Evolutionary knowledge discovery from RDF data graphs

Rémi Felin

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Découverte évolutive de connaissance à partir de graphes de données RDF

Rémi FELIN

Laboratoire d'Informatique, de Signaux et Systèmes de Sophia Antipolis (I3S)
UMR7271 UCA CNRS

Présentée en vue de l'obtention du grade de docteur en Informatique **d'**Université Côte d'Azur

Dirigée par : Andrea G. B. TETTAMANZI, Professeur, Université Côte d'Azur
Co-dirigée par : Catherine FARON, Professeure, Université Côte d'Azur
Soutenue le : 22 Novembre 2024

Devant le jury, composé de :
Fabien GANDON, Directeur de recherche, Inria
Fatiha SAIS, Professeure, Université Paris Saclay
Michael O'NEILL, Professeur, University College Dublin
Olivier CURÉ, Professeur, Université Gustave Eiffel

Découverte évolutive de connaissance à partir de graphes de données RDF

Evolutionary Knowledge Discovery From RDF Data Graphs

Rémi Felin

Jury :

Président du jury

Fabien Gandon, Directeur de recherche, Inria

Rapporteurs

Fatiha Sais, Professeure, Université Paris Saclay
Michael O'Neill, Professeur, University College Dublin

Examinateurs

Olivier Curé, Professeur, Université Gustave Eiffel

Directeur de thèse

Andrea G. B. Tettamanzi, Professeur, Université Côte d'Azur

Co-directeur de thèse

Catherine Faron, Professeure, Université Côte d'Azur

Université Côte d'Azur

Rémi Felin

Découverte évolutive de connaissance à partir de graphes de données RDF

xv+142 p.

à ma grand-mère, Ghislain

Découverte évolutive de connaissance à partir de graphes de données RDF

Résumé

Les graphes de connaissance sont des collections de descriptions interconnectées d'entités (objets, événements ou concepts). Ils mettent les données en contexte par le biais de liens sémantiques, fournissant ainsi un cadre pour l'intégration, l'unification, l'analyse et le partage des données. Aujourd'hui, nous disposons d'un grand nombre de graphes de connaissance riches en données factuelles, dont la construction et l'enrichissement est une tâche relativement bien maîtrisée. Ce qui est plus difficile et plus coûteux, c'est de doter ces graphes de schémas, règles et contraintes qui permettent de vérifier leur cohérence et de déduire des connaissances implicites par raisonnement. Cette thèse présente une approche basée sur la technique d'évolution grammaticale pour la découverte automatique de nouvelles connaissances à partir d'un graphe de données représenté en RDF. Cette approche repose sur l'idée que les connaissances candidates sont générées à partir d'un mécanisme heuristique (exploitant les données du graphe), testés contre les données du graphe, et évoluent à travers un processus évolutionnaire de sorte à ce que seules les connaissances candidates les plus crédibles soient conservées. Dans un premier temps, nous nous sommes concentrés sur la découverte d'axiomes OWL qui permettent, par exemple, d'exprimer des relations entre concepts et d'inférer, à partir de ces relations, de nouvelles informations factuelles. Les axiomes candidats sont évalués à partir d'une heuristique existante basée sur la théorie des possibilités, permettant de considérer l'incomplétude des informations d'un graphe de données. Cette thèse présente les limites de cette heuristique et une série de contributions permettant une évaluation significativement moins coûteuse en temps de calcul. Cela a permis l'évaluation efficace d'axiomes candidats lors du processus évolutif, nous menant ainsi à la découverte d'un grand nombre d'axiomes candidats pertinents vis-à-vis d'un graphe de données RDF. Dans un second temps, nous avons proposé une approche pour la découverte de shapes SHACL qui expriment des contraintes que les données RDF doivent respecter. Elles sont utiles pour contrôler la cohérence (par exemple, structurelle) des données du graphe et facilitent l'intégration de nouvelles données. L'évaluation de shapes candidates repose sur l'évaluation SHACL des données vis-à-vis de ces formes, à laquelle nous ajoutons un cadre probabiliste pour prendre en compte les erreurs et l'incomplétude inhérente des graphes de données lors de l'évaluation de shapes candidates. Enfin, nous présentons `RDFminer`, une application Web open-source permettant d'exécuter notre approche pour découvrir des axiomes OWL ou des formes SHACL à partir d'un graphe de données RDF. L'utilisateur peut contrôler l'exécution et analyser les résultats en temps réels à travers une interface graphique interactive. Les résultats obtenus montrent que l'approche proposée permet de découvrir un large ensemble de nouvelles connaissances crédibles et pertinentes à partir de graphes de données RDF volumineux.

Mots-clés : Évolution grammaticale, Extraction de forme SHACL, Apprentissage d'ontologies, OWL, SHACL, SPARQL

Evolutionary Knowledge Discovery From RDF Data Graphs

Abstract

Knowledge graphs are collections of interconnected descriptions of entities (objects, events or concepts). They provide context for the data through semantic links, providing a framework for integrating, unifying, analysing and sharing data. Today, we have many factual data-rich knowledge graphs, and building and enriching them is relatively straightforward. Enriching these graphs with schemas, rules or constraints that allow us to check their consistency and infer implicit knowledge by reasoning is more difficult and costly. This thesis presents an approach based on the Grammatical Evolution technique for automatically discovering new knowledge from the factual data of a data graph expressed in RDF. This approach is based on the idea that candidate knowledge is generated from a heuristic mechanism (exploiting the graph data), is tested against the graph data, and evolves through an evolutionary process so that only the most credible candidate knowledge is kept. First, we focused on discovering OWL axioms that allow, for example, the expression of relationships between concepts and the inference of new facts previously unknown from these relationships. Candidate axioms are evaluated using an existing heuristic based on possibility theory, which makes it possible to consider the incompleteness of information in a data graph. This thesis presents the limitations of this heuristic and a series of contributions allowing an evaluation that is significantly less costly in computation time, thus opening up the discovery of candidate axioms using this heuristic. Second, we propose discovering SHACL shapes that express constraints that RDF data must respect. These shapes are useful for checking the data graph's consistency (*e.g.*, structural) and facilitating new data integration. The evaluation of candidate shapes is based on the SHACL evaluation mechanism, for which we proposed a probabilistic framework to take into account errors and the inherent incompleteness of the data graphs. Finally, we present `RDFminer`, an open-source Web application that executes our approach to discovering OWL axioms or SHACL shapes from an RDF data graph. Through an interactive interface, the user can also control the execution and analyse the results in real-time. The results show that the proposed approach can be used to discover a wide range of new, credible and relevant knowledge from large RDF data graphs.

Keywords: Grammatical Evolution, Shape Mining, Ontology Learning, OWL, SHACL, SPARQL

Acknowledgement

During these three years of thesis, I feel extremely lucky and honoured to have met and worked with people whom I would like to thank in this section. The same concerns the people around me who have supported me during this period.

I cannot begin this section without expressing my sincere thanks to my supervisors: Andrea Tettamanzi and Catherine Faron. For the trust they placed in me, their invaluable advice and their constant presence, without which this thesis would never have been possible. They helped me to grow as a human being and taught me the rudiments of research. I naturally dedicate the results of this work to them.

I would like to thank my thesis jury: Fatiha Saïs, Michael O'Neill, Olivier Curé and Fabien Gandon. For taking the time to examine my work and for their invaluable feedback, which has helped me to improve this work and its perspectives.

I would like to extend my warmest thanks to the members of my Personal Monitoring Committee: Nathalie Hernandez and Fabien Gandon, who have followed the annual progress of my work and given me their expert advice.

My thoughts are with the people who helped me with my publications (co-authors): Olivier Corby, for his precious help and expertise in producing optimal SPARQL queries and significant results. Pierre Monnin for his kindness, expertise and extremely valuable advice, in producing a high-performance evolutionary algorithm of crucial importance to this thesis.

Naturally, I'd like to express my deepest thanks to the members of the Wimmics team: "*Wimmicians*". I feel honored and lucky to have shared these three years with you. I've grown professionally and as a person. I'm grateful for all the moments I've shared with you during seminars, meetings and coffee breaks. I have a special thought for those who have helped me throughout this journey. I'm thinking, for example, of Olivier Corby and Rémi Cérès, for their invaluable help with the Corese software. I would also like to thank the SPARKS team, I3S laboratory and Inria for giving me an ideal working environment.

I would like to thank the people who have helped me along the way: Caroline Poirier, who enabled me to change career direction and do a research placement (before my thesis), this would not have been possible without her devotion and I am extremely grateful.

In the same way, I would like to thank Lionel Tavanti for his assistance throughout my thesis, particularly in the organization of my conference trips.

I'd like to extend my deepest thanks to my friends for their unconditional support and the times we've shared together, which have kept me on the right path throughout my journey. A special thought to Benjamin Molinet, who motivated me to undertake this extraordinary adventure of research, and his wise and invaluable advice. I feel extremely lucky and grateful for their kindness, I also dedicate the success of this work to them.

I have a deep thought for Maroua Tikat, for her presence and unconditional support, which enabled me to never feel lonely, especially at the end of my journey. Thank you is not enough to describe what you have done and what a wonderful person you are.

To conclude these few words, I want to address my family: my little sister, my parents and grand-parents, for their unconditional love, their presence and their constant support in all my life and professional projects. I feel extremely grateful and I want to dedicate my academic journey, this thesis work, the results that arise from it, to them, and I hope to make them proud.

Contents

Chapter 1

Introduction

1.1 Context

In recent years, we have observed a significant and continuous increase in data over the Web. This trend results from multiple and varied initiatives aimed at exploiting data by humans and artificial agents by creating methods for structuring them through various Web standards.

In this context, facts are grouped and contextualized (through semantic links and metadata) within **knowledge graphs** (KGs). More specifically, KGs are collections of facts and observations from a *universe of discourse* structured using the **RDF** (Resource Description Framework) data model. Semantic Web standards include various languages: *e.g.*, **Web Ontology Language** (OWL), and **SPARQL Protocol and RDF Query Language** (SPARQL), which enable the *sharing*, and *querying* of structured data on the Web. We distinguish two essential components in constructing these KGs:

- **Ontologies** formally represent a set of concepts within a domain and the relationships between those concepts. They describe the universe of discourse by defining the types, properties, and interrelationships of the existing resources in the domain. Ontologies are vocabularies that enable both humans and artificial agents to comprehend the *structure* and the *meaning* of data: *e.g.*, the *FOAF* (Friend Of A Friend) ontology describes people, their activities, and their relations to other people. **OWL** is the semantic Web language used for writing ontologies.
- **Data graphs** are collections of facts represented in RDF. They represent the fact instances within the structured context provided by the ontology. These facts are depicted as *nodes* (representing entities) and *edges* (representing relationships) in a graph structure.

KGs enable the aggregation and interpretation of diverse data sets, facilitating data integration, discovery and inference capabilities across diverse domains. One of the most

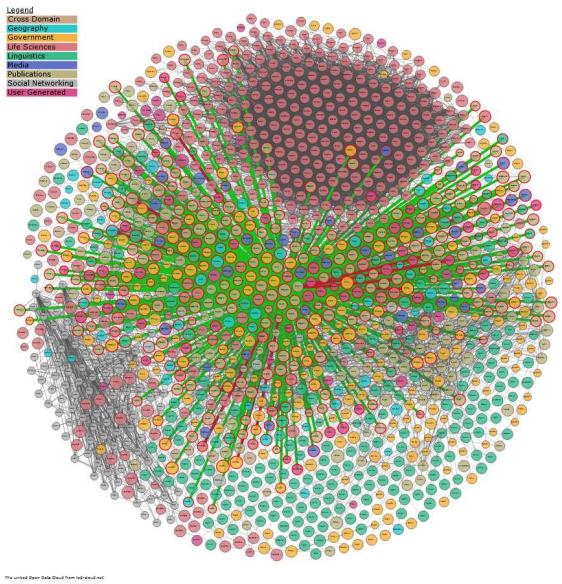


Figure 1.1: Linked Open Data Cloud (LOD-Cloud) and DBpedia in the centre: its connections with other KGs are in green

significant applications is the **Linked Open Data** (LOD), which is a collection of freely accessible RDF knowledge graphs published on the Web, especially through SPARQL endpoints, covering diverse domains such as social networks, science, media, and publications. According to the **LOD-Cloud** catalog,[1] the number of integrated datasets has steadily grown. Fig. 1.1 represents a schema of the LOD-Cloud KGs interconnected between them, where the **DBpedia** KG[2] is one of the most outstanding graphs due to its links (in green) with the others. The DBpedia project focuses on automatically extracting data from *Wikipedia*. It aims to deliver a structured and standardized representation of Wikipedia's content and connect facts with other open RDF datasets from the Web of Data.

Constructing knowledge graphs with large factual information from diverse sources is well-established. However, elaborating schemes and semantics, in other words, models representing these data (rules, constraints, relations), is a more challenging and resource-intensive task. The lack of information in the rules and the constraints that explain the data leads to different kinds of *inconsistencies* and *incompleteness*, which has a negative impact on the use of these data graphs for various applications.

[1] https://lod-cloud.net/
[2] https://www.dbpedia.org/

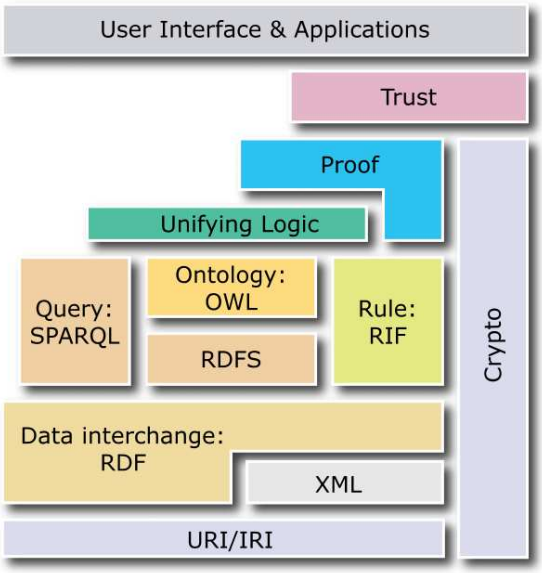


Figure 1.2: Semantic Web standards hierarchy

1.2 Motivations

We understand that the schemes, rules, and constraints that add semantics to the facts must ensure consistency within a specific context, validate them, and infer new facts by reasoning. Based on the principle that knowledge graphs are rich in factual information, we believe it is an essential starting point for enriching knowledge graphs: this is why we assume that a *bottom-up* approach (*i.e.*, from RDF data to schemes that explain them) is suited for discovering new knowledge. However, such an approach must consider RDF facts' incompleteness and inherent inconsistencies to discover relevant and exploitable rules and constraints.

In this thesis, we define the scope of new knowledge in the form of (1) **OWL subsumption axioms** and (2) **SHACL shapes**. On the one hand, subsumption axioms provide rich information on the relations between concepts described in ontologies, which can infer new facts and enrich the connections between entities and concepts. On the other hand, SHACL shapes are used to express **constraints** that RDF facts must respect in a knowledge graph. These constraints capture the knowledge domain and are used in the context of model validation, type checking, etc. Moreover, users can use SHACL shapes to integrate new data continuously.

We propose searching and discovering new knowledge from an RDF data graph using an *evolutionary process*. Therefore, the approach must consider how candidate knowledge

is modelled within this process: we believe that the most pertinent approach should be based on an **Evolutionary Algorithm** (EA) and, more specifically, on a **Grammatical Evolution** (GE) technique.

Finally, we consider Web standards (see Fig. 1.2) essential for discovering credible knowledge that can be produced and fairly evaluated so that humans and artificial agents can exploit these candidate solutions (*i.e.*, integrate them, infer new facts, ...).

1.3 Foundation

1.3.1 Web of Data

1.3.1.1 RDF data graphs

An **RDF data graph** is a set of interconnected RDF triples whose terms are IRIs, literals and blank nodes (anonymous resources). The use of RDF to express facts enables **interoperability** between systems, notably through the addition of standards allowing the creation and distribution of knowledge graphs through the Web, making them usable by both humans and artificial agents, and it enables interconnection between them. An **RDF triple** $< \mathbf{s}, \mathbf{p}, \mathbf{o} >$ expresses a relation $\mathbf{p}$, called *predicate*, between a *subject* $\mathbf{s}$ and an *object* $\mathbf{o}$: $\mathbf{s}$ is a resource (an *IRI* or a *blank node*); $\mathbf{p}$ is an *IRI* and $\mathbf{o}$ is any RDF term. Fig. 1.3 is an example of an RDF data graph around *Johnny Depp*, and it describes the following piece of knowledge:

"*Johnny Depp has been an actor in activity since 1984. He starred in several films, including the film series 'Pirates of the Caribbean' in which he portrays the character of Jack Sparrow.*"

As an example, the RDF triples `dbr:Johnny_Depp rdf:type dbo:Actor` and `dbr:Johnny_Depp dbp:yearsActive "1984"^^xsd:integer` models the first sentence. Each fact in this graph (which is not literal) is a synthetic representation of the IRIs through prefixes, simplifying the notation of the IRIs: *e.g.*, `dbo:Actor` is the synthetic expression of `<http://dbpedia.org/ontology/Actor>`. Table 1.1 lists all the prefixes commonly used in the works presented in this thesis. RDF literals can be typed in various ways: *core types* (*e.g.*, `xsd:boolean`, `xsd:decimal`, ...), *time and date* (*e.g.*, `xsd:date`, `xsd:dateTime`, ...), ... and the language can be specified as well, *e.g.*, `"Capitão Jack Sparrow"@pt` means that the sentence is written in Portuguese. RDF data can be serialized with several syntaxes, *Turtle* being the simplest and most readable one.

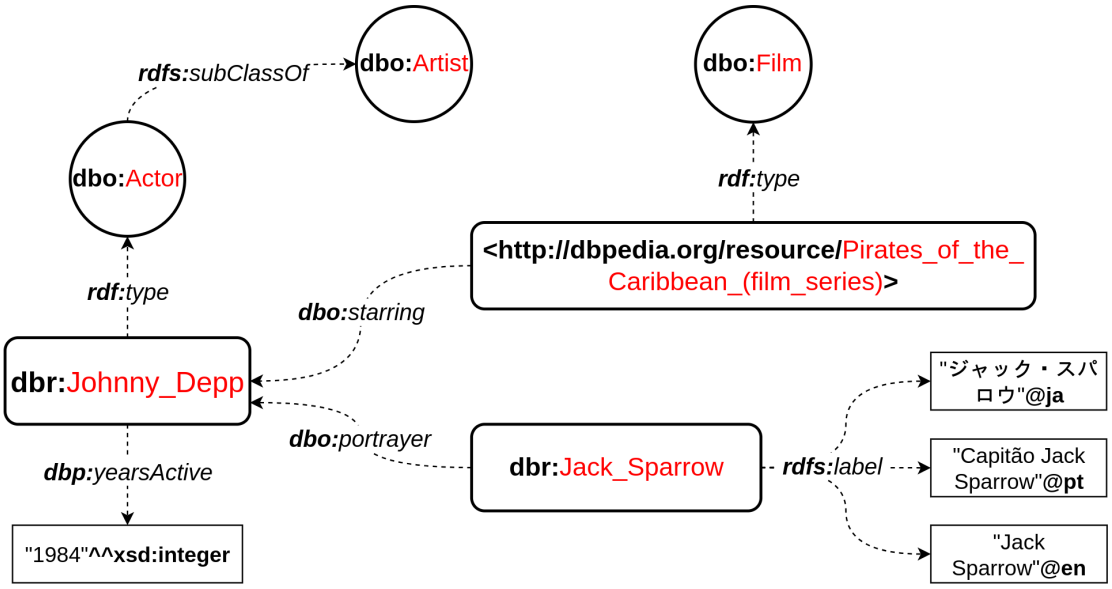


Figure 1.3: Example of an RDF data graph

Prefix	URI
rdf	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs	<http://www.w3.org/2000/01/rdf-schema#>
owl	<http://www.w3.org/2002/07/owl#>
dbo	<http://dbpedia.org/ontology/>
dbr	<http://dbpedia.org/resource/>
foaf	<http://xmlns.com/foaf/0.1/>
sh	<http://www.w3.org/ns/shacl#>
psh	<http://ns.inria.fr/probabilistic-shacl#>

Table 1.1: Prefixes commonly used in this thesis and their full URI

1.3.1.2 SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) [HSP13] is the RDF query language used to extract and manipulate RDF facts in a datastore over the Web using SPARQL endpoints. Moreover, it comprises a representation language for query results: *e.g.*, in *XML*, *HTML*, *JSON*, ... SPARQL enables the modelling of graph pattern as a set of RDF triples where one or more of the triple components (**s**, **p** or **o**) may be a **variable**: each variable starts with a question mark and is named, *e.g.*, `?x`. A SPARQL query can be designed for:

- *extracting* bindings between variables and resources: **`SELECT`**
- *asking* if a query pattern has a solution or not: **`ASK`**
- *building* others RDF data graphs using RDF facts: **`CONSTRUCT`**
- *adding/deleting* explicit RDF triples: **`INSERT DATA`**/**`DELETE DATA`**
- *adding/deleting* computed RDF triples: **`INSERT`**/**`DELETE`**
- *extracting* the description of resources: **`DESCRIBE`**

Fig. 1.4 presents an example of a SELECT query (Fig. 1.4a) to search the actor who portrays *Jack Sparrow* in the RDF data graph presented in Fig. 1.3 and its result is presented in Fig. 1.4b. Naturally, it is possible to select more than one variable as long as they are modelled in the graph pattern. Moreover, graph patterns in SPARQL query may be more complex: it can be a set of basic graph patterns (Group Graph Pattern), and some can be optional (Optional Graph Pattern).`FILTER` clauses are used to restrict solutions (*i.e.*, set of bindings of variables to RDF terms) to those for which the filter expression evaluates to true [HSP13], *e.g.*, by adding `FILTER(?when > 1990)` in the body of the SPARQL query presented in Fig. 1.4a, we limit the possible solutions to RDF terms (*i.e.*, integers bound to the variable `?when`) whose value is strictly greater than 1990: this filter means that the binding shown in Fig. 1.4b is no longer a solution to the query, so the final solution is empty.

