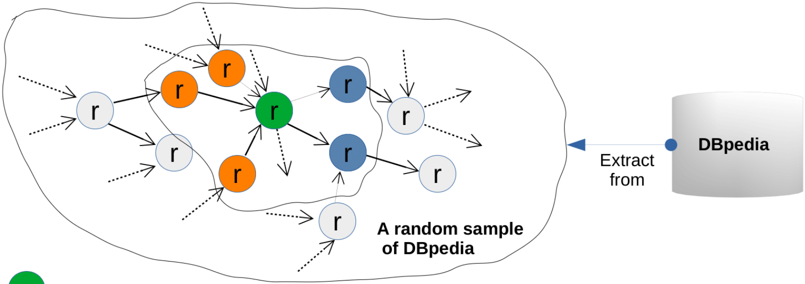
Experimental Protocol

- Training-Testing Model
- Experiments are divided into two phases:
  1. mining class disjointness axioms with GE from a training RDF dataset, i.e., a 1% random sample of DBpedia 2015-04
  2. testing the resulting axioms against the test dataset, i.e., the entire DBpedia 2015-04
- An objective benchmark to evaluate the effectiveness of the method.

Training-Testing Model



Training Set Construction



r Initial resource, e.g. *http://dbpedia.org/ontology/Plant*

r Resource as a subject in the triples relevant to r, i.e r *is the object of the extracted triple*

r Resource as an object in the triples relevant to r, i.e r *is the subject of the extracted triple*

Relation between resources as the predicate in the triples

Experimental Setup

- 20 different runs on different parameter settings
- To allow fair comparisons, we define total effort

$$k = \text{total number of fitness evaluations}$$

- *maxGenerations* is set so that

$$popSize \cdot maxGenerations = k$$

Parameter	**Value**
Total effort k	100,000; 200,000; 300,000; 400,000
initLenChrom	6
pCross	80%
pMut	1%
popSize	1000; 2000; 5000; 10000

Measuring Accuracy

Since $\Pi(\phi)$ may be viewed as a fuzzy degree of membership, we use a fuzzy extension of the usual definition of *precision*, based on fuzzy set cardinality

$$\|F\| = \sum_{x \in \Delta} F(x),$$

The value of precision can thus be computed against the test dataset, i.e., DBpedia 2015-04, according to the formula

$$\text{precision} = \frac{\|\text{true positives}\|}{\|\text{discovered axioms}\|} = \frac{\sum_{\phi} \Pi_{\mathrm{DBpedia}}(\phi)}{\sum_{\phi} \Pi_{\mathrm{Training}}(\phi)}$$

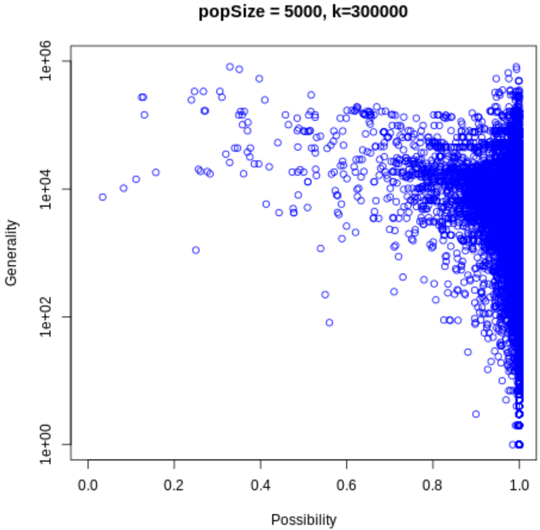
Axioms Discovered Over 20 Runs

k \ popSize	1000	2000	5000	10000
100000	8806	11389	4684	4788
200000	6204	13670	10632	9335
300000	5436	10541	53021	14590
400000	5085	9080	35102	21670

Average Precision per Run (±**std**)

k \ *popSize*	1,000	2,000	5,000	10,000
100,000	0.981 ±0.019	0.999 ±0.002	0.998 ±0.002	0.998 ±0.003
200,000	0.973 ±0.024	0.979 ±0.011	0.998 ±0.001	0.998 ±0.002
300,000	0.972 ±0.024	0.973 ±0.014	0.993 ±0.007	0.998 ±0.001
400,000	0.972 ±0.024	0.969 ±0.018	0.980 ±0.008	0.998 ±0.001

Π and g Distribution of Discovered Axioms



Examples of Discovered Axioms

Dis(∀author.Place, ∀placeofBurial.Place)	$\Pi(\phi) = 1.0;\ g_\phi = 4$
Dis(Writer, ∀writer.Agent)	$\Pi(\phi) = 0.982;\ g_\phi = 79,464$
Dis(Journalist, ∀distributor.Agent)	$\Pi(\phi) = 0.992;\ g_\phi = 32,533$
Dis(Stadium, ∀birthPlace.Place)	$\Pi(\phi) = 1.0;\ g_\phi = 10,245$

Conclusions and Future Work

- Grammar-based GP method for mining disjointness axioms involving complex class expressions
- The use of a training-testing model allows to objectively validate the method, while also alleviating the computational bottleneck of SPARQL endpoints
- The experimental results confirm that the proposed method is capable of discovering highly accurate and general axioms
- Future Work:
  - Mining disjointness axioms involving operators such as `owl:hasValue` and `owl:OneOf`
  - Forbid atomic classes at the root of class expressions
  - Refining the evaluation of candidate axioms with some measure of their complexity

The End

Thank you for your attention!