Federated queries can compute graph patterns over different SPARQL endpoints. They allow RDF data (resulting from these queries) from a remote source to be used with other local (or remote) RDF data sources.[3] Furthermore, a wide range of options and operators enable more in-depth manipulation of RDF data: all those used in this thesis are presented in Table 1.2.

[3] Chapter 3 presents some practical uses of federated queries

```
# Prefixes definition
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
# Searching the actor (?who) who portrays Jack Sparrow
SELECT ?who ?when WHERE {
    # Basic Graph Pattern:  set of RDF triples that must match
    # ?who must be an actor
    ?who rdf:type dbo:Actor .
    # ?who must portray Jack Sparrow
    dbr:Jack_Sparrow dbo:portrayer ?who .
    # In what year did ?who start his career?
    ?who dbp:yearsActive ?when .
    # ?when must be greater than 1980
    FILTER(?when > 1980)
}
```

(a) "*Which actor portrays Jack Sparrow? In what year did he start his career?*" interpretation in SPARQL

?who	?when
<http://dbpedia.org/resource/Johnny_Depp>	1984

(b) Results of the query

Figure 1.4: Example of a SPARQL query on the RDF data graph presented in Fig. 1.3 and its result

1.3.1.3 Ontology Modeling Languages

OWL is the language proposed by the W3C [MvHE04] to formally describe the meaning of the terminology used in documents written in RDF, enhancing interoperability between agents. It is an extension of the RDF Schema (RDFS), a data-modelling vocabulary for RDF data to describe groups of resources and the relationships between these resources [BGE14]. OWL has been designed to perform reasoning tasks on RDF documents and extend vocabulary to describe relations between classes (*e.g.*, subsumption), class intersection, cardinality, etc. Moreover, OWL ontologies are designed under the **Open-World Assumption** (OWA), which assumes that the absence of a statement does not mean the statement is false.

OWL 2 provides new functionality such as property chains, qualified cardinality restrictions, etc.; a new syntax (OWL 2 Manchester syntax) and three OWL 2 profiles (in increasing order of expressiveness) [BFH+12]:

- `OWL 2 EL`: enables polynomial computation time for reasoning tasks, suitable for *large* ontologies (it is possible to focus on performance instead of expressive power).
- `OWL 2 QL`: enables conjunction queries computed in LogSpace, focused on ontologies which organise a large set of individuals that must be accessed through relational queries, *e.g.*, SQL.
- `OWL 2 RL`: enables rule-extended database technologies operating directly on RDF triples.

OWL axioms are used to express factual statements about RDF data (concepts, ...) that are accepted as self-evident [BFH+12]. There exist 32 types of axioms divided into 6 specific categories: **class expression** axioms; **object property expression** axioms; **data property expression** axioms; **datatype definition** axioms; **keys** axioms and **assertion** axioms.

In Fig. 1.3, the RDF triple `dbo:Actor rdfs:subClassOf dbo:Artist` is a subsumption axiom (part of the class expression axioms) which can be written in functional notation: `SubClassOf(dbo:Actor dbo:Artist)`. This axiom means that all actors are artists, and consequently, it is possible to infer that "*Johnny Depp is an artist*": `dbr:Johnny_Depp rdf:type dbo:Artist`.

The OWL 2 functional-style syntax[4] allows OWL 2 ontologies to be written in a compact form and allows to write abbreviated IRIs, facilitating the reading/writing of ontologies.

[4] https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Functional-Style_Syntax

Keyword	Definition	Example
`DISTINCT`	modifier to ensure **unique** solutions in results	`SELECT DISTINCT ?x WHERE ...`
`COUNT`	count the number of **occurrences** in results	`SELECT COUNT( ?x ) WHERE ...`
`VALUES`	it provides inline data as **a solution sequence** of variable(s)	`...VALUES ?x { dbo:Actor dbo:Film }...`
`FILTER NOT EXISTS`	it tests whether a given graph pattern **does not match** the dataset	`...FILTER NOT EXISTS { ?x rdf:type dbo:Cinema }...`
`ORDER BY`	it defines the **order** of a solution sequence	`...} ORDER BY ?age`
`LIMIT...[ OFFSET ]`	**restrict** the number of solution and (optionally) control **where the solutions start** from the whole set result	`...} LIMIT 100 OFFSET 50`
`SERVICE`	express a graph pattern to **query over a SPARQL endpoint**	`...SERVICE <http://dbpedia.org/sparql> {...}`

Table 1.2: SPARQL options [HSP13]: definitions and examples

1.3.1.4 Shapes Constraint Language

SHACL (Shapes Constraint Language) is a W3C recommendation for validating RDF graphs against a defined set of conditions, *i.e.*, constraints [KK17]. These conditions are represented as **shapes**: a set of shapes is called *shapes graph* while the RDF facts being validated against a shapes graph are part of the *RDF data graphs*. In addition to validation, SHACL shape graphs can describe data graphs satisfying specific conditions. These descriptions can serve various purposes: *e.g.*, data quality control, data integration, ...

A shape s is an instance of `sh:NodeShape` or `sh:PropertyShape`. Focusing on node shapes, they must satisfy the following structure: first, they must **target** a specific set of nodes in the RDF data graph. 4 different targets are available as a *predicate* of the subject s:

- **`sh:targetNode`**: target nodes directly specified as objects
- **`sh:targetClass`**: target nodes that are **instances** of the specified *object*
- **`sh:targetSubjectsOf`**: target nodes that are **subject of the specified predicate**
- **`sh:targetObjectsOf`**: target nodes that are **object of the specified predicate**

Examples of different target types applied to the RDF data graph shown in Fig. 1.3 are presented in Table 1.3. Second, constraints are expressed in a shape using the parameters of **constraint components**. Various constraint components are available in **SHACL core**[5], such as *value type constraint* components (with parameters `sh:class`, `sh:datatype`, ...), *cardinality constraint* components (with parameters `sh:minCount` and `sh:maxCount`). Additionally, the SPARQL-based constraints, or **SHACL-SPARQL**[6], can be used to express constraints through SPARQL `SELECT` query to address use cases not covered by SHACL core.

Target type	Example	Targeted nodes
`sh:targetNode`	`:s sh:targetNode dbr:Jack_Sparrow`	`dbr:Jack_Sparrow`
`sh:targetClass`	`:s sh:targetClass dbo:Actor`	`dbo:Johnny_Depp`
`sh:targetSubjectsOf`	`:s sh:targetSubjectsOf dbp:portrayer`	`dbr:Jack_Sparrow`
`sh:targetObjectsOf`	`:s sh:targetObjectsOf dbp:portrayer`	`dbo:Johnny_Depp`

Table 1.3: Example of targets applied on RDF data graph presented in Fig. 1.3

[5] https://www.w3.org/TR/shacl/#core-components
[6] https://www.w3.org/TR/shacl/#sparql-constraints

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX sh: <http://www.w3.org/ns/shacl#>
# Shape definition
:s  a sh:NodeShape ;
    # targeting instances of dbo:Film
    sh:targetClass dbo:Film ;
    sh:property [
        # focusing object of the dbo:starring predicate
        sh:path dbo:starring ;
        # testing if (at least) one value is an instance
        # of dbo:Actor
        sh:class dbo:Actor ;
    ] .
```

Figure 1.5: Example of a SHACL shape with a class constraint component

```
PREFIX sh: <http://www.w3.org/ns/shacl#>

[
    a sh:ValidationReport ;
    sh:conforms true
] .
```

Figure 1.6: SHACL validation report obtained from the assessment of the RDF data graph presented in Fig. 1.3 against the SHACL shape presented in Fig. 1.5

Let us consider the assessment of the RDF data graph presented in Fig. 1.3 with the SHACL shape presented in Fig. 1.5 expressing the fact that "*all film stars must be actors*". A focus node conforms to a shape if and only if the validation of the focus node against the shape results in an empty set and does not report any failure. In this example, the focus node is `<http://dbpedia.org/resource/Pirates_of_the_ Caribbean_(film_series)>`[7] and it is the only instance of `dbo:Film` in the RDF data graph to be validated during the **SHACL validation process**. Next, the object of the predicate `dbo:starring` is considered, *i.e.*, `dbr:Johnny_Depp`, and a test is done to check whether it is an instance of `dbo:Actor`.

As `dbr:Johnny_Depp rdf:type dbo:Actor` exists in the RDF data graph, the data graph conforms to the shape, producing the **SHACL validation report** presented in Fig. 1.6.

[7] The prefix cannot be used because of the parentheses "(" ")" in the URI

1.3.2 Evolutionary Algorithms and Grammatical Evolution

Evolutionary Algorithms (EA) are a family of population-based stochastic optimisation algorithms that simulate natural selection mechanisms to solve complex problems by producing individuals as "candidate solutions" to the considered problem. The principles of Darwinian evolution heavily influence evolutionary algorithms. *Genetic algorithms*, *evolutionary programming*, *evolution strategies* and **Genetic Programming** (GP) are the four subdomains of evolutionary algorithms. Concerning the genetic programming approach, individuals in a population are represented as symbolic expressions, *e.g.*, sets of binary values, that can be interpreted and executed by a computer system [BFM00]. The aim is to solve the given problem by obtaining *local* optimum solutions (that approximate the solution) or *global* optimum solution (the most satisfying ones).

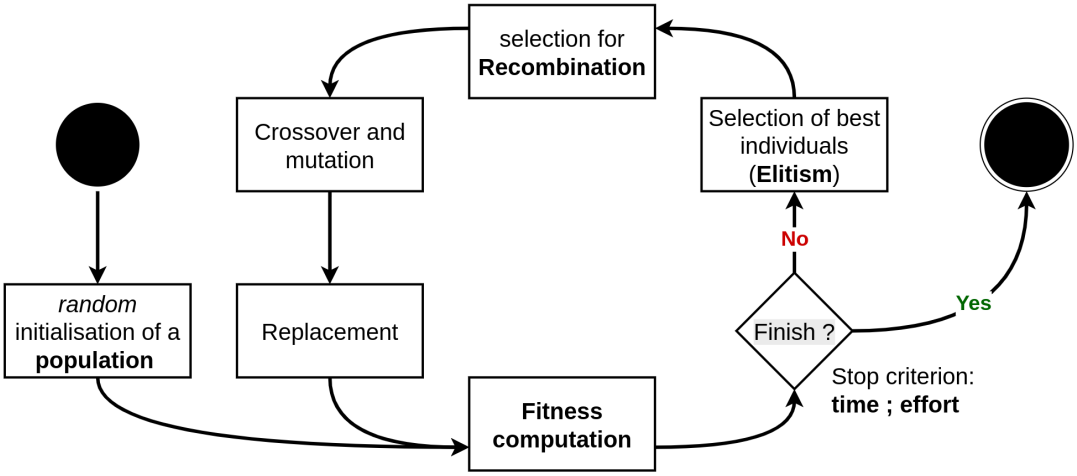


Figure 1.7: Evolutionary algorithms pipeline

In Figure 1.7, the fundamental steps of the algorithm are presented and described as follows:

1. Randomly generates a **population** composed of candidate solutions to the problem: they are individuals of this population.
2. Each individual is assessed against a **fitness function**, a mathematical expression that describes how well an individual fits as a solution to the problem. Depending on the objectives, we may want to *maximise* this fitness value, *minimise* it or *converge* towards a precise value.
3. Checking the algorithm's stopping criterion:

4. **If the stop criterion is reached**, it returns the final solutions found during the evolutionary process.

5. **Else (the stop criterion is not reached)**, we select the *best fitted* individuals to preserve them from the evolutionary process: they are part of the **elitist** sub-population and have a great chance of producing new solutions (*i.e.*, **offsprings**) that "look like" them.

6. Non-elitist individuals are subject to the process of evolution, in which most will transform into new solutions and, in rare cases, will be kept as they are. We select (in a *pseudo-random* way) a sub-population (from the whole population) that will reproduce with each other and become a new subset of solutions: *i.e.*, offspring.

7. Selected individuals for reproduction undergo **crossover** and **mutation** phasis with a certain probability of occurrence. Crossover occurs fairly often and involves *swapping one or more segments* of genetic information between two individuals. In contrast, mutation occurs less frequently and results in a *random alteration* of one or more characteristics of an individual.

8. Obtained individuals from the recombination phasis replace non-elitist individuals in the population (*i.e.*, **replacement**).

9. Repeat step 2 and so on until **the stop criterion is reached** (Step 4).

The population will evolve through several generations, and natural selection, described in this way, progressively improves the quality and credibility of the solutions in a population. The challenge is, therefore, to find the optimal parameters for discovering the best solutions at a reasonable computing cost.

Grammatical Evolution (GE) is a particular type of genetic programming (GP), an evolutionary algorithm. This approach automatically makes it feasible to generate variable-length expressions in any language [OR01]. This technique is based on the expression of grammar, and more specifically **BNF grammar** (Backus-Naur Form), which is used for the **genotype/phenotype mapping**: the Grammatical Evolution architecture is presented in Fig. 1.8. A BNF grammar is a context-free grammar composed of terminals and non-terminals, allowing instantiating of expressions that conform to the rules expressed using a sequence of integers. A **genotype** is a set of integers, *i.e.*, a set of **codons**, corresponding to the unique identifier of the information it characterises, which is used in the evolutionary process. Each codon translates a chunk of information that

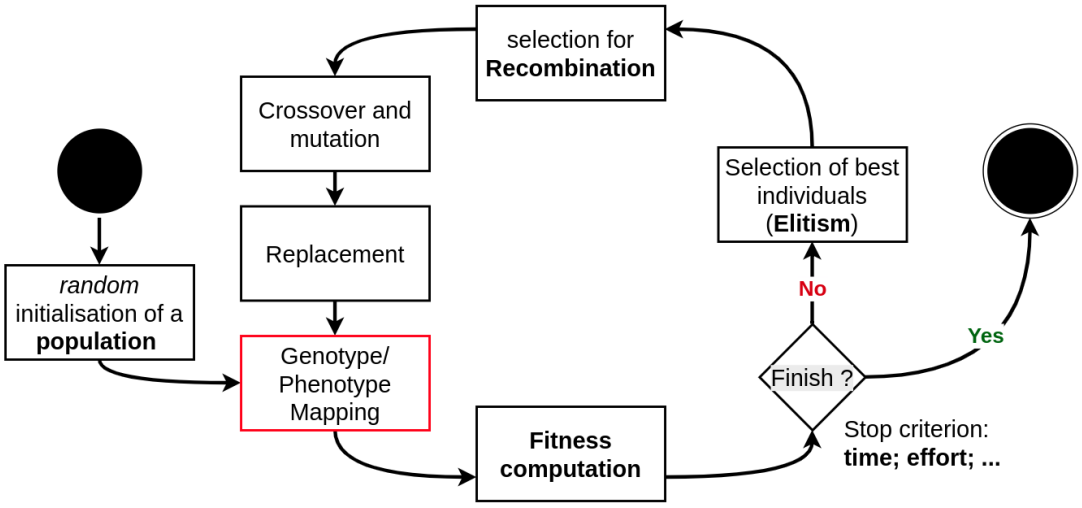


Figure 1.8: Grammatical Evolution pipeline

forms a human-readable individual, whose information is expressed in the **phenotype**. In other words, the genotype/phenotype mapping relies on a *derivation process* to select the adequate production rule (replacing non-terminal). In GE, the **wrapping** parameter enables a maximum number of iterations to produce a well-formed expression, *e.g.*, useful for grammar rich in derivation rules with a limited number of codons (resulting in an incomplete phenotype).

To illustrate this, let us consider the BNF grammar in Fig. 1.9 to produce math formulas as individuals in an evolutionary process: *e.g.*, "`x + y`", "`2`", "`9 - x`", ... Each codon corresponds to the rule to be selected for production (and so on for as long as it is a non-terminal): the result of the **modulo** between the *codon* and the *number of productions* for the current rule determines the index of the rule be considered at each step (until a terminal is reached). Considering the example in Fig. 1.10, the rule `<Formula>` is the starting point to produce a phenotype, and 18 is the first integer in the codons set. As the number of productions for this rule is 2 (`<ope> <exp> <exp>` or `<exp>`), the chosen production is the *first one*: `<ope> <exp> <exp>` ($18 \mod 2 = 0$, *i.e.*, the first rule).

```
<Formula>    :=  <exp> <ope> <exp> | <exp>
<ope>        :=  "+" | "-" | "*"
<exp>        :=  <letter> | <digit>
<digit>      :=  "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" |
                 "9" | "10"
<letter>     :=  "x" | "y" | "z"
```

Figure 1.9: Example of BNF grammar

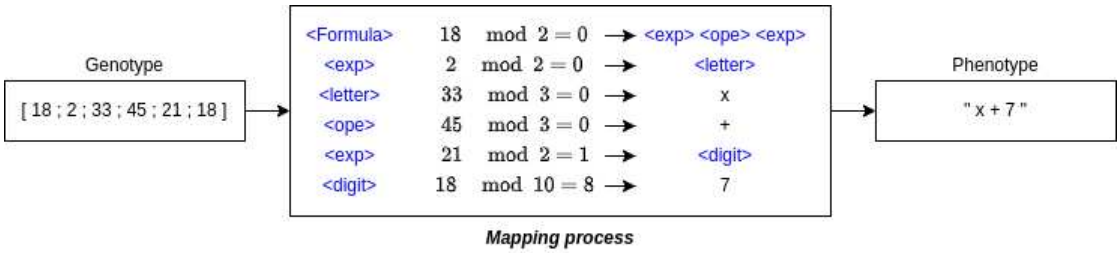


Figure 1.10: Applied example of genotype/phenotype mapping using the BNF grammar presented in Fig. 1.9

1.4 Research Questions

Motivations presented in Section 1.2 led us to consider the discovery of knowledge from RDF data using an evolutionary process. This is why we tackle the following research questions:

RQ1: *How to overcome computation time issue to assess subsumption axioms in the Open-World Assumption?*

RQ2: *How to automatically discover class subsumption axioms from RDF data?*

RQ3: *How to design a validation process considering physiological errors in real-life data?*

RQ4: *How to automatically discover SHACL shapes from RDF data?*

The computation time issue for assessing candidate subsumption axioms is crucial because it is an obstacle to the scalability of the proposed method: this is why **RQ1** underlies **RQ2**. Regarding the discovery of SHACL shapes representative of an RDF data graph (**RQ4**), we suggest that inherent inconsistencies from an RDF data graph should be considered when validating an RDF data graph against candidate shapes (**RQ3**).

1.5 Contributions

In this thesis, we propose an evolutionary method for the discovery of new knowledge (described in Section 1.2) based on the use of standard semantic Web technologies combined with a Grammatical Evolution algorithm to (1) produce candidate solutions (*i.e.*, new knowledge assumed), (2) fairly *assess* these candidate solutions using RDF facts and (3) discover a large set of credible solutions by retaining the best individuals through the evolutionary process.

We propose to focus on the evolutionary discovery of OWL subsumption axioms `SubClassOf` that can be composed of complex class expressions. The foundations of this work are based on the evolutionary discovery of disjointness class axioms [NT19b] and the possibilistic assessment of `SubClassOf` axioms [TFG17]. This route required a two-stage approach: (1) we tackle the computation time issue for retrieving exceptions of candidate subsumption axioms, and (2) we propose an adaptation of the Grammatical Evolution to produce candidate subsumption axioms representative of RDF data graph (that can be composed of complex class expressions) and assess them using RDF data graph and the possibilistic framework [TFG17].

Second, we propose a framework for including and exploiting probabilistic information in SHACL validation reports. This framework considers inherent errors and incompleteness in RDF data graphs when validating them against SHACL constraints.

We studied the evolutionary discovery of candidate SHACL shapes regarding an RDF data graph using the proposed probabilistic framework adapted for the candidate shapes assessment.

Finally, we developed the `RDFminer` software as a user interface to perform the evolutionary discovery of candidate shapes or candidate axioms: it is a tool to (1) create projects through an interactive dashboard, (2) analyse the results of their projects on the dashboard in real-time, and (3) control their execution.

1.6 Publications

In relation to the research questions presented in Section 1.4, the following contributions have been published:[8]

1. **[RQ2]** [FT21] Rémi Felin and Andrea G. B. Tettamanzi. "**Using grammar-based genetic programming for mining subsumption axioms involving complex class**

[8] They are available on HAL: `https://cv.hal.science/remi-felin`

expressions". In: WI-IAT 2021 - 20th IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology. Melbourne, Australia

2. **[RQ1]** [FCFT22] Rémi Felin, Olivier Corby, Catherine Faron and Andrea G. B. Tettamanzi. "**Optimizing the Computation of a Possibilistic Heuristic to Test OWL SubClassOf Axioms Against RDF Data**". In: WI-IAT 2022 - 21th IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Nov 2022, Niagara Falls, Canada

3. **[RQ3]** [FFT23a] Rémi Felin, Catherine Faron and Andrea G. B. Tettamanzi. "**A Framework to Include and Exploit Probabilistic Information in SHACL Validation Reports**". In: ESWC 2023 - 20th European Semantic Web Conference, May 2023, Hersonissos, Greece

4. **[RQ3]** [FFT23b] Rémi Felin, Catherine Faron and Andrea G. B. Tettamanzi. "**Un cadre pour inclure et exploiter des informations probabilistes dans les rapports de validation SHACL**". In: IC 2023 - 34es Journées francophones d'Ingénierie des Connaissances @ Plate-Forme Intelligence Artificielle (PFIA 2023), Jul 2023, Strasbourg, France

5. **[RQ4]** [FMFT24b] Rémi Felin, Pierre Monnin, Catherine Faron and Andrea G. B. Tettamanzi. "**Extraction probabiliste de formes SHACL à l'aide d'algorithmes évolutionnaires**". In: EGC 2024 - Extraction et Gestion de la Connaissance, Jan 2024, Dijon, France

6. **[RQ4]** [FMFT24a] Rémi Felin, Pierre Monnin, Catherine Faron and Andrea G. B. Tettamanzi. "**An Algorithm Based on Grammatical Evolution for Discovering SHACL Constraints**". In: EuroGP 2024 - 27th European Conference on Genetic Programming, Apr 2024, Aberystwyth, United Kingdom

7. **[RQ2 & RQ4]** [FMFT24c] Rémi Felin, Pierre Monnin, Catherine Faron and Andrea G. B. Tettamanzi. "**RDFminer: an Interactive Tool for the Evolutionary Discovery**

of SHACL Shapes". In: ESWC 2024 - 21st European Semantic Web Conference. Hersonissos, Greece

1.7 Outline

The remainder of this thesis is organised as follows:

- Chapter 2 is a literature review that presents existing works in the scope of Ontology Learning techniques, SHACL shapes mining, and the existing works related to Grammatical Evolution.
- Chapter 3 presents (1) optimisations to address the problem of computation time for retrieving exceptions of candidate subsumption axiom (**RQ1**) and (2) the evolutionary discovery of `SubClassOf` axioms composed of complex class expression (**RQ2**).
- Chapter 4 introduces the probabilistic framework (**RQ3**): the mathematical foundations of the model, the associated vocabulary for expressing probabilistic measures in SHACL validation reports and how to accept RDF data against shapes using hypothesis testing.
- Chapter 5 presents the evolutionary discovery of candidate SHACL shapes (**RQ4**): a description of the BNF grammar to produce candidate shapes, an adaptation of the probabilistic framework to assess candidate shapes, an extended recombination and GE operators.
- Chapter 6 presents the `RDFminer` software: its architecture and a comprehensive description of the features needed to discover candidate SHACL shapes or candidate axioms.
- Chapter 7 is a general conclusion to this thesis, with an overview of the perspectives.

Chapter 2

Literature Review

2.1 Ontology Learning

Developing structured and consistent ontologies for a given domain is a resource-intensive task, as this task often requires in-depth analysis by domain experts and knowledge engineers to ensure consistency and interoperability of ontologies. Furthermore, ontologies are built under the OWA, where not knowing a given information does not make it false. In this way, knowledge engineers are also faced with incompleteness issues, which impact the coverage of the domain within ontologies. Lehmann and Völker define these issues as the *knowledge acquisition bottleneck* [LS11] which must be addressed by the community.

Ontology Learning (OL) is a multifaceted research area focused on (semi-)automating the creation, enhancement, and maintenance of ontologies [MS04] from various data sources: *e.g.*, text, databases, structured data, ... This research area addresses the *knowledge acquisition bottleneck* issues and can be seen as a *pipeline* from (un)structured data (*e.g.*, plain text) to the final ontology, where ontology learning techniques are successively used to parse data, extract information and organize it in a structured knowledge base [AWK+18]: Figure 2.1 presents an overview of the OL pipeline. From a literature review, we distinguish three sub-domains (although some of the presented approaches are hybrid).

2.1.1 Linguistic Techniques

Methods based on linguistic techniques, part of Natural Language Processing (NLP), are commonly used on unstructured text as the first steps of the whole process: *i.e.*, pre-processing. The idea is to give machines the ability to process complex grammatical structures and the semantics of sentences, enabling the development of artificial agents for various real-world applications, including ontology building and knowledge enrichment.

First, the *speech tagging*, or POS tagging (Part-Of-Speech tagging), technique is used to annotate each word in a sentence with its corresponding part of speech (*i.e.*, nouns,

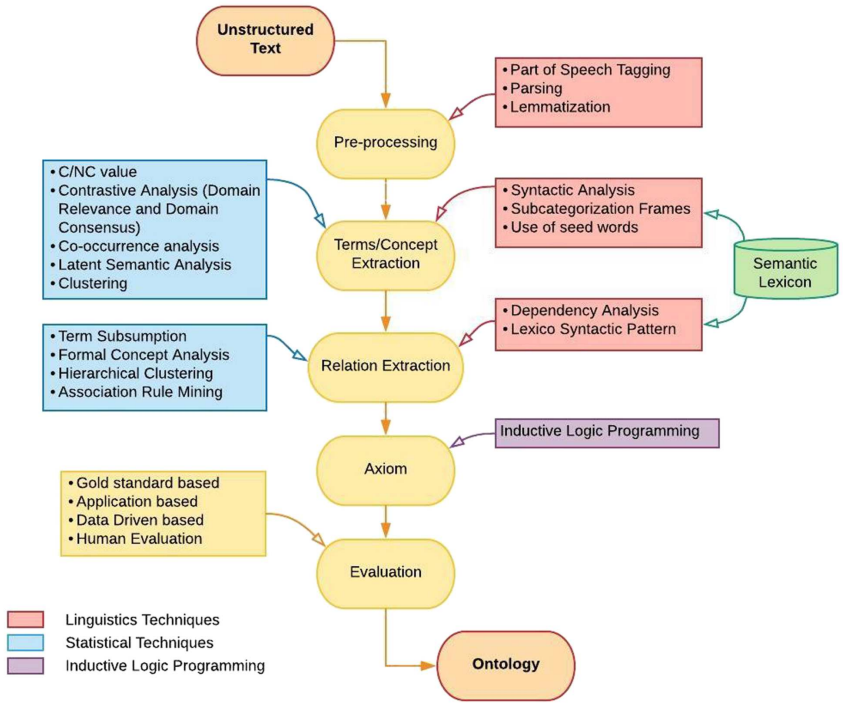


Figure 2.1: Ontology Learning pipeline presented by Asim et al. [AWK+18]

verbs, adjectives, adverbs, pronouns, conjunctions, prepositions, and interjections) to provide the grammatical structure (in other words, a grammatical classification) of sentences and make it understandable by machines. Recently, many Deep Learning and Machine Learning-based approaches have been proposed for POS tagging tasks to tackle *ambiguity* issues and overcome the tagging of *complex words* over different domains [ZY22].

The second aspect to consider is the identification of concepts and their relations in sentences, namely *entity extraction* or Named-Entity Recognition (NER) and *relation extraction*. NER algorithms classify named entities in pre-defined categories, *e.g.*, Person, Organisation, Animal, ... Nowadays, several software packages are available to perform NER algorithm through unstructured text: *e.g.*, SpaCy [1], Stanford Named Entity Recognizer [2] or DBpedia Spotlight [3], ... and their performance fluctuates depending on the corpus of texts under consideration [SKR+19].

The ever-increasing volumes of data (here: unstructured text) drive the need for high-performance tools to create and enrich knowledge in highly complex domains, such

[1] https://spacy.io/api/entityrecognizer
[2] https://nlp.stanford.edu/software/CRF-NER.shtml
[3] https://www.dbpedia-spotlight.org/api

as biomedical. Research is trending towards models based on transformers or Large Language Models (LLMs) that perform better than conventional Deep Learning models (*e.g.*, Recurrent Neural Networks RNN, Long-Short Term Memory LSTM, ...) because they allow very large volumes of data to be learned efficiently. For a comprehensive overview of research based on these models for ontology construction, see [ZVFD24]. However, Mai et al. [MCP24] empirically demonstrate that LLMS does not adapt as well as hoped to arbitrary domains and perform less well on relation extraction and taxonomy discovery tasks.

Interestingly, some Deep Learning-based approaches have been proposed to extract axioms from text which, in the meantime, performs all the intermediate tasks (detection of entities, concepts, ...): Petrucci et al. [PGR16] proposed an RNN model for a transduction task, associating sentence transduction and sentence tagging, to transform a sentence to a DL-formula (that can be written with OWL). Cai et al. [CKXS17] proposed a Deep Neural Network (DNN) model, combining symbolic manipulation and reasoning for axiom discovery. `Neural Reasoner` [PLLW15] is a framework for neural network-based reasoning over natural language sentences to extract axioms and rules.

2.1.2 Statistical Techniques

Most statistical-based approaches, mainly focus on probabilistic models, aimed to address the *concepts*, *terms* and *relations extraction* from semi-structured text or structured datasets.

As a hybrid approach, the *C/NC value* combines linguistic techniques and statistical information for automatic recognition of multi-word terms [FAT98]. Similarly, `OntoLearn` [NVG03] is an architecture based on NLP and ML techniques to automatically translate multi-word terms from English to Italian.

Subsequently, clustering methods have been used for concept and term extraction: Karoui et al. [KAB06] proposed an unsupervised hierarchical clustering algorithm based on K-means and guided by a structural context to extract concepts from HTML documents. More recently, Xu et al. [XHGI20] combined word embeddings and clustering techniques (K-means, K-medoids, affinity propagation, DBscan and co-clustering algorithms) as a term clustering method for building modular ontologies. `OntoGain` [DZP10] is a framework that exploits inherent multi-word terms lexical information, hierarchical clustering and Formal Concept Analysis (FCA) methods to extract taxonomic relations and association rules mining (and probabilistic) techniques for non-taxonomic relations.

FCA algorithms combine statistics and concept analysis to identify conceptual structures among datasets, producing graphical visualisations of the structures among data.

Association Rules Mining (ARM) methods are exclusively used for relation extraction based on the *co-occurrence* of elements (or *items*) in the dataset. Let us consider $\mathcal{I}$ the *item* set and $\mathcal{T}$ the *transaction* set, an Association Rule (AR) between $\mathcal{X}$, called the *antecedent*, and $\mathcal{Y}$, *i.e.*, the *consequent* (considering $\mathcal{X} \in \mathcal{T}$, $\mathcal{Y} \in \mathcal{T}$ and $\mathcal{X} \cap \mathcal{Y} = \emptyset$) is written as follows: $\mathcal{X} \rightarrow \mathcal{Y}$, reflecting co-occurrence rather than causality. Many algorithms have been proposed for AR learning from data: one of the most impactful is the `Apriori` algorithm [AIS93], whose behaviour (*i.e.*, exploiting frequent item-sets) is still used as a basis for optimising this algorithm, especially its time consumption on large datasets [POP01, YC06]. The ARs are generally assessed using *confidence* and *support* measures. However, some research aims to extend these metrics to fit more complex use cases: *e.g.*, for assessing ARs extracted from OWL ontologies [TdTN19] or defining an *interestingness* metric [TKS02, CT20] aimed to highlight surprising and relevant ARs.

Interestingly, some research has been conducted in the context of Linked Data to extract association rules from RDF facts: `AMIE` [GTHS13] (more recently `AMIE+` [GTHS15]) is a scalable framework to mine Horn-like rules on large RDF knowledge bases under the OWA; Cadorel and Tettamanzi [CT20] proposed an ARM method based on clustering and community detection to extract association rules from a large RDF knowledge graph related to the scientific domain.

2.1.3 Inductive Logic Programming

Methods based on Inductive Logic Programming (ILP) techniques are used at the end of the ontology learning process (before the ontology assessment) to extract **rules** and **axioms** from previously established concepts and relations. The idea is to use concepts and their relations, *i.e.*, examples and background knowledge, to logically infer facts (*i.e.*, hypothesis) that can be exploited as a top-level schema. In other words, ILP is a type of ML that uses logic programming techniques to derive logical theories (*i.e.*, first-order rules) from knowledge representation [VN11]. However, ILP techniques are used for logical inference (from knowledge representation), whereas ML techniques rely on statistical inference [CD22]. As notable advantages of ILP techniques, they (1) do not need many examples for the training phase, (2) provide high-level explainable rules from assertions, and (3) easily allow knowledge transfer methods [CD22].

As a hybrid ILP approach, Lima et al. [LEO+13] proposed a method that combines text preprocessing techniques (lexico-syntactic analysis, ...) to parse a given corpus

and then induce rules (as Horn clauses) to populate ontologies from the corpus. More recently, Lima et al. proposed `OntoILPER` [LEF18], which exploits an ontology and rules induction for extracting entities and relations from unstructured text.

The First-Order Inductive Learner (FOIL) algorithm, introduced by Quinlan [Qui90] in 1990, is one of the most significant contributions in the ILP field: `FOIL` learns function-free Horn rules based on positive and negative examples, *i.e.*, these rules express relations between concepts supported by examples in the dataset. Many works have been proposed as an extension of the FOIL algorithm to tackle scalability and performance issues: Fanizzi et al. [FdE08] proposed `DL-FOIL`, a FOIL-like algorithm, to learn concept description expressed in description logics (more specifically the OWL-DL language) by considering background knowledge and inherent incompleteness from data, *i.e.*, under the OWA, through different gain function exploited in their algorithm. For the same purpose, `QuickFOIL` [ZPP14] has been proposed to tackle the scalability of the FOIL algorithm on large datasets. They proposed a generic top-down greedy algorithm, combining pruning heuristics and database optimizations, to discover function-free Horn clauses. However, it appears that `QuickFOIL` is unsuitable to mine Horn clauses under the OWA.

Several works in the literature consider the discovery of axioms (and particularly subsumption axioms) as a crucial task to enrich and/or align ontologies: Spiliopoulos et al. [SVV08] proposed a method for discovering subsumption relations among concepts from a supervised classification-based learning technique (part of ML techniques) using evidence in the training dataset. Lehmann proposed `DL-Learner` [Leh09], a framework exploiting top-down refinement approaches, reasoners, SPARQL queries and genetic programming to learn subclass axioms and definitions from different knowledge sources (OWL files, SPARQL endpoints, knowledge bases, ...). Bühmann and Lehmann extend the proposed method to learn OWL 2 axioms from large KGs [BL12].

We remark that the community widely explores ILP and ARM tasks to leverage ontology enrichment issues: Völker et Niepert [VN11] proposed the `Statistical Schema Induction` framework for the induction of schemas (axioms) from large RDF data graphs. By applying terminology acquisition on named classes, class expressions, object properties and property chains, they build transaction tables and perform an ARM task based on the `Apriori` algorithm to discover subsumption axioms composed of atomic classes, domain restriction axioms and range restriction axioms. To the best of our knowledge, they do not mine association rules under the OWA, but they use a confidence threshold to assess association rules instead. The same year, Fleischhacker et Völker suggested a set of inductive methods, both instance-based and schema-based methods,

including ARM and correlation techniques, to learn disjoint class axioms from large knowledge repositories [FV11]. Naturally, AMIE [GTHS13] (presented in Section 2.1.2) is a part of an ILP technique, able to mine logical rules (ARM) from knowledge graphs despite the absence of explicit counter-examples for candidate rules. They perform the mining tasks under the *Partial Completeness Assumption* (PCA) introduced to perform ARM tasks under the OWA, find counterexamples for rules, and estimate their quality with the PCA-confidence measure.

EDMAR [TDdNT17] is an evolutionary approach for discovering multi-relational association rules using OWL ontologies: the search strategy is supported by reasoners and evolutionary algorithm operators. The discovered association rules are expressed in *SWRL* (Semantic Web Rule Language): it is a syntax to express Horn-like rules that combines *OWL DL* and *OWL Lite* (sub languages of OWL) [HPSB+04] and can be integrated into ontologies. They assess individuals (*i.e.*, ARs) using a fitness function that considers facts (from ontologies) under the OWA.

2.2 SHACL Constraints Mining

SHACL (presented in Section 1.3.1.4) is a recent W3C recommendation (2017) which has motivated intensive research in both the academic and industrial domains for diverse purposes: *e.g.*, checking access policies [RIV23], expressing constraints against skills [KdSF21], construction scheduling constraints [KS19], ... Most research tends to focus on the SHACL semantics to describe its features [BJVdB22, PK22], methods to decide about shapes *containment* [LSR+20] whereas [ACO+20], [CRS18] and [PKM22] address issues on the semantics of *recursive* shapes (not defined yet).

SHACL engines are implemented on various semantic Web applications, *e.g.*, in Protégé [EL16], Corese [Cé23], Schimatos [WRMH+20] and Trav-SHACL [FRV21]. Interestingly, some research focuses on the usability of the SHACL validator for evaluating remotely accessible data graphs: [CFRS19b] and [FRV21] proposed SHACL engines for validating nodes against shapes (that can be recursive) through SPARQL endpoints enabling validation of remote RDF data graphs, *e.g.*, KGs from the LOD-Cloud.

Producing relevant SHACL shapes (expressing the domain's constraints) is one of the most challenging research areas that tackle data quality and integration issues. A recent state-of-the-art proposed by Rabbani et al. [RLH22] shows that most validating shapes are extracted manually. However, this method is not scalable for very large RDF data graphs or very complex RDF data, *e.g.*, medical data (requiring domain experts).

They discussed the usefulness and reliability of shapes extracted by (semi-)automatic approaches: most of the extracted shapes are limited to specific shapes and are not entirely reliable since spurious data support some shapes. Lieber et al. [LDV20] have studied statistics of widely used data shapes found on GitHub, and they shared the same conclusion about specification limit: most shapes implement cardinality, class, datatype and disjunction constraints, whereas very few shapes implement literal values constraints.

2.2.1 Extracting SHACL Shapes From Ontologies

Knublauch compared OWL and SHACL [Knu17], suggesting that both can be used successively to (1) infer new facts (*i.e.*, RDF triples) from existing ontology with an OWL inferencing engine and (2) assess these triples against defined shapes with a SHACL validator engine. Moreover, syntactic translation between OWL components to SHACL Core constraints is generally straightforward (see the Table presented in [Knu17]). Pandit et al. [POL18] argue that Ontology Design Pattern (ODP) axioms can be translated into SHACL shapes, suggesting that ODP axioms are more suitable than OWL axioms for capturing domain constraints. However, they do not propose an implementation of the proposed framework (*i.e.*, ODP axioms are manually translated into SHACL shapes), and the mapping between OWL axioms and SHACL shapes remains for future work.

Cimmino et al. proposed `Astrea` [CFG20], a framework for automatically generating SHACL shapes from ontologies: they convert ontology constraint patterns to SHACL constraints through mappings and SPARQL queries. In the same direction, Duan et al. proposed mappings to automatically translate XSD constraints for XML data sources in SHACL shapes [DCFD23]. More recently, `SCOOP` [DCFDD24] has been developed to merge existing approaches [CFG20, DDSMO+21, DCFD23] to automatically generate SHACL shapes from three RDF graph construction artefacts: OWL ontologies, XSD constraints and RML rules.

Ontology-based approaches are limited to the degree of coverage of the ontologies regarding the RDF data graph, which impacts the type of constraints that can be extracted: *e.g.*, `Astrea` does not scale to large ontologies [RLH22].

2.2.2 Extracting SHACL Shapes From Instances

Considering that knowledge graphs are rich in facts, instance-based approaches appear to address this limitation. However, one of the most significant challenges (in addition to the lack of comprehensive constraint coverage) in extracting SHACL shapes is scaling these methods to handle substantial data graphs.

The `shaclgen` library [Kee] has been developed in that sense: it is a Python library that automatically generates shape files from both RDF data graphs and ontologies. Despite its versatility, it appears difficult to assess the validity and usefulness of produced shapes, and `shaclgen` do not scale on large datasets and ontologies [RLH22].

Fernandez-Àlvarez et al. proposed `Shexer` [FÀLGGA22] (which is also a Python library) to automatically extract both SHACL and ShEx [PLGS14] shapes from RDF data. It takes an RDF dataset, target shapes and shape features as input to perform a mining strategy. They proposed a trustworthiness score to assess shapes, considering errors and incompleteness from RDF data. Moreover, they tackle the scalability issue on large RDF data graphs by optimizing machine memory consumption and limiting the number of instances during the mining process. Similarly, Rabbani et al. proposed the Quality Shapes Extraction (`QSE`) [RLH23a] approach, consisting of entity extraction, followed by the extraction of entity constraints for which support and confidence measures are computed. Finally, `QSE` extracts shapes from the most relevant constraints. The proposed method considers incompleteness and spurious data through shape pruning (based on their support and confidence measures). Moreover, they proposed a dynamic reservoir-sampling technique to store and process large RDF datasets. More recently, the same authors proposed `SHACTOR` [RLH23b], a GUI to exploit extracted shapes from the `QSE` algorithm to analyse data shapes validation against existing KGs.

A work very close to ILP techniques has been proposed by Omran et al. to learn Inverse Open Path (IOP) rules through the `SHACLearner` framework [OTRMH22]. The framework learns IOP rules by adapting an Open Path Rule Learner (OPRL) and filtering spurious rules using quality measures. However, `SHACLearner` is limited to extracting a specific type of rules: *e.g.*, they do not extract any constraints on RDF literals.

`ShapeDesigner` [BDFAG19] is a semi-automatic method to extract both SHACL and ShEx shapes that are not intended to be definitive: Users provide an RDF data graph in input, the system generates shapes based on pre-defined queries, then generated shapes can be modified through an interactive GUI. They address the scalability issues by limiting the number of query results.

Some of the research focuses on data profiling: Mihindukulasooriya et al. [MRR+18] proposed an approach inspired by existing ILP methods, as they induce SHACL shapes from data profiling information using ML algorithms. However, they limit their experiment to cardinality constraints; the others remain as future work. Interestingly, `ABSTAT` is a hybrid approach based on an ontology-driven data abstraction method to summarise datasets [SPP+16]; they extend this framework to transform obtained semantic profiles into SHACL shape graphs [SMP18]. More recently, Principe et al. extended this frame-

work: ABSTAT-HD [PMP+21] to address scalability issues (on ABSTAT) in profiling large knowledge graphs.

2.3 Grammatical Evolution

Relying on the Grammatical Evolution [OR01] (GE) presented in Section 1.3.2, the problem definition and the boundaries of the search space are designed with BNF grammar, and the GE operators are used to explore the search space. The versatility of this approach enables a very wide range of optimisation problems to be tackled [ROC18]: symbolic regression problems [OR04, LPC16b], code generation [CMRRI24], time series forecasting [RKCJ20], hyper-parameter optimisation on Deep Learning models (*e.g.*, CNN models) [VIK+22, VKR23], cryptography [RKV+22], etc. Many implementations of GE are available over different development environments: *e.g.*, GEVA in Java [OHG+11] (used in this thesis), PonyGE2 in Python [FMF+17] or gramEvol in R [NdSL16].

Some of the literature is focused on methods that extend standard GE techniques. O'Neill and Ryan proposed an extension of GE by introducing a co-evolution of the BNF grammar and the genetic code: "(GE^2)" [OR04]. They consider meta-grammar (*i.e.*, grammars' grammar) and optimal solution grammars to find (as well as the genetic code relying on these grammars) through the evolutionary process. They demonstrated the feasibility of (GE^2) over the study of symbolic regression problem instances. The same year, O'Neill et al. proposed a Position-Independent variation on GE: πGE [OBN+04], which impacts the genotype-phenotype mapping by extending the definition of a codon to become the pair (*nont*,*rule*) where *nont* is used in the genotype and *rule* selects which production rule should be used from *nont*. Thus, they remove the positional dependence observed during the derivation phasis in standard GE.

However, some recent work discusses GE limitations, in particular limitations on grammar design [DW22], their complexity and a "*poor*" initialisation of individuals [Har10]. Some contributions address these limitations by proposing general guidelines for grammar design [NA18], automated techniques for finding optimal GE hyper-parameters [AKNR21], or new methods for improving the population initialisation (*i.e.*, non-random initialisation), *e.g.*, in [NOB12]. We distinguish two issues from GE techniques: the *redundancy* between individuals, implying a decreasing diversity of solutions, and low *locality* [LFPC17, Med17], *i.e.*, "*how well-neighbouring genotypes correspond to neighbouring phenotypes*", which is a limitation in particular use cases [RO06].

Lourenço et al. address the locality issue by proposing an extension of GE called Structured Grammatical Evolution (SGE) [LPC16b, LPC16a], a novel genotypic representation

for GE that enables one-to-one mapping between genes and non-terminals belonging to the grammar. Consequently, changes do not affect the derivation options of other non-terminals, limiting the number of recursions during the derivation. In [LAP+18], they extend SGE by proposing a dynamic approach, *i.e.*, Dynamic SGE (DSGE), to limit the derivation during the mapping process, thus avoiding the whole grammar being pre-processed at the beginning. Some works suggest that probabilistic approaches address some criticisms in GE: Kim et Ahn proposed an extended GE relying on a Probabilistic Context-Free Grammar (PCFG) [KA15], where each production rule has a certain chance of being selected. Moreover, the population evolution is based on a probabilistic model relying on the relationship between production rules. Similarly, Mégane et al. proposed Probabilistic Grammatical Evolution (PGE) based on a PCFG. In contrast to [KA15], their experiments showed that PGE is significantly better than GE and comparable to SGE. More recently, Mégane et al. suggested (1) the Probabilistic Structured Grammatical Evolution (PSGE) [MLM22b], a framework that combines SGE and PGE and outperforms them, and (2) an extension of PSGE, *i.e.*, Co-evolutionary PSGE (Co-PSGE) [MLM22a] where the genotype and the grammar evolve through generations. As [MLM22b], Co-PSGE outperforms PGE and SGE.

Considering the discovery of new knowledge from RDF data, Nguyen and Tettamanzi proposed a method based on grammar-based genetic programming method for mining OWL class disjointness axioms composed of atomic classes [NT19c, NT19b] and non-atomic classes [NT20b, NT20d]. The framework relies on BNF grammars in input to build candidate class disjointness axioms dynamically through SPARQL queries. Moreover, they assess candidate axioms against RDF data based on generality and possibility measures (their computation relies on SPARQL queries) under the OWA [NT19c]. Interestingly, in [NT19b], they proposed a *deterministic crowding* method [M+92] as a survival selection to preserve an appropriate population diversity across generations. First, they perform a *genotypic* comparison between pairs of parents (p_1 and p_2) and their respective offspring pairs (o_1 and o_2) to estimate *distance* $\mathcal{D}(x, y)$ measures (where x and y are candidate class disjointness axiom genotypes). Second, they preserve the maximum value of the sum of the computed distances d_1 and d_2 such as $d_1 = \mathcal{D}(p_1, o_1) + \mathcal{D}(p_2, o_2)$ and $d_2 = \mathcal{D}(p_1, o_2) + \mathcal{D}(p_2, o_1)$. Finally, if $d_1 > d_2$, they compare the fitness $f(x)$ values of pairs (p_1, o_1) and (p_2, o_2) and return the best-suited individuals, *e.g.*, the pair (p_1,o_2) if and only if $f(p_1) > f(o_1)$ and $f(p_2) \leq f(o_2)$. Same process if $d_1 \leq d_2$. They address the scalability issue by proposing a method for extracting a reduced sub-graph, *i.e.*, a *training* dataset, of the whole RDF data graph: in [NT20b], they collect 1% of the RDF triples from DBpedia 2015-04 (English version) which contains $665,532,306$

RDF triples, resulting in a sub-graph of $6,739,240$ RDF triples used to discover class disjointness axioms through the evolutionary process.

2.4 Conclusion

In this literature review, we studied works related to the Ontology Learning research field in Section 2.1 and presented three sub-domains: linguistic techniques, statistical techniques and ILP techniques. The approach proposed in this thesis is one of the ILP techniques, where background knowledge is the fundamental basis for the induction of schemes that explain data. However, the exploration strategy, *i.e.*, the search of all possible solutions in the hypothesis space, differs from traditional methods in this domain [CD22].

Then, we studied existing work on SHACL constraint mining in Section 2.2: our approach fits in with the approaches for extracting SHACL shapes from RDF data, where *scalability* and *incompleteness/inconsistencies* issues are predominant and our main concerns in the scope of this work.

Finally, we studied existing work based on GE, where the flexibility of this technique makes it possible to answer a wide range of research questions in a multitude of domains, allowing it to be applied widely. Relying on the framework proposed by Nguyen and Tettamanzi and analysed in Section 2.3 (which is the most closely related work), the proposed deterministic crowding method tends to limit the algorithm's exploratory capabilities for two reasons: (1) the comparison between individuals is limited to a genotypic comparison instead of a phenotypic comparison (which appears to be more accurate), and (2) the selection of the best fitness score between parents-offspring makes it possible to preserve a good overall fitness score to the detriment of a broader exploration of the solution space (even if it means accepting less suitable individuals). This is why we propose a novel approach focusing on an expansive exploration of the solution space: we assume this research strategy uncovers a more comprehensive set of credible and relevant knowledge from RDF data graphs. Moreover, we propose novel optimisations related to SPARQL queries, process parallelization, etc., to consider using our approach with large RDF data graphs, *e.g.*, DBpedia.

CHAPTER 3

Evolutionary Discovery of Subsumption Axioms From RDF Data

3.1 Introduction

In this chapter, we focus on the ontology enrichment issue, especially the lack of axioms in ontologies, by proposing the discovery of `SubClassOf` subsumption axioms involving class expression (see Definition 3.1), providing domain knowledge from existing RDF data. The considered class expressions are extended to **complex** class expressions. We tackle the research question **RQ2**: "*How to automatically discover class subsumption axioms from RDF data?*"

Definition 3.1: SubClassOf axiom [BFH+12]

Let C and D be two class expressions. `SubClassOf(`C D`)` is *satisfied* if all the individuals x of the *subclass* C are also instances of the *superclass* D:

$$C(x) \subseteq D(x) \ ([a])$$

[a] https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Class_Expressions

Considering the *DBpedia 2015-04 ontology*[1] whose statistics are given in Table 3.1, we observe that there are only few OWL axioms explicitly included in the ontology. Although classes are well represented (*i.e.*, 1 subsumption axiom per class), each class is involved in at most 3 subsumption axioms.The `owl:DisjointWith` (disjointness class

[1] https://downloads.dbpedia.org/wiki-archive/Downloads2015-04.html

Table 3.1: DBpedia 2015-04 ontology and class expressions axioms (`owl:Thing` class is not considered)

Metrics	Value
#`owl:Class`	735
#`rdfs:subClassOf`	692
#`owl:equivalentClass`	356
#`owl:disjointWith`	24

axiom) is the most rarely represented class axiom: this motivated research to discover these axioms over RDF data using an implementation of Grammatical Evolution [NT19b, NT19c, NT20b, NT20d, NT20a] but these works did not consider subsumption axioms.

We started from an implementation of Grammatical Evolution proposed by Nguyen and Tettamanzi [NT19b] to extract disjointness axioms from an RDF data graph by extending the framework for discovering subsumption axioms composed of *complex* class expressions. We consider a possibilistic framework to assess candidate subsumption axioms over RDF data graph under the Open-World Assumption (OWA), considering **ignorance** about RDF facts [TFG17]. Tettamanzi et al. demonstrate that such a heuristic leads to high computation times, requiring time-restricted solutions [TFG15]. As a matter of fact, they tested $5,050$ subsumption axioms over the *DBpedia 3.9* RDF data graph (+400M RDF triples) by setting a time-cap at 20 minutes: only 12.51% of them do not exceed this time-cap. Consequently, we acknowledge that these performances do not allow the evolutionary discovery of these axioms from large RDF data graphs in reasonable computation time.

First, we address these limitations (**RQ1**) and propose an optimisation [FCFT22] consisting of three axes:

(A) **a multi-threading system** to parallelize axiom assessment,

(B) **an extension of the original heuristic to avoid redundant computation**, with an explanation of the computational problem,

(C) **an optimisation of SPARQL query chunking** relying on an extension of SPARQL federated query [CFG+21] to automatically iterate a federated query service call.

Second, we proposed a BNF grammar to build subsumption axioms over an RDF data graph: each individual represents *candidate* subsumption axiom ϕ where its OWL

functional-style syntax, *e.g.*, `SubClassOf(dbo:Cat dbo:Animal)` which states that "*all cats are animals*", represents the *phenotype*.

In Section 3.2, we introduce the background for these contributions. In Section 3.3, we present the optimisations and an analysis of the integrity of the obtained results and its impact in terms of CPU time saving, using the previous work [TFG15] as a benchmark. In Section 3.4, we discover candidate subsumption axioms over the *DBpedia 2015-04* RDF data graph using the proposed optimisations and an implementation of Grammatical Evolution discussed in Chapter 5. Finally, we conclude in Section 3.5.

3.2 Preliminaries

3.2.1 A Possibilistic Heuristic to Assess Subsumption Axioms

Possibility theory is a mathematical theory of epistemic uncertainty which uses the events, variables, ... denoted ω of a universe of discourse Ω ($\omega \in \Omega$) where each ω has a degree of possibility such that $\pi : \Omega \to [0, 1]$. $\pi(\omega) = 0$ means that ω is *impossible* and $\pi(\omega) = 1$ means that ω is *fully possible* [Zad99]. Let S a set of events from a universe of discourse Ω, *i.e.*, $S \subseteq \Omega$ and a possibility distribution π, the **possibility** (Π) measure is defined in Eq. (3.1). The **necessity** N (defined in Eq. (3.2)) measures the impossibility of its complement $\bar{S}$.

Equation 3.1: Possibility measure [Zad99]

$$\Pi(S) = \max_{\omega \in S} \pi(\omega), \ \Pi(S) \in [0, 1]$$

Equation 3.2: Necessity measure [Zad99]

$$N(S) = 1 - \Pi(\bar{S}) = \min_{\omega \in \bar{S}} \{1 - \pi(\omega)\}, \ N(S) \in [0, 1]$$

Here are some properties of the possibility and necessity measures:

empty set and universe of discourse: $\Pi(\emptyset) = N(\emptyset) = 0$, $\Pi(\Omega) = N(\Omega) = 1$

duality: $\Pi(S) = 1 - N(\bar{S})$

values: $N(S) > 0 \implies \Pi(S) = 1$, $\Pi(S) < 0 \implies N(S) = 0$

total ignorance on S: $\Pi(S) = \Pi(\bar{S}) = 1$

Tettamanzi et al. [TFG17] proposed a heuristic to assess the **possibility** and the **necessity** of an axiom ϕ. They consider v_ϕ^+, the number of confirmations observed among the RDF facts v_ϕ, the support of ϕ, and v_ϕ^-, the number of exceptions observed. They define the possibility $\Pi(\phi)$ in Eq. (3.3) and the necessity measure $N(\phi)$ in Eq. (3.4).

Remark 3.1

Under the OWA, RDF facts from v_ϕ can be neither a confirmation nor an exception to an axiom ϕ:

$$|v_\phi^+| + |v_\phi^-| \leq |v_\phi|$$

Equation 3.3: Possibility measure (Π) of an axiom ϕ [TFG17]

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

Equation 3.4: Necessity measure (N) of an axiom ϕ [TFG17]

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

To decide the acceptance of an axiom ϕ, Tettamanzi et al. proposed the Acceptance/Rejection Index criterion (**ARI**) defined in Eq. (3.5).

Equation 3.5: Acceptance/Rejection Index measure (ARI) of an axiom ϕ [TFG17]

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

The implementation of the above formulas was carried out in SPARQL. The queries presented in Fig. 3.1 and 3.2 return (respectively) the number of confirmations v_ϕ^+ and the number of exceptions v_ϕ^- for a given subsumption axiom $C \sqsubseteq D$. The confirmations

```
SELECT (COUNT(DISTINCT ?x) AS ?n) WHERE {
    ?x a <C>, <D> .
}
```

Figure 3.1: Counting confirmations $|\upsilon_\phi^+|$ of an axiom ϕ: `SubClassOf(<C> <D>)` in SPARQL

```
SELECT (COUNT(DISTINCT ?x) AS ?n) WHERE {
    ?x a <C>, ?t .
    FILTER NOT EXISTS { ?y a ?t, <D> . }
}
```

Figure 3.2: Counting exceptions $|\upsilon_\phi^-|$ of an subsumption axiom ϕ in SPARQL

cardinality computation is quite simple: we count the number of instances belonging to the subclass C and the superclass D.

The exceptions υ_ϕ^-, using the possibilistic heuristic, are instances of C and another class denoted T that does not share any instance with the superClass D (see Fig. 3.2). This SPARQL query, which gives of course still an approximation, even though a much finer one, of the actual number of true exceptions, turns out to be computationally quite expensive.

3.2.2 A Fitness Function for SubClassOf Axiom Assessment

To assess candidate subsumption axioms and define their fitness value, Nguyen and Tettamanzi proposed the following fitness function based on the possibility (Eq. (3.3)) and the necessity (Eq. (3.4)) measures. This function is defined in Eq. (3.6).

Equation 3.6: Fitness function for candidate axioms [NT19b]

$$f(\phi) = |\upsilon_\phi| \times \frac{\Pi(\phi)+N(\phi)}{2}$$

```
SELECT (COUNT(DISTINCT ?t) AS ?nic) WHERE {
    ?x a <C>, ?t .
}
```

Figure 3.3: SPARQL query used to compute the number of intersecting classes (`nic`) for a subclass C.

3.3 Optimizing the Computation of a Possibilistic Heuristic to Test Subsumption Axioms Against RDF Data

3.3.1 Multi-Threading System

We implemented a multi-threading system to parallelize the evaluation of the candidate axioms, which allows to significantly reduce the overall computation time. Fig. 3.4 presents the architecture of the multi-threading system. Its general operating principle is as follows: Let Φ a *stack* of candidate axioms (*i.e.*, $\Phi = \{\phi_i,\ i \in [1, n]\}$) to assess using SPARQL queries presented in Fig. 3.1, 3.2 and 3.3. While $\Phi \neq \emptyset$, the **threads manager** allocates the axioms $\phi \in \Phi$ to assess to each available threads t_i (*e.g.*, t_1 assess ϕ_1, t_2 assess ϕ_3, ...).

The higher the number of available CPU cores, the greter the execution time gain, since the program creates threads depending on the number of cores available on the machine on which the software is run. Nevertheless, while this optimization can reduce the latency of axiom evaluation if a large number of cores is available, it does not reduce the overall cost of the task.

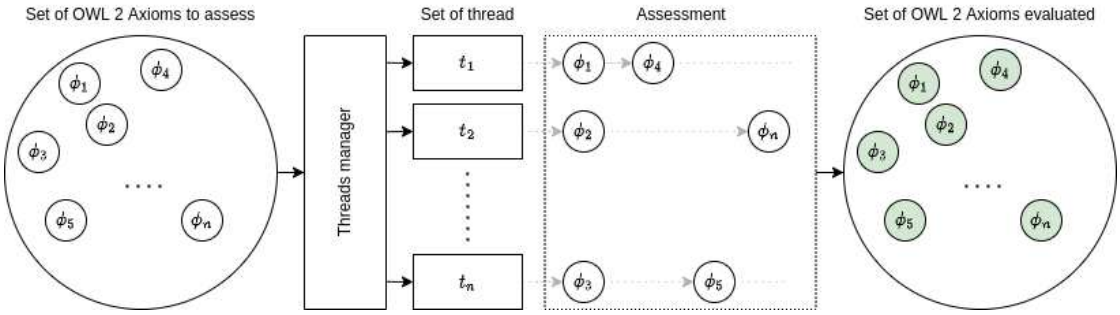


Figure 3.4: Overview of the multi-threading system

3.3.2 A Heuristic to Avoid Redundant Computation

Considering the implementation of the exceptions query presented in Fig. 3.2, we schematized it in Fig. 3.5 to highlight the possible and useless repetition of the same types `?t` (in red and yellow) for different instances `?x` of a subclass `<C>` satisfying a filter condition that does not depend on `?x`.

The same types are likely found many times for different individuals, implying the repetition of these same computations. Considering the RDF data graph *DBpedia 3.9*, these redundant computations are not negligible: Table 3.2 shows that the most represen-

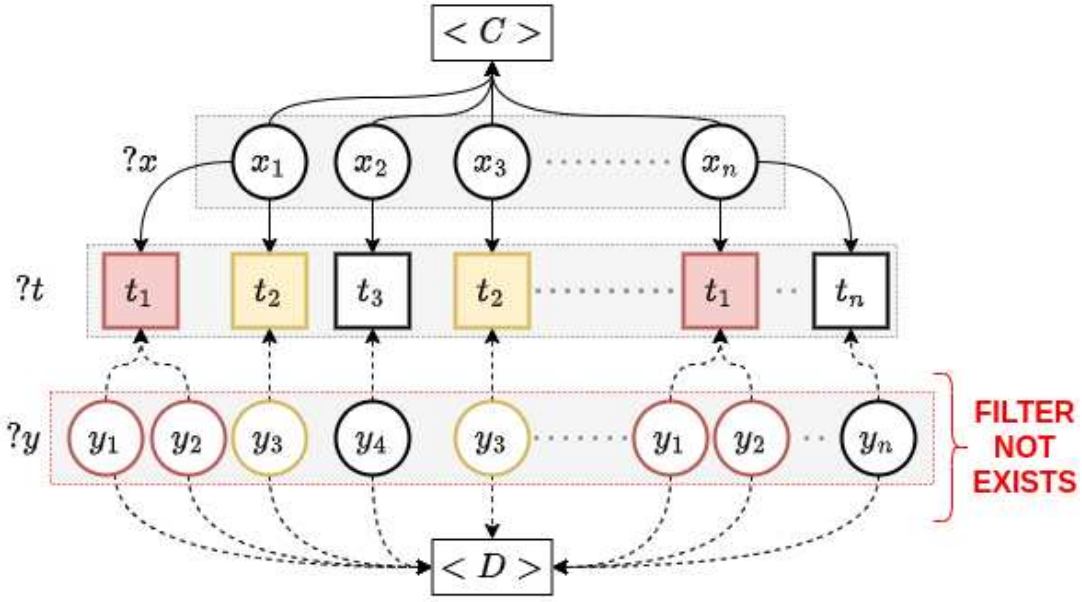


Figure 3.5: Overview of the redundant computation issue

Table 3.2: The 5 most representative concepts in the *DBpedia 3.9* RDF data graph

dbo concepts	# instances
dbo:Agent	1,472,369
dbo:Person	1,124,388
dbo:Place	754,415
dbo:CareerStation	577,196
dbo:PopulatedPlace	531,228

tative concepts in the *DBpedia 3.9* RDF dataset can count up to millions of individuals which can be computed more than once (?y layer in Fig. 3.5).

We propose to split the SPARQL query presented in Fig. 3.2 to compute the exceptions υ_{ϕ}^{-} in two phases:

1. Querying distinct types (*i.e.*, classes) assessed as potentially containing exceptions to an axiom (Fig. 3.6).

2. Querying instances that belong to both subclass <C> and **at least one** of the classes retrieved by the previous query, *i.e.*, exceptions of an axiom (Fig. 3.7).

However, the computational cost of FILTER NOT EXISTS grows more than linearly with the number of instances that have to be filtered. Therefore, we have also developed

```
SELECT DISTINCT ?t WHERE {
    {
        # We retrieve the other classes
        # of subclass <C> instances.
        SELECT ?t WHERE {
            SELECT DISTINCT ?t WHERE {
                ?x a <C> , ?t .
            } ORDER BY ?t
        } LIMIT $limit OFFSET $offset
    }
    # From these classes, we remove those
    # sharing instances with superclass <D>.
    FILTER NOT EXISTS {
        ?z a ?t, <D> .
    }
}
```

Figure 3.6: Retrieval of the classes `t` for which instances are possible exceptions of an axiom in SPARQL

```
SELECT DISTINCT ?x WHERE {
    ?x a <C>, ?t
    VALUES ?t { <t_1> <t_2> ... <t_n> }
} LIMIT $limit OFFSET $offset
```

Figure 3.7: Retrieval of the exceptions υ_ϕ^- of an axiom ϕ in SPARQL (using the classes t_i computed in Fig. 3.6)

```
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 . }
    }
}
```

Figure 3.8: Implementation of our optimized heuristic with a SPARQL federated query using parameters `loop` and `limit`: querying classes for which instances are possible exceptions to an axiom.

a chunking technique for SPARQL queries to split SPARQL queries into several steps using pagination (`LIMIT` ... `OFFSET`). We proposed an implementation of this approach in Algorithm 1.

3.3.3 Optimizing the Chunking of SPARQL Queries

In general, it is quite tedious to implement chunking of SPARQL queries. Moreover, one may still want to resort to chunking for a SPARQL query with a `VALUES` clause, since some servers limit the number of elements handled in such a clause. Consequently, we proposed a generic SPARQL operator to automatically integrate the pagination of the results with an iteration system, using URL parameters in SPARQL federated query services [CFG+21]. We set the following parameters: `loop=true` and `limit` to the chosen number of first results (denoted `$limit`) to be returned by a SPARQL query. First, this syntax makes it easier to code the iteration and chunking of SPARQL queries. Second, the iteration and chunking are delegated to the SPARQL engine [Cé23], assuming that this method is more efficient.

By using this novel `loop+page` operator, both the query to retrieve the classes potentially containing exceptions (Fig. 3.8) and the query to retrieve the exceptions (Fig. 3.9) are SPARQL federated queries using the parameters `loop` and `limit` in the URL of the remote query service in the `SERVICE` clause. The resulting algorithm is detailed in Algorithm 3.

3.3.4 Experiences

We conducted the scoring of 722 candidate axioms against the *DBpedia 3.9* RDF dataset comprising 463, 343, 966 RDF triples and 532 OWL classes. In previous works [TFG15],

Algorithm 1 Compute exceptions υ_{ϕ}^{-} to a `SubClassOf` axiom according to the contribution B.

Output: υ_{ϕ}^{-}
Require: $|\upsilon_{\phi}^{+}| \neq |\upsilon_{\phi}|$

$q_1 \leftarrow$ SPARQL query presented in Fig. 3.3
$q_2 \leftarrow$ SPARQL query presented in Fig. 3.6
$offset \leftarrow 0$
$limit \leftarrow 1000$
$\upsilon_{\phi}^{-} \leftarrow \{\}$
$types \leftarrow \{\}$
$nic \leftarrow eval(q_1)$
while $offset \neq nic$ **do**
    $q_2 \leftarrow q_2 +$ **LIMIT** $limit$ **OFFSET** $offset$
    $types \leftarrow types \cup eval(q_2)$
    $offset \leftarrow offset + min(nic - offset, limit)$
end while
$start \leftarrow 0$
$step \leftarrow 100$
$limit \leftarrow 10000$
while $start \neq |types|$ **do**
    $offset \leftarrow 0$
    $end \leftarrow start + min(step, |types|)$
    while true **do**
        $q_3 \leftarrow$ SPARQL query presented in Fig. 3.7
          using VALUES { $t_i \in types, i \in [start, end]$ }
        $q_3 \leftarrow q_3 +$ **LIMIT** $limit$ **OFFSET** $offset$
        $e \leftarrow eval(q_3)$
        $\upsilon_{\phi}^{-} \leftarrow \upsilon_{\phi}^{-} \cup e$
        if $|e| = limit$ **then**
            $offset \leftarrow offset + limit$
        else break
        end if
    end while
    $start \leftarrow start + min(|types| - start, step)$
end while
return υ_{ϕ}^{-}

Algorithm 2 Iterate and page a SPARQL query

$\mathcal{S} \leftarrow \{\}$
$q \leftarrow$ the body of a SPARQL query
for i=$start; i<=$until; i++ **do**
    $q \leftarrow q$+LOOP=$true&LIMIT=$limit&OFFSET=i*$limit
    $res \leftarrow$ eval(SERVICE url$\{q\}$)
    if $|res| == 0$ **then**
        break
    end if
    $\mathcal{S} \leftarrow \mathcal{S} \cup res$
end for
return $\mathcal{S}$

```
SELECT DISTINCT ?x WHERE {
    SERVICE <$url/sparql?loop=true&limit=$limit> {
        ?x a <C>, ?t VALUES ?t { <t1> <t2> ... <tn> }
    }
}
```

Figure 3.9: Implementation of our optimized heuristic with a SPARQL federated query using parameters `loop` and `limit`: querying exceptions v_ϕ^- of an axiom ϕ (using the classes t_i computed in Fig. 3.8).

Algorithm 3 Compute exceptions v_ϕ^- to a `SubClassOf` axiom using contributions B and C.

Output: v_ϕ^-
Require: $|v_\phi^+| \neq |v_\phi|$

$limit \leftarrow 1000$
$q_1 \leftarrow$ SPARQL query presented in Fig. 3.8
$types \leftarrow eval(q_1)$
$start \leftarrow 0$
$step \leftarrow 50$
$limit \leftarrow 10000$
while $start \neq |types|$ **do**
    $end \leftarrow start + min(step, |types|)$
    $q_2 \leftarrow$ SPARQL query presented in Fig. 3.9
      using VALUES { $t_i \in types, i \in [start, end]$ }
    $v_\phi^- \leftarrow v_\phi^- \cup eval(q_2)$
    $start \leftarrow start + min(|types| - start, step)$
end while
return v_ϕ^-

the computation times for assessing these axioms could take hours and sometimes days! First, we analysed the reliability of the results obtained: *are the evaluation results the same as those obtained in previous works?* [TFG15]. Finally, we discuss the computation time savings obtained.

The experiments were performed on a server equipped with an Intel(R) Xeon(R) CPU E5-2637 v2 processor at 3.50GHz clock speed, with 172 GB of RAM, 1 TB of disk space running under the Ubuntu 18.04.2 LTS 64-bit operating system.

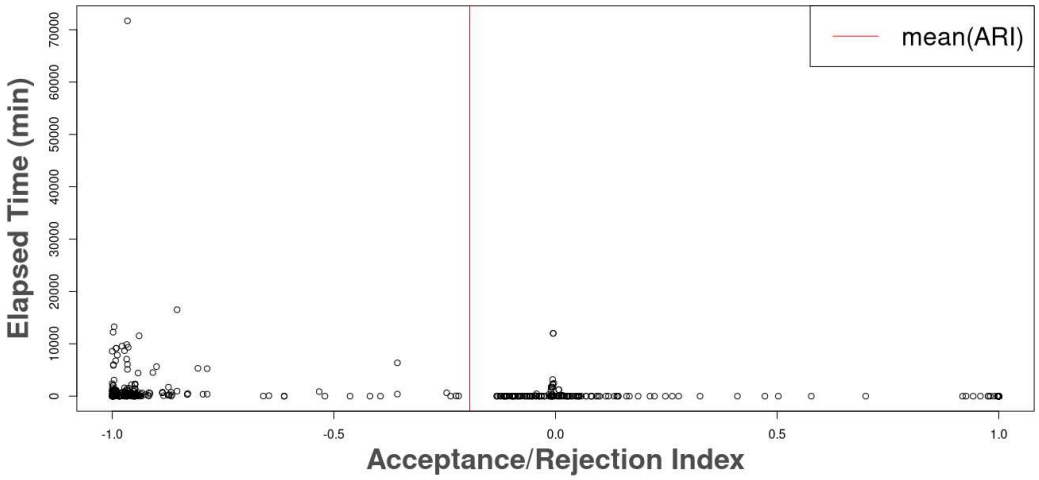
3.3.5 Results

Fig. 3.10 show that the computation time is significantly reduced, with a maximum computation time reduced from 71,699 to 489 minutes. The ARIs values computed for each axiom remain unchanged, giving the same average ARI value (~ -0.1936) and sharing the same conclusion: the number of accepted axioms ϕ, *i.e.*, $\phi \geq 1/3$, is only 197 against 525 rejected.

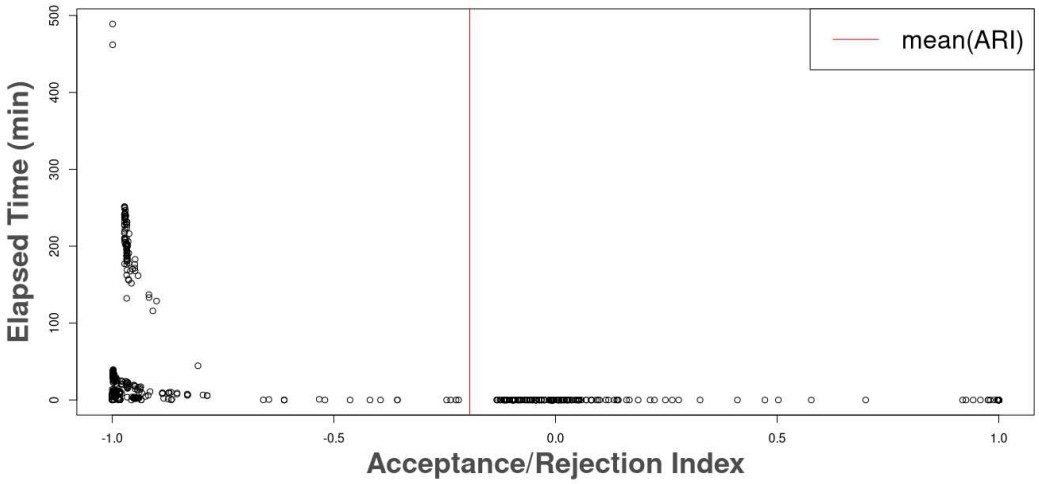
We compared the computation times of the ARI of each of the 722 candidate axioms using the original heuristic and our proposed optimization: Fig. 3.11 presents the initial computation time of the axioms ARIs using the original heuristic compared to the computation time using our proposed optimization. The average CPU computation time for evaluating an axiom is 30 minutes with our proposed optimization against 578 minutes with the original heuristic, with a significant average time saving of 548 minutes. For most axioms, we observe a lower computation time: 593 axioms are faster to assess using the optimisation (Algorithm 1), *i.e.*, 82% of the candidate axioms tested. This solves the problem of extremely long CPU computation times for some axioms: up to 71,699 minutes (!) using the original heuristic (see Fig. 3.10a) against 489 minutes, a reduction by a factor of ~ 150.

Some axioms involve the assessment of instances that do not have common types. Consequently, the execution of the initial *single* SPARQL query (Fig. 3.2) is faster than the execution of the *two* SPARQL queries in our optimisation. However, only 129 axioms are longer to assess with our optimised queries, increasing the average computation time by around 57 minutes and a maximum increase of 244 minutes: this represents a reasonable cost compared with the computation time saved.

To assess the contribution presented in Section 3.3.3 using a `loop+page` operator, we compared the computation times obtained against the previous one and present the results in Fig. 3.12: 94.6% of the axioms are assessed more quickly, reducing the average computation time by $\sim$ 12 minutes. This suggests that the implementation of this



(a) Original heuristic



(b) Results obtained with contributions A+B

Figure 3.10: Comparison of the ARI values of 722 axioms computed against *DBpedia 3.9* using the original heuristic against our contributions A+B

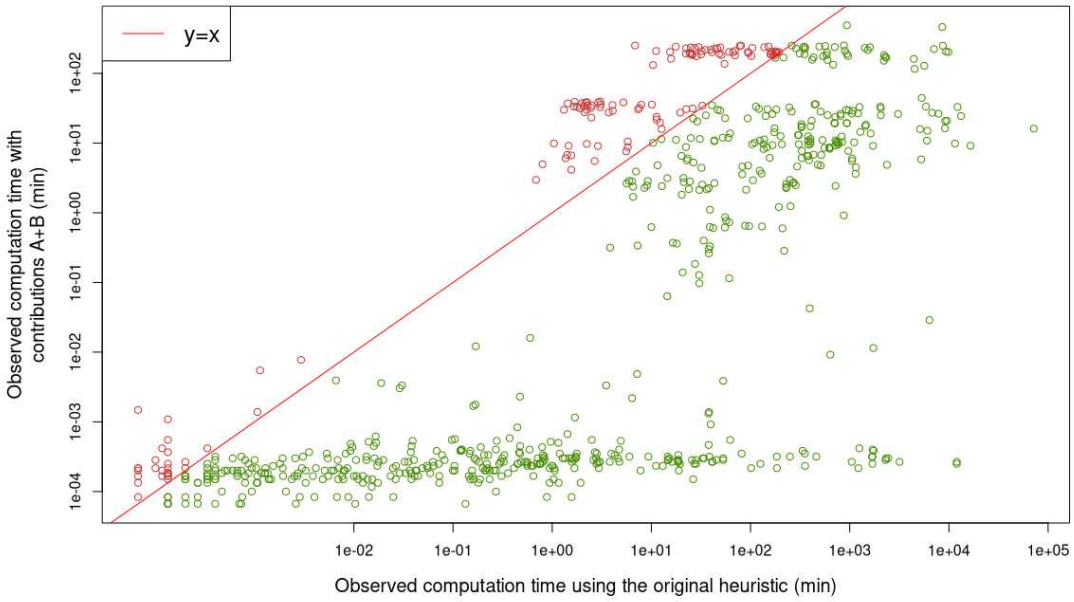


Figure 3.11: Comparison of the computation times (CPU) of axioms ARIs with the original heuristic and with our proposed optimization (A+B), highlighting the proportion of axioms for which our optimization saves time (in green) or loose time (in red). Both axes are logarithmic.

operator optimises the chunking of queries in the *Corese* semantic web factory [Cé23], making the computation cost less important than the chunking technique presented in the Algorithm 1.

3.4 Discovering Subsumption Axioms Involving Complex Class Expressions

The proposed optimisations tackle the *bottleneck* issues due to the approach itself: not scalable over the largest RDF data graph (*e.g.*, DBpedia) due to the previous implementation of the possibilistic heuristic to compute exceptions of candidate subsumption axioms. We are interested in the discovery of subsumption axioms composed of *complex* class expressions, assessing them over the proposed optimisation, more specifically with the Algorithm 1. We limit the scope of class expression to the **existential quantification**, **universal quantification** and the **intersection** of classes.

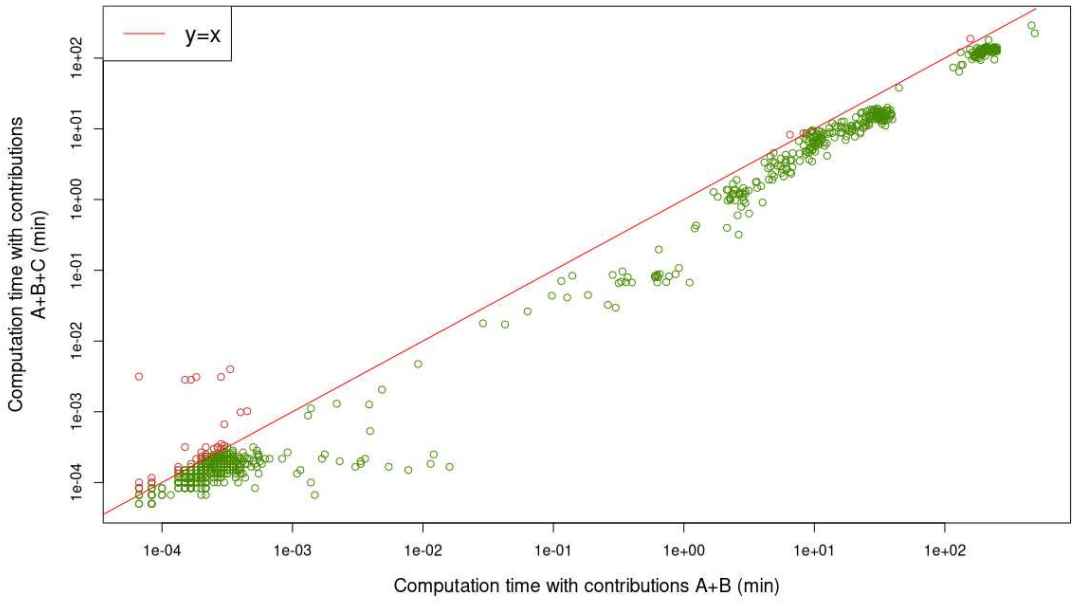


Figure 3.12: Comparison of the computation times (CPU) of axioms ARIs with our first (A+B) and second (A+B+C) proposed optimizations, highlighting the proportion of axioms for which our last optimization saves (in green) or loose (in red) time with A+B+C. Both axes are logarithmic.

Definition 3.2: Existential quantification [BFH+12]

Let OPE an *object property expression* and CE a *class expression*, `ObjectSomeValuesFrom(`OPE CE`)` defines all the individuals x that are connected by OPE to an individual y ($OPE(x, y)$) that is an instance of CE:

$$\{x \mid \exists y(OPE(x, y)) \wedge CE(y)\} \ ([a])$$

[a] https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Class_Expressions

Definition 3.3: Universal quantification [BFH+12]

Let OPE an *object property expression* and CE a *class expression*, `ObjectAllValuesFrom(`OPE CE`)` contains all the individuals x that are connected by OPE **only** to individuals y ($OPE(x, y)$) that are instances of CE:

$$\{x \mid \forall y(OPE(x, y)) \implies CE(y)\} \ ([a])$$

[a] https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Class_Expressions

Definition 3.4: Intersection of *two* class expressions [BFH+12]

Let CE_1 and CE_2 two class expressions, `ObjectIntersectionOf(`CE_1 CE_2`)` defines all the individuals x that are instances of CE_1 and CE_2:

$$\{x \mid CE_1(x) \wedge CE_2(x)\} \ ([a])$$

[a] https://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/#Class_Expressions

3.4.1 A BNF Grammar for *SubClassOf* Axioms

The purpose is to discover `SubClassOf` subsumption axioms between class expressions C and D, where C and D can represent an *atomic class*, an *existential quantification* class expression (Definition 3.2), an *universal quantification* class expression (Definition 3.3) or the *intersection of two atomic class* (Definition 3.4).

Nguyen and Tettamanzi proposed an implementation to extract instances of these class expressions in SPARQL [NT20d]. We reuse it for extracting *OWL classes* defined

```
SELECT DISTINCT ?Class WHERE {
    ?Class a owl:Class .
}
```

Figure 3.13: Extracting OWL classes from the *DBpedia 2015-04* ontology in SPARQL

```
SELECT DISTINCT ?Property WHERE {
    ?s ?Property ?o .
} LIMIT 1000
```

Figure 3.14: Extracting predicates from the *DBpedia 2015-04* RDF data graph in SPARQL

in the ontology and the *predicates* from the RDF data graph. The SPARQL query presented in Fig. 3.13 is used to extract classes and the SPARQL query presented in Fig. 3.14 is used to extract predicates. For the last one, we limit the number of query results to the first $1,000$ properties because of the large set of resources that can be retrieved with this query (which is time-consuming).

The BNF grammar to build `SubClassOf` subsumption axioms between complex class expressions is presented in Fig. 3.15.

3.4.2 Experiences

We consider a subset of the *DBpedia 2015-04* RDF data graph and its ontology to discover a large set of subsumption axioms. This dataset is composed of $124,698,021$ RDF triples, *i.e.*, 29.87% of the whole RDF data graph. These RDF data have been taken from a public repository [2] and focus on *instance types*, *same-as URI*, *infobox properties* (and their definitions) and *mapping-based properties*.

The Grammatical Evolution parameters are presented in Table 3.3. They are inspired by previous work on discovering subsumption axioms composed of complex class expressions [FT21]. We define the effort value to fairly compare the impact of the population size $|\mathcal{P}|$, renewing 60% of the whole population on each generation to support a wide exploration of the solution space. As some candidates can unavoidably result in very high computation time, we used the *time-cap* limit [TFG15], defining its value at 1 minute in order not to have a major impact on the computation time of the whole evolutionary process.

Since the implementation of Grammatical Evolution (and described in Chapter 5) differs from the algorithm used in previous work [FT21], we cannot fairly compare the

[2] https://downloads.dbpedia.org/wiki-archive/Downloads2015-04.html

```
<Axiom>           := <ClassAxiom>
<ClassAxiom>      := <SubClassOf>
<SubClassOf>      := "SubClassOf(" <ClassExpression> " "
                                   <ClassExpression> ")"
<ClassExpression> := <ObjectSomeValuesFrom> |
                     <ObjectAllValuesFrom>  |
                     <ObjectIntersectionOf> |
                     <Class>
# Class expressions :
<ObjectIntersectionOf> := "ObjectIntersectionOf("
                            <Class> " " <Class> ")"
<ObjectSomeValuesFrom> := "ObjectSomeValuesFrom("
                            <Property> " " <Class> ")"
<ObjectAllValuesFrom>  := "ObjectAllValuesFrom("
                            <Property> " " <Class> ")"
# Rules exploiting RDF nodes:
<Class>           := "SPARQL ?Class a owl:Class ."
<Property>        := "SPARQL ?s ?Property ?o ."
```

Figure 3.15: BNF grammar used to build candidate subsumption axioms composed of complex class expression

Parameter	Values
#run per settings	10
$\|\mathcal{P}\|$	100 ; 200 ; 500
Total effort E	$10,000$
|chromosome|	6
% Selection (elitism)	25%
Selection (recombination)	Tournament (60%)
Tournament size	25% of $\mathcal{P}$
Type crossover - P	Single point - 80%
Type mutation - P	Int flip - 5%
Time-cap	1 min.

Table 3.3: Parameters of Grammatical Evolution

obtained results with the previous ones. Moreover, previous work [FT21] has focused on the discovery of candidate `SubClassOf` axioms on a reduced RDF data graph of *DBpedia 2015-04* which contains 6,534,658 RDF triples (*i.e.*, 1.57% of the whole data graph) and the population is assessed over the whole RDF data graph at the last generation with a very low time-cap (30 seconds).

In addition to an in-depth analysis of the discovered subsumption axioms, we will discuss the computation time of the candidate subsumption axioms to assess the impact of the optimized computation of exceptions.

The experiments were performed on a server equipped with an Intel 11th Gen Core i7-11850H processor (16 **threads**), with 32 GB of RAM, 2 TB of disk space running under the Fedora Linux 35 operating system.

3.4.3 Results

Fig. 3.16 presents the discovered axioms according to their ARI value (see Eq. (3.5)) and the CPU time spent to assess them. It appears that the best average value of the ARI is observed for $|\mathcal{P}| = 100$, but it is not very significant compared to the other ARI values. Moreover, we observed a non-negligible set of axioms with a null ARI value (*i.e.*, total ignorance): 65.4% of the 995 distinct discovered axioms for $|\mathcal{P}| = 100$ against 47.5% (1,999 distinct axioms) for $|\mathcal{P}| = 200$ and 52.3% (4,976 distinct axioms) for $|\mathcal{P}| = 500$.

The evolution of the CPU time confirms that it is directly correlated to the number of exceptions: it is presented in Fig. 3.17. However, it appears to be negligible: we observe an average of 58.84 ms. to assess axioms for $|\mathcal{P}| = 100$ (with a maximum value of 5,297 ms.), 12.71 ms. for $|\mathcal{P}| = 200$ (max: 5,549 ms.) and 14.8 ms. for $|\mathcal{P}| = 500$ (max: 4,709 ms.). The time-cap limit appears unavoidable because of a high number of axioms that reach the limit on each execution: on average, 45 axioms for $|\mathcal{P}| = 100$ reach the time-cap against 27.6 axioms for $|\mathcal{P}| = 200$ and 58.3 for $|\mathcal{P}| = 500$. Surprisingly, one of the execution (with $|\mathcal{P}| = 100$) has led to 263 axioms that reach the time-cap, impacting the average value (without it, the average is about 20.78 axioms for $|\mathcal{P}| = 100$) as we can see on Fig.3.18a (the "light green" line, on the right).

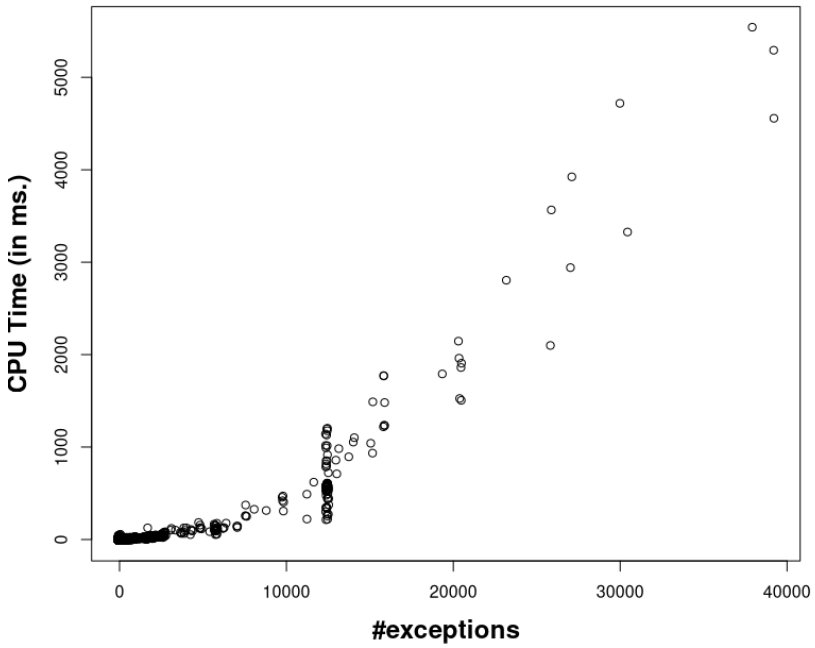


Figure 3.17: Number of exceptions $|\upsilon_{\phi}^{-}|$ of discovered subsumption axioms ϕ and the CPU time (in ms.)

The fitness evolution and the sum of CPU times are presented in Fig. 3.18 and highlight some trends: first, we observe that the "best" fitness evolution does not imply the "worst" evolution of the CPU times, which suggests that the higher the fitness value is, the lower the expected computation time be. Second, the fitness evolution gradually becomes less pronounced as the size of the population $|\mathcal{P}|$ increases but the sum of CPU times evolution suggests that a large population size leads to a wider exploration, involving the discovery of more or less credible candidate axioms.

Tettamanzi et al. [TFG17] suggest to accept subsumption axioms ϕ regarding its ARI value such as $ARI_{\phi} \geq 1/3$. Applying this acceptance criterion, we report the accepted discovered subsumption axioms in Table A.1 (The first part). They are composed of atomic classes, and 4 of them already exist in the *DBpedia 2015-04* Ontology. However, the following candidate axioms do not exist in the ontology and seem highly credible:

```
SubClassOf(dbo:Baronet dbo:Person)

SubClassOf(dbo:Historian dbo:Agent)
```

```
SubClassOf(dbo:ScreenWriter dbo:Agent)
```

All the discovered axioms that do not exist in the ontology are subject to a certain proportion of instances that we do not know if they are exceptions to these candidates. However, most of them seem to be consistent, *e.g.*, the following axiom:

```
SubClassOf(ObjectAllValuesFrom(dbo:artist dbo:TelevisionShow)
                               dbo:Work)
```

which states that "*instances where the artist is a television show are works*" is fully possible because no evidence contradicts this fact, but only 2 facts confirm this candidate axiom. This demonstrates that our approach effectively discovers subsumption axioms that express *subtle* relationships between domain concepts and RDF graph properties. The following candidate axiom:

```
SubClassOf(ObjectSomeValuesFrom(dbp:seasonTopscorer
      dbo:SoccerPlayer) dbo:SportsTeamMember)
```

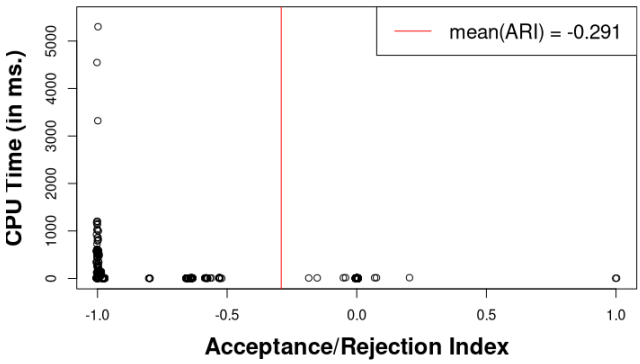
which states that "*instances that have at least one top scorer in the season who is a football player are members of a sports team.*" is widely contradicted by $1,534$ facts. This is inconsistent with the expectation that *a top scorer* is a *member of a sports team* and a sports team is a subset of a *sports structure*. The 37 confirmations of this candidate axiom are about their *end-of-season review*.

3.5 Conclusion

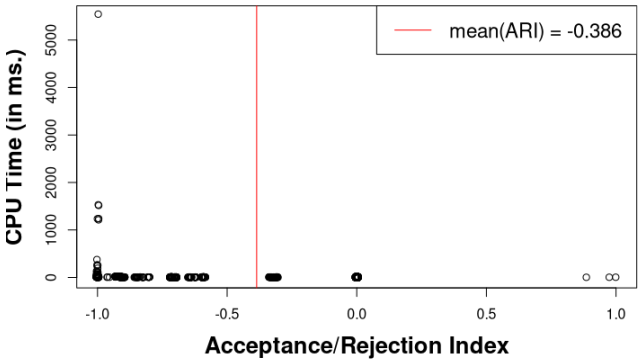
In this chapter, we tackle the computation time issues due to the possibilistic heuristic for subsumption axiom assessment by proposing some optimisations based on (1) a mult-threading system to assess axioms simultaneously, (2) a heuristic to avoid redundant computation for the candidate subsumption axiom assessment and (3) an optimisation of the federated query for SPARQL query chunking. The conducted experiments show that these contributions significantly reduce the computation time allocated for `SubClassOf` axiom assessment. Consequently, they opened up the perspectives of discovering candidate subsumption axioms using an adaptation of the Grammatical Evolution.

Secondly, we proposed a BNF grammar to build candidate subsumption axioms composed of complex class expressions. The evolutionary process has been carried out over a large RDF data graph (+100M RDF triples) using the Grammatical Evolution implementation and the proposed optimisations. The results show that this approach is

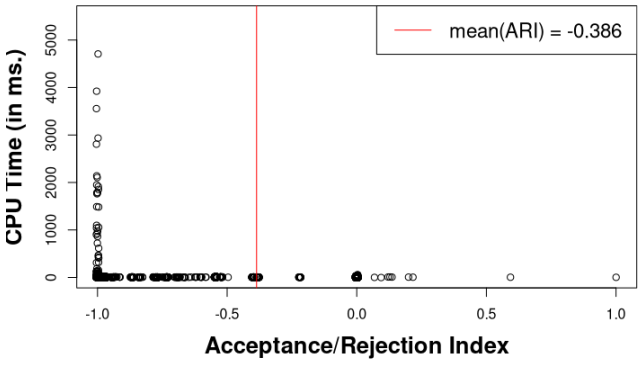
effective in discovering credible candidate subsumption axioms expressing subtle domain knowledge.



(a) $|P| = 100$

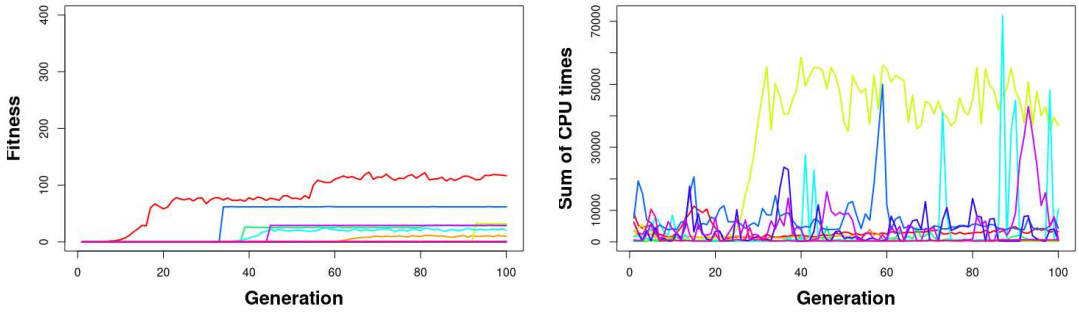


(b) $|P| = 200$

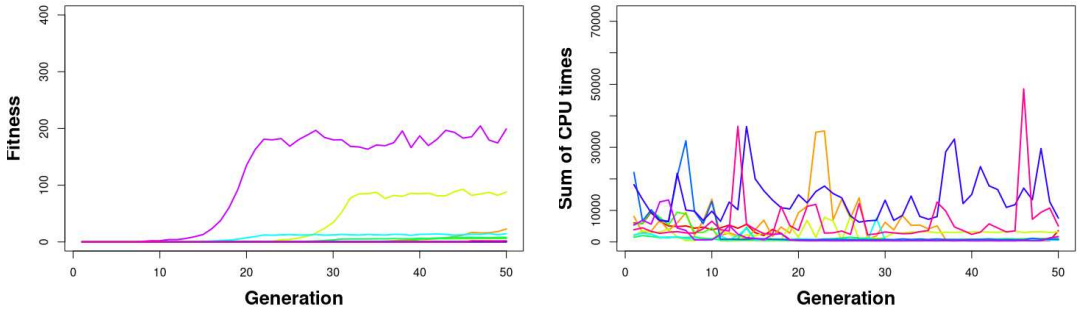


(c) $|P| = 500$

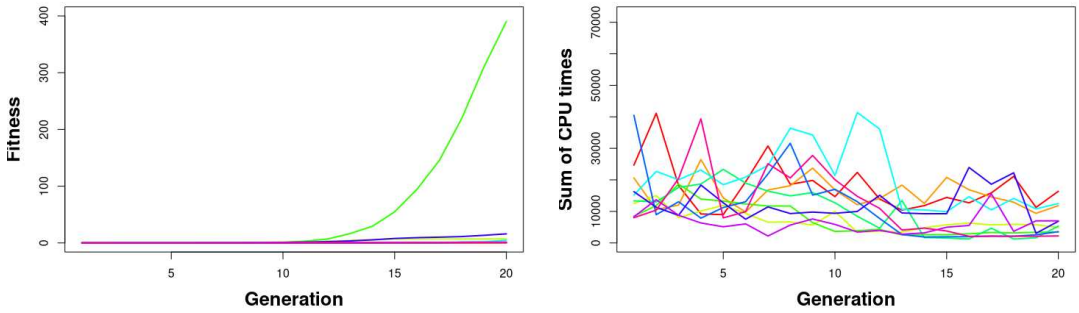
Figure 3.16: ARI values of the candidate axioms assessed with the optimized algorithm (Algorithm 1) against *DBpedia 2015-04*



(a) $|P| = 100$



(b) $|P| = 200$



(c) $|P| = 500$

Figure 3.18: Evolution of average fitness (on the left) and the sum of CPU times for the individual assessment (on the right) over 10 executions

Chapter 4

A Framework to Include and Exploit Probabilistic Information in SHACL validation Reports

4.1 Introduction

The notable growth of RDF data is driven by the development of automatic or semi-automatic techniques for extracting data from different sources (*e.g.*, *DBpedia* is built from *Wikipedia* data), constructing and enriching RDF data graphs. This dynamic has led to the need to control the RDF data quality, as it directly impacts its use by humans or artificial agents. We consider it essential to recognise that errors and inconsistencies are inherent in RDF data graphs.

In this chapter, we focus on **SHACL**: the language recommended by the W3C [1] to express **constraints** that RDF data must respect. SHACL shapes are instances of `sh:NodeShape` that target a specific set of nodes in an RDF data graph and assess them against a set of SHACL constraints. The shapes graph is used to search nodes in the RDF data graph that do not conform to the shapes through the SHACL validation. Therefore, SHACL addresses the requirements for RDF data quality control and helps reduce the inherent inconsistencies in RDF data graphs.

[1] https://www.w3.org/TR/shacl/

We observe that violations generated during a SHACL validation of a shape are a significant factor: any violation indicates that the whole validation report is not compliant. Considering a large collaborative RDF dataset with a massive and constant increase of RDF triples (*e.g.*, DBpedia), we assume that many RDF data violations against a set of shapes seem inevitable due to incomplete and/or incorrect data. Regarding the perspective of discovering relevant and representative SHACL shapes of an RDF data graph, we suggest that these errors (which are inherent) should be taken into account when validating an RDF data graph against candidate shapes.

Considering the state-of-the-art presented in Section 2.2, no contribution addresses the extension of the standard to include additional information in validation reports, considering a proportion of inherent errors. However, recent work considers the inconsistencies in RDF data graphs when automatically extracting different kinds of SHACL shapes. This work will be rigorously discussed in Chapter 5, which focuses on the evolutionary discovery of SHACL shapes.

Here, we tackle the following research question (**RQ3**): "*How to design a validation process considering physiological errors in real-life data?*" by proposing a framework based on a probabilistic model to consider a rate of violations, denoted p, which is assumed to be inherent in an RDF data graph, overcoming the '*binary*' nature of the validation process. Moreover, we extend the validation report to include probabilistic information considering the assumption that *the validation of RDF data follows a binomial distribution*, and we assess this assumption using hypothesis testing [FFT23a].

First, we present the probabilistic framework in Section 4.2: the probabilistic model of the validation process in Section 4.2.1, the extended validation report (and an extended vocabulary to express probabilistic results) in Section 4.2.2 and a method based on hypothesis testing to assess validation results in Section 4.2.3. Section 4.3 focuses on the conducted experiments, and the obtained results are discussed in Section 4.4. Finally, we conclude this chapter in Section 4.5.

4.2 A Probabilistic Framework for Shape Assessment

4.2.1 Probabilistic Model

We propose to extend the validation of RDF data against SHACL shapes by considering a physiological error proportion p in real-life RDF data (see Definition 4.1). We suggest that the mathematical modelling of the SHACL evaluation process, considering p, is based on a probabilistic model.

Definition 4.1: Physiological error in RDF data graphs

In a *real-life context*, RDF datasets are **imperfect** and **incomplete** (in the sense that expected data is missing). There are various reasons for this statement, *e.g.*, from the collaborative building of large RDF graphs (*e.g.*, Wikidata) or automatically constructed RDF graphs (*e.g.*, DBpedia) [FBMR17]. We define $p \in [0, 1]$ as the physiological error proportion of an RDF data graph.

Let υ an RDF data graph, $\mathcal{S}$ the SHACL shapes graph and s a shape contained in this graph ($s \in \mathcal{S}$)

Definition 4.2: Support (or reference cardinality) of a SHACL shape

The cardinality (or support) of a shape s, denoted υ_s, is the set of RDF triples in the RDF data graph υ targeted by s and tested during the validation. We define its cardinality $|\upsilon_s|$ as the **reference cardinality**.

Definition 4.3: Confirmation(s) and violation(s) of a SHACL shape

The confirmations denoted υ_s^+, and violations denoted υ_s^- of a shape s are the disjoint sets (*i.e.*, $\upsilon_s^+ \cap \upsilon_s^- = \emptyset$) that correspond, respectively, to the RDF triples t ($t \in \upsilon$) that are **consistent** with s and those that **violate** s (*i.e.*, inconsistent):

$$\forall t \in \upsilon_s^+, \quad t \wedge s \nvDash \bot$$

$$\forall t \in \upsilon_s^-, \quad t \wedge s \models \bot$$

The confirmations and the violations of a shape s (Definition 4.3) compose the set of RDF triples targeted by s:

$$\upsilon_s = \upsilon_s^+ \cup \upsilon_s^-$$

Remark 4.1

We consider **triples** instead of *nodes* to ensure the consistencies of the reference cardinality definition (Definition 4.2) for shapes containing more than 1 constraint: *i.e.*, one node is assessed against all constraints and can involve more than one violation.

The modelling based on the SHACL validation process is defined as follows: let X a **random variable** which conceptualises a set of observations (*i.e.*, RDF triples) from the SHACL validation of a shape s. A single tested triple $t \in \upsilon_s$ for which the SHACL validation defines whether t is consistent. Considering the assumption that a SHACL validation is a set of n experiences (and so $n = \upsilon_s$) and p the physiological error proportion, an RDF triple t that violates s is a **success** of the considered experience, otherwise it is a **failure**:

$$\forall t \in \upsilon_s, \quad t \wedge s \models \bot \implies X = 1$$

$$\forall t \in \upsilon_s, \quad t \wedge s \not\models \bot \implies X = 0$$

Consequently, applying a **Bernoulli distribution** seems an intuitive idea for computing the probability of a triple t being a success or a failure when validated against a shape s considering a physiological error proportion p:

$$x \in \{0, 1\},\ \mathbb{P}(X = x) = p^x (1 - p)^{(1-x)}$$

Considering the SHACL validation of RDF triples targeted by a shape s *i.e.*, υ_s, the **Binomial distribution** models this probabilistic approach. Let assume n experiences, p the probability of success and $X \sim B(n, p)$, the probability to obtain k success among n experiences is:

$$\forall k \in \{0, 1, \ldots, n\},\ \mathbb{P}(X = k) = \binom{n}{k} \cdot p^k \cdot (1 - p)^{n-k}$$

Considering the SHACL validation of a shape s, $X \sim B(|\upsilon_s|, p)$ for which we define the **likelihood measure** L_k as the plausibility to obtain exactly k violations is presented in Definition 4.4.

Definition 4.4: The likelihood of a shape

The likelihood to observe a number of violations $|\upsilon_s^-|$ among the RDF triples targeted by a shape s, i.e. $|\upsilon_s|$, considering $X \sim B(|\upsilon_s|, p)$ is:

$$L_{|\upsilon_s^-|} = \mathbb{P}(X = |\upsilon_s^-|) = \binom{|\upsilon_s|}{|\upsilon_s^-|} \cdot p^{|\upsilon_s^-|} \cdot (1 - p)^{|\upsilon_s^+|},\ L_{|\upsilon_s^-|} \in [0, 1]$$

```
[ a sh:ValidationReport ;
    sh:conforms boolean ;
    [...]
    # Probabilistic SHACL extension
    psh:summary [
        a psh:ValidationSummary ;
        psh:referenceCardinality |v_s| ;
        psh:numConfirmation |v_s^+| ;
        psh:numViolation |v_s^-| ;
        psh:generality G(s) ;
        psh:likelihood L_{|v_s^-|} ;
        psh:focusShape s
    ] ;
] .
```

Figure 4.1: Structure of the extended SHACL validation report of a shape s

4.2.2 Extension of the SHACL Validation Report Model

We propose an enriched model of the SHACL validation report to express additional information for each shape considered in the report. We defined an extension to the SHACL validation report vocabulary denoted by prefix `psh`.[2]

As a SHACL validation report considers the conformity of an RDF data graph against a shapes graph $\mathcal{S}$ and possibly $|\mathcal{S}| > 1$, we define the **focus shape** property presented in Definition 4.5. For each source shape s considered in the validation of an RDF data graph, we generate a `psh:summary` property that links the validation report to a blank node of type `psh:ValidationSummary`. This blank node is the subject of several properties whose values result from probabilistic metrics.

Definition 4.5: Focus shape property

The focus shape is the value of property `psh:focusShape`. It is **the source shape s of the validation result** further described in the validation summary.

Regarding the assessment of a shape s, the blank node of type `psh:ValidationSummary` includes the following properties:

- the **reference cardinality** $|v_s|$ (Definition 4.2) is the value of the property `psh:referenceCardinality`

[2] `prefix psh: <http://ns.inria.fr/probabilistic-shacl/>`

– the **number of confirmations** $|\upsilon_s^+|$ and the **number of violations** $|\upsilon_s^-|$ (Definition 4.3) are the values of properties `psh:numConfirmation` and `psh:numViolation`

– the **likelihood** $L_{|\upsilon_s^-|}$ (Definition 4.4) is the value of property `psh:likelihood`

– the **generality** $G(s)$ (Definition 4.6) is the value of property `psh:generality`

Fig. 4.1 presents the structure of the extended SHACL validation report, *i.e.*, how the extension works with other standard information in the validation report.

Definition 4.6: The generality of a shape

The **generality** $G(s)$ measures the *representativeness* of a shape s, *i.e.*, the number of RDF triples targeted by a shape $|\upsilon_s|$ divided by the number of triples in the RDF data graph $|\upsilon|$:

$$G(s) = \frac{|\upsilon_s|}{|\upsilon|},\ G(s) \in [0, 1]$$

An extended example of an extended SHACL validation report is presented in the Appendix (Fig. B.1): it considers the validation of a shape s_1 (represented by the IRI `:s1`) against an RDF data graph υ where $|\upsilon| = 1,000$ under the assumption that the physiological error proportion of υ is 10%, *i.e.*, $p = 0.1$. We assume $\upsilon_{s_1} = 200$ and 22 violations against the shape (*i.e.*, $|\upsilon_{s_1}^-| = 22$ and so $|\upsilon_{s_1}^+| = 178$). Consequently, the extended SHACL validation computes the likelihood metric:

$$L_{|\upsilon_{s_1}^-|} = \mathbb{P}(X = 22) = \binom{200}{22} \cdot 0.1^{22} \cdot (0.9)^{178} \approx 0.081$$

At the same time, it defines the generality value:

$$G(s_1) = \frac{200}{1000} = 0.2$$

The proposed ontology, describing the extended SHACL validation report vocabulary, has been published in Linked Open Vocabulary [3] and the documentation is available here. [4]

4.2.3 Data Graph Validation Against a Shape as a Hypothesis Test

The previous section relies on a probabilistic model of the standard SHACL validation as the assumption that the validation follows a binomial distribution, *i.e.*, $X \sim B(|\upsilon_s|, p)$.

[3] https://lov.linkeddata.es/dataset/lov/vocabs/psh
[4] https://ns.inria.fr/probabilistic-shacl/

However, this assumption on the SHACL validation, implying the estimation of the physiological error proportion p in an RDF data graph (possibly defined empirically), must be assessed to ensure the consistency of our assumption (as it can lead to incorrect conclusions). We propose an approach based on *hypothesis testing*, more specifically the **testing for Goodness of Fit**, to assess the obtained validation results of a shape s against the physiological error proportion estimation. Consequently, we define the **null hypothesis** H_0: "*the RDF data υ_s follow a binomial distribution with the given error rate*" and the **alternate hypothesis** H_1: "*the RDF data υ_s do not follow a binomial distribution*":

$$H_0 : \ X \sim B(|\upsilon_s|, p)$$

$$H_1 : \ X \nsim B(|\upsilon_s|, p)$$

Let $\hat{p_s}$ be the observed proportion of violations from the SHACL validation of a shape s, *i.e.*, $\hat{p_s} = \frac{|\upsilon_s^-|}{|\upsilon_s|}$. We assess the proportion $\hat{p_s}$ against the estimated physiological error proportion p and we propose to **accept** H_0 if the observed error proportion of a shape $\hat{p_s}$ is *lower than or equal* to the physiological error proportion,

$$\hat{p_s} \leq p \implies \upsilon \models s$$

Regarding the case for which *the proportion of violations observed* $\hat{p_s}$ is higher than the estimation, we are interested in the significance of this gap: *is the difference significant enough to reject the null hypothesis H_0?*

The testing of Goodness of Fit is defined as follows: let X_s^2 the **test statistic** for a shape s which follows a **Chi-square distribution** assuming H_0, i.e. $X_s^2 \sim \chi^2_{k-1,\alpha}$ with $k-1$ degrees of freedom and a level of significance $1-\alpha$. This test is performed at the α level of significance defined at 5%. It considers k the total number of groups, i.e. $k = 2$, n_i the observed number of individuals for each group and T_i the theoretical number of individuals for each group:

$$X^2 = \sum_{i=1}^{k} \frac{(n_i - T_i)^2}{T_i} \sim \chi^2_{k-1;\alpha}$$

Let n_1 be the observed number of violations $|\upsilon_s^-|$, T_1 the theoretical number of violations denoted $|\hat{\upsilon_s^-}|$, n_2 the observed number of confirmations $|\upsilon_s^+|$ and T_2 the theoretical number of confirmations denoted $|\hat{\upsilon_s^+}|$ where the T_i values strictly depend on the reference cardinality of a shape and the physiological error rate:

$$T_1 = |\hat{\upsilon_s^-}| = p \times |\upsilon_s|$$

$$T_2 = |\hat{v_s^+}| = (1-p) \times |v_s|$$

Finally, we define X_s^2 the test statistic of a shape s in Eq. (4.1).

Equation 4.1: Test statistic of a shape

$$X_s^2 = \sum_{i=1}^{k} \frac{(n_i - T_i)^2}{T_i} = \frac{|v_s^-| - |\hat{v_s^-}|}{|\hat{v_s^-}|} + \frac{|v_s^+| - |\hat{v_s^+}|}{|\hat{v_s^+}|}$$

Remark 4.2

The testing for Goodness of Fit (Formula (4.1)) is **applicable** if and only if:

$$\forall i \in [1,k], T_i \geq 5$$

$$i.e., |\hat{v_s^-}| \geq 5 \textbf{ and } |\hat{v_s^+}| \geq 5$$

The **critical region**, i.e. the rejection region of H_0, is defined by the value $\chi^2_{k-1;\alpha}$. Considering $\alpha = 0.05$ and $k = 2$, we define the critical value:

$$\chi^2_{k-1;\alpha} = \chi^2_{1;\alpha=0.05} = 3.84$$

Let I_a the acceptance interval of a χ^2 distribution with $k = 2$ and $\alpha = 0.05$, i.e. $I_a = [0, \chi^2_{k-1;\alpha}] = [0, 3.84]$ which accepts H_0 if $X_s^2 \in I_a$ (or $X_s^2 \leq \chi^2_{k-1;\alpha}$), the **acceptance** of H_0 implies the acceptance of a shape. The criteria are presented in Definition 4.7.

Definition 4.7: Acceptance of a shape

The acceptance of H_0, *i.e.*, $X \sim B(|v_s|, p)$, relies on the observed violations proportion $\hat{p_s}$ and the test statistic value X_s^2 (*i.e.*, $I_a \in [0, 3.84]$):

$$\hat{p_s} \leq p \textbf{ or } X_s^2 \in I_a \implies v \models s$$

If the Remark 4.2 is not satisfied, the testing for Goodness of Fit cannot be established and $\hat{p_s}$ is the only criterion used:

$$\hat{p_s} \leq p \implies v \models s$$

$$\hat{p_s} > p \implies v \not\models s$$

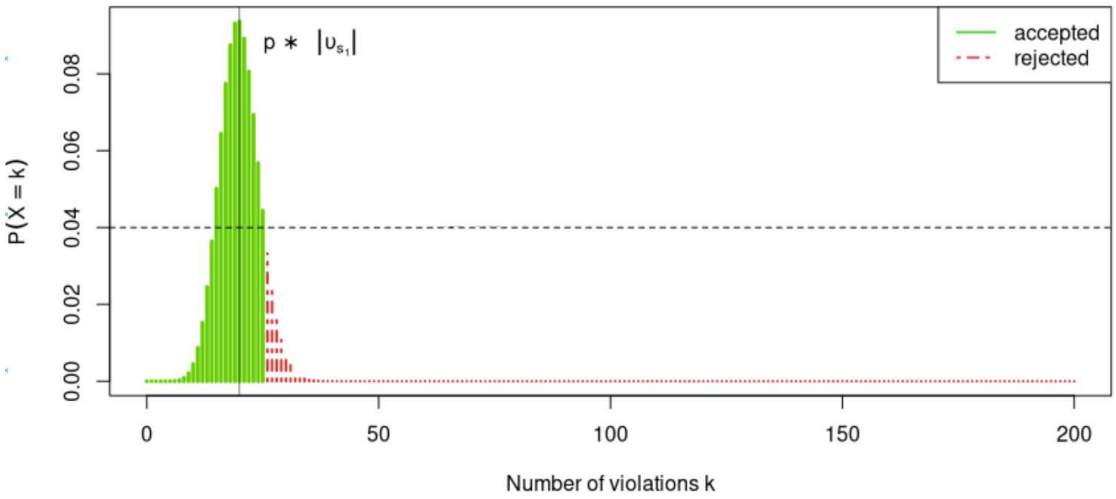


Figure 4.2: Acceptance zone of shape s_1, considering $X \sim B(|\upsilon_{s_1}|, p)$ where $|\upsilon_{s_1}| = 200$ and $p = 0.1$.

The Fig. 4.2 presents a practical example of the acceptance zone of the SHACL shape s_1 presented in Fig. B.1: we observe that the observed proportion of violations is *slightly higher* than the physiological error proportion, *i.e.*, $\hat{p_{s_1}} = \frac{|\upsilon_{s_1}^-|}{|\upsilon_{s_1}|} = 0.11$ and so $\hat{p_{s_1}} > p$. The testing for Goodness of Fit is applicable for s_1 because $|\hat{\upsilon_{s_1}^-}| = 0.1 \times 200 = 20 \geq 5$ and $|\hat{\upsilon_{s_1}^+}| = 0.9 \times 200 = 180 \geq 5$ (see Remark 4.2).

Let $X^2_{s_1}$ be the test statistic of shape s_1. We accept H_0 (and accept the shape) if $X^2_{s_1} \in I_a$ (with $\alpha = 0.05$), we reject it otherwise:

$$X^2_{s_1} = \frac{(22-20)^2}{20} + \frac{(178-180)^2}{180} = \frac{4}{20} + \frac{4}{180} \approx 0.222$$

The test statistic demonstrated that $X^2_{s_1} \leq 3.84$ and so $X^2_{s_1} \in I_a$. Consequently, we **accept** H_0 with a level of significance of 95% and we accept the shape s_1.

4.3 Experiments

The probabilistic framework assumes that the RDF triples tested during the validation of a shape follow a binomial distribution. Hypothesis testing (*i.e.*, testing for Goodness of Fit) validates the consistency of these assumptions. At the same time, we are exploring whether this approach can capture the knowledge domain more comprehensively, *i.e.*, a wider range of accepted shapes that are consistent despite observed violations from validation reports. Considering a shape graph representative of an RDF dataset, we are

proposing the search for a physiological error proportion p for which it is reasonable to consider the acceptance of shapes on a subset of the global RDF dataset.

We conducted the experiments on a subset of the *Covid-on-the-Web RDF dataset*[5] [MGAK+20] against **a set of 377 SHACL shapes** obtained from a translation of the experimental results of Cadorel et al. [CT20] which are considered as **representative** shapes of the global *Covid-on-the-Web* dataset.

We run the probabilistic SHACL validation engine implemented in the *Corese* semantic web factory [Cé23] to assess the shapes graph against the considered RDF data subgraph, conducting an analysis of the theoretical error rate to find an optimal rate. We assume the values of p empirically such that $p \in \{0.05, 0.1, 0.15, \ldots, 0.95, 1\}$, *i.e.*, 20 values for p to be tested.

The experiments were performed on a Dell Precision 3561 equipped with an Intel(R) 11th Gen Core i7-11850H processor, with 32 GB of RAM running under the Fedora Linux 35 operating system. The source code is available in a public repository.[6]

4.3.1 *Covid-on-the-Web* Dataset

Covid-on-the-Web is an RDF knowledge graphs produced from *COVID-19 Open Research Dataset (CORD-19)*. It contains $1,361,451,364$ RDF triples relying on $111,256$ **scientific articles**, described by URIs and named entities (**NE**) identified in these articles, disambiguated by *Entity-Fishing* and linked to *Wikidata* [MGAK+20].

Table 4.1: Summary of the *Covid-on-the-Web* RDF subgraph.

$\|v\|$	**#distinct articles**	**#distinct NE**	**avg. #NE per article**
$226,647$	$20,912$	$6,331$	10.52

We exploited the results obtained in previous works [CT20] by extracting articles URIs, their related named entities and labels: scientific articles are associated with their NE by the predicate `rdf:type`. Fig. 4.3 shows a subset of RDF triples contained in the subgraph (in *turtle* format), and the characteristics of the RDF dataset are presented in Table 4.1. The RDF dataset contains 18.79% of the global set of scientific articles and 0.01% of the global set of named entities.

[5] https://github.com/Wimmics/CovidOnTheWeb
[6] https://github.com/RemiFELIN/RDFMining/tree/eswc_2023

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix covid:  <http://ns.inria.fr/covid19/> .
@prefix entity:  <http://www.wikidata.org/entity/> .

# Scientific articles
covid:ec1...2c5   rdf:type    entity:Q4407  .
covid:fff...86d   rdf:type    entity:Q10876 .
[...]
# Labels of named entities
entity:Q4407      rdfs:label  "methyl"@en .
entity:Q10876     rdfs:label  "bacteria"@en .
```

Figure 4.3: Example of RDF data extracted from *Covid-on-the-Web* in *turtle* format.

4.3.2 Shapes Graph

From the experimental results of Cadorel et al. (*i.e.*, association rules that we consider as *representative* of the whole RDF dataset), we extracted the named entities corresponding to the *antecedent* and the *consequent* labels of these association rules. We have carried out a treatment[7] allowing the conversion of these rules into SHACL shapes. First, we target scientific articles belonging to an NE, representing the *antecedent*, with the property `sh:targetClass`. Among the targeted articles, we assessed their affiliation to another NE, representing the *consequent*: We apply a constraint to the article type by targeting an NE using the property `sh:hasValue`. Any violation will invoke a `sh:HasValueConstraintComponent` violation. Figure 4.4 presents an example of a used shape.

4.4 Results

The global results (*i.e.*, non-dependant of the physiological error proportion p) are presented in Table 4.2. We observe that the average number of triples tested during the validation of the subgraph against the shapes graph is very low, impacting the average generality value: it represents 0.05% of the whole graph. Moreover, 68.9% of tested triples are violations: it supports the assumption that the sub-graph is incomplete and highlights the interest of a probabilistic evaluation of the RDF data by varying p error rates and understanding what we can consider a reasonable error rate.

[7] The treatment is extensively detailed in the public repository:
https://github.com/RemiFELIN/RDFMining/tree/eswc_2023/AR-SHACL

```
@prefix : <http://www.example.com/myDataGraph#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix entity: <http://www.wikidata.org/entity/> .

:1 a sh:NodeShape ;
    sh:targetClass  entity:Q10295810 ;
    sh:property [
        sh:path rdf:type ;
        sh:hasValue entity:Q43656 ;
    ] .
```

Figure 4.4: Example SHACL shape representing an association rule with `entity:Q10295810` (`"hypocholesterolemia"@en`) as an *antecedent* and `entity:Q43656` (`"cholesterol"@en`) as a *consequent*.

Table 4.2: Results obtained from the probabilistic validation report

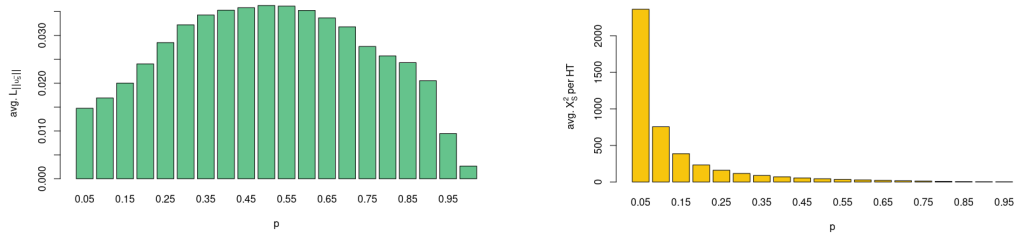
Metric	Value
avg. $\lvert\upsilon_s\rvert$	106.69
avg. $\lvert\upsilon_s^+\rvert$	33.19
avg. $\lvert\upsilon_s^-\rvert$	73.50

As the likelihood and the test statistic values depend on the physiological error proportion p, they are represented, respectively, in Fig. 4.5a and Fig. 4.5b. Intuitively, the evolution of the likelihood value suggests a "*bell-shaped curve*" for which the maximal average value (0.036%) is obtained with a physiological error proportion $p = 0.5$ (50%), which appears to be the most reasonable error rate.

Fig. 4.6 shows the decisions made regarding shapes (acceptance or rejection) based on the theoretical error proportion p, highlighting the significance of hypothesis testing: the number of tests conducted increases until $p = 0.3$, after which it begins to decrease. Similarly, hypothesis testing tends to reject shapes for "small" values of p, but as p increases, the number of accepted shapes rises, and the test statistic value decreases (refer to Fig. 4.5b).

Considering the physiological error proportion that maximises the likelihood value, *i.e.*, $p = 0.5$, it appears that 245 shapes require a hypothesis test to decide their acceptance: most of them are *rejected* (182) whereas shapes that are *accepted* with a hypothesis test represent 33.7% of the total accepted shapes (187 accepted shapes).

In addition, the extended SHACL validation reports are presented in HTML format with an STTL transformation [CF15]. STTL is an extension to the SPARQL query



(a) $L_{|\upsilon_s^-|}$ average

(b) X_s^2 average

Figure 4.5: Average value of (a) likelihood measures and (b) statistic test as functions of the theoretical error proportion p.

language to transform RDF in any template-specified text result format (*e.g.*, CSV, HTML, ...), which is populated with the results of a SPARQL query. Consequently, we extract values from the extended validation reports to produce an HTML page, presenting the report's results in a table. Fig. 4.7 presents an excerpt of 20 out of 377 results obtained for a theoretical error proportion $p = 0.5$.

As a first step in comparing computation times of the probabilistic validation against the standard validation, we measure the time spent (wall clock) to assess the *Covid-on-the-Web* subgraph against the shapes graph using both probabilistic and standard validation methods. We notice that the probabilistic validation framework took 95 seconds to complete, while the standard validation took 89 seconds: the probabilistic framework takes **6.31%** more time than standard validation. Despite the fact that this increased time appears to be linear, a more in-depth analysis (*e.g.*, CPU time analysis) is needed to come to a conclusion on this question.

4.5 Conclusion

In this chapter, we have introduced a probabilistic framework for SHACL validation. We extend the SHACL validation report by proposing a probabilistic model and an extended vocabulary to express additional information, such as the likelihood measure. Additionally, we propose a decision model for the acceptance of probabilistic assumptions. The experiments demonstrated the approach's capabilities to validate a real-world RDF dataset against a set of SHACL shapes while accepting a reasonable error rate of p. As future work, we plan to extend our proposed framework to *complex* shapes: *e.g.*, recursive shapes which are the focus of ongoing research [CRS18, ACO+20], SHACL

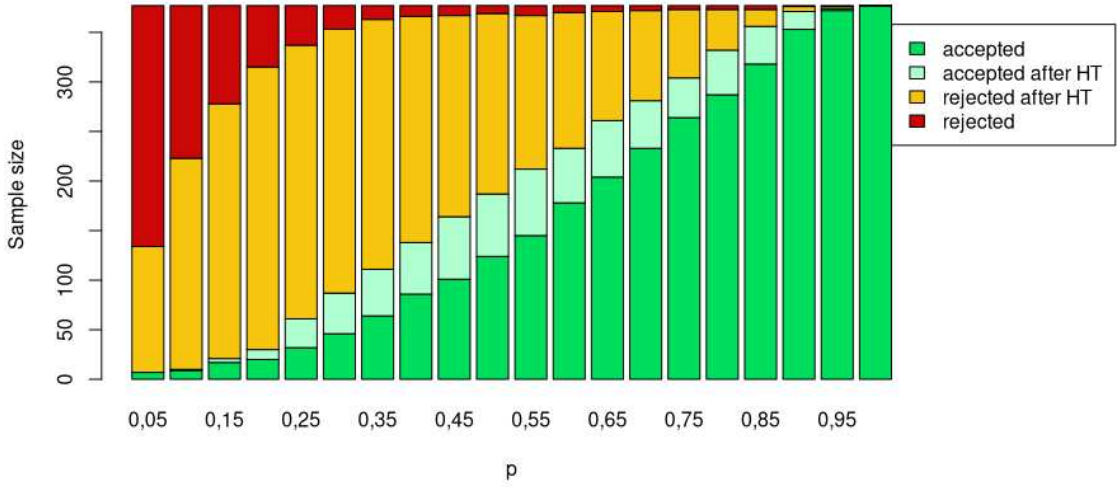


Figure 4.6: Shapes acceptance as a function of the theoretical error proportion p (HT= Hypothesis Testing).

antecedent	consequent	referenceCardinality	#violation	likelihood	generality	X^2_S	Acceptance
two-hybrid screening	protein–protein interaction	48	19	0.041004880900459284	0.00021178308117910231		true
nidovirales	proteolysis	80	69	8.6669313322632E-12	0.00035297180196517053	42.05	false
intensive care medicine	acute respiratory distress syndrome	166	139	9.193409214822706E-20	0.00073241648907772 88	75.56626506024097	false
astrocyte	central nervous system	70	34	0.09238587705330051	0.0003088503267195242		true
dopamine	serotonin	10	6	0.205078125	0.00004412147524564632	0.4	true
crystallography	crystal structure	20	7	0.0739288330078125	0.00008824295049129263		true
human parainfluenza	adenoviridae	237	133	0.00880821375320367	0.0010456789633218177	3.548523206751055	true
carbohydrate	lectin	114	75	2.4200572197826046E-4	0.000502984817800368	11.368421052631579	false
mycoplasma bovis	bovine coronavirus	12	6	0.2255859375	0.00005294577029477558		true
crystallization	diffraction	31	21	0.020653086248785257	0.0001367765732 6150358	3.903225806451613	false
membrane raft	methyl	32	19	0.08087921887636185	0.0001411887207860682	1.125	true
ifitm1	ifitm3	27	9	0.03491956740617752	0.0001191279831 6324504		true
multiple sclerosis	myelin	139	97	1.0209205741082355E-6	0.0006132885059144837	21.762589928057555	false
wheeze	asthma	85	44	0.08188889187584301	0.00037503253958799367	0.10588235294117647	true
influenza a virus subtype h5n1	avian influenza	277	165	2.969648471686876E-4	0.001222164864304403	10.140794223826715	false
hepatocellular carcinoma	liver cirrhosis	72	46	0.005843155895129734	0.00031767462176865343	5.555555555555555	false
diffraction	x-ray crystallography	16	7	0.174560546875	0.0000705943603930341		true
feline infectious peritonitis	feline coronavirus	130	46	2.6051939133257 92E-4	0.000573579178193402		true
aedes aegypti	culicidae	21	4	0.002853870391845703	0.00009265509801585726		true
monomer	oligomer	83	70	5.4692741602999564E-11	0.0003662082445388644	39.144578313253014	false

Figure 4.7: SHACL validation report in HTML format for $p = 0.5$.

shapes that express more than one constraints, ... We also plan to investigate the automatic extraction or generation of SHACL shapes from reference RDF datasets to capture domain knowledge as constraints.

This probabilistic framework has opened up the perspective of an evolutionary discovery of candidate shapes (to capture domain knowledge as constraints) that takes into account physiological errors in RDF data graphs. This is further discussed in the next chapter.

Chapter 5

An Algorithm Based on Grammatical Evolution for Discovering SHACL Constraints

5.1 Introduction

We are interested in discovering SHACL shapes that express domain constraints from RDF data graphs. As SHACL is a relatively new language, real-world data graphs have few associated SHACL shapes, which motivated the following research question (**RQ4**): *How to discover SHACL shapes from RDF data automatically?* We suggest that a generative approach for the automatic construction of candidate shapes using RDF data is one of those that can fulfil this purpose. To this end, we propose an evolutionary discovery of candidate shapes based on Grammatical Evolution using the probabilistic framework presented in Chapter 4 for assessing candidate shapes during the SHACL validation, which is required because of the heterogeneity and incompleteness inherent in open RDF data.

This chapter focuses on an algorithm based on Grammatical Evolution for generating candidate SHACL shapes using BNF grammar and RDF data as input. The approach addresses the limitations regarding the kind of SHACL shapes that can be extracted from an RDF data graph.

While the GE approaches presented in Section 2.3 have been tested on well-known benchmarks, *e.g.*, *Santa Fe Trail*, *Boston Housing*, ... their applicability on RDF data mining has not yet been demonstrated. Only Nguyen and Tettamanzi have proposed

an adaptation of GE for discovering OWL disjointness axioms [NT19a] and complex disjointness axioms [NT20c], with some promising results.

The remainder of this chapter is organized as follows: Section 5.2 presents the BNF grammar design to build well-formed candidate SHACL shapes. Section 5.3 focuses on the fitness function (Section 5.3.2) based on the probabilistic framework by defining an acceptability measure (Section 5.3.1). A recombination operator responding to the redundancy problem and variation operators is presented in Section 5.4. Section 5.5 presents the experiments carried out on the *Covid-on-the-Web* subgraph (see Section 4.3.1). The results of our experiments are presented in Section 5.6, and we conclude the chapter in Section 5.7.

5.2 BNF Grammars of SHACL Shapes

We propose an extensive method for writing BNF grammars in order to produce and exploit well-formed SHACL shapes as individuals in an evolutionary process. We defined a BNF grammar compliant with the SHACL W3C recommendation [KK17] in Fig. 5.1 to produce shapes targeting nodes of a specified class (`sh:targetClass`) and constraining them to be linked through the predicate `rdf:type` to another specified class.[1] This grammar provides both the *phenotypic* and *genotypic* characterization for each individual through a set of *static rules* and *dynamic rules*. The dynamic rules system, proposed by Nguyen and Tettamanzi [NT19a, NT20c], allows the mapping between rules and RDF data using SPARQL queries to build candidate disjointness axioms.

The static rules are the *immutable* components of the phenotypic character. In contrast, the dynamic rules are the *problem instance-dependent* components of the phenotypic character, where each rule has *one or many* possible values (*i.e.*, RDF nodes), and a genotype identifies each value. However, the dynamic rules (proposed by Nguyen and Tettamanzi) were hard-coded in their system, limiting their approach to specific kinds of RDF nodes in the RDF data graph. For this reason, we extended the dynamic rules design by directly enabling the user to write **embedded SPARQL queries** as values of one or more production rules in the grammar to perform the mapping with the desired granularity. The whole process is presented in Fig. 5.2.

To illustrate, in Fig. 5.1, the `<Class>` non-terminal is defined by a dynamic rule to extract all possible classes from the RDF dataset using a SPARQL query: the keyword `SPARQL` is used to specify the query graph pattern to be matched on RDF data. The query results are the set of nodes in the RDF data graph $\mathcal{C}$ bound to variable `?Class`

[1] It should be noted that the two classes may be the same

```
<Shape>      := "a " <NodeShape>
<NodeShape>  := "sh:NodeShape; " <ShapeBody>
<ShapeBody>  := "sh:targetClass " <Class> "; "
                  <ShapeProp>
<ShapeProp>  := "sh:property [ " + <PropBody> " ] ."
<PropBody>   := "sh:path rdf:type ; sh:hasValue " <Class> " ;"
<Class>      := "SPARQL ?x rdf:type ?Class"
```

Figure 5.1: An extract of the BNF grammar for SHACL shapes

in the query graph pattern: $\mathcal{C} = \{c_i, i \in [1, n]\}$. Finally, the initial rule (*i.e.*, `"SPARQL ?x rdf:type ?Class"`) is replaced by the SPARQL results $\mathcal{C}$, *i.e.*, $c_1 \mid c_2 \mid \cdots \mid c_n$.

Using the BNF grammar presented in Fig. 5.1, the genotype of an individual is a pair of codons $[i, j]$, which are decoded into two classes (c_i, c_j) from the dataset using a classic genotype-phenotype mapping, and produce the following phenotype structure:

`"a sh:NodeShape ; sh:targetClass` c_i `; sh:property [ sh:path rdf:type ; sh:hasValue` c_j`" ; ] ."`

It is noteworthy that the proposed grammar can be extended to produce a wider array of SHACL shapes using a variable-length template, e.g. replacing the rule `<ShapeProp>` from Fig. 5.1 by:

```
<ShapeProp> := <Prop> <ShapeProp> | <Prop>
<Prop> := "sh:property [ " + <PropBody> " ] ."
```

Such an extended grammar would produce SHACL shapes specifying one or more constraints (depending on the chosen length). Every kind of target declarations[2] can be exploited as well (see Fig 5.3).

5.3 Probabilistic SHACL Validation as a Fitness Function

5.3.1 Acceptability Function of Candidate Shapes

In order to iteratively produce a final population of SHACL shapes expressing some domain constraints that are implicit in an RDF dataset, we propose a fitness function

[2] https://www.w3.org/TR/shacl/#targets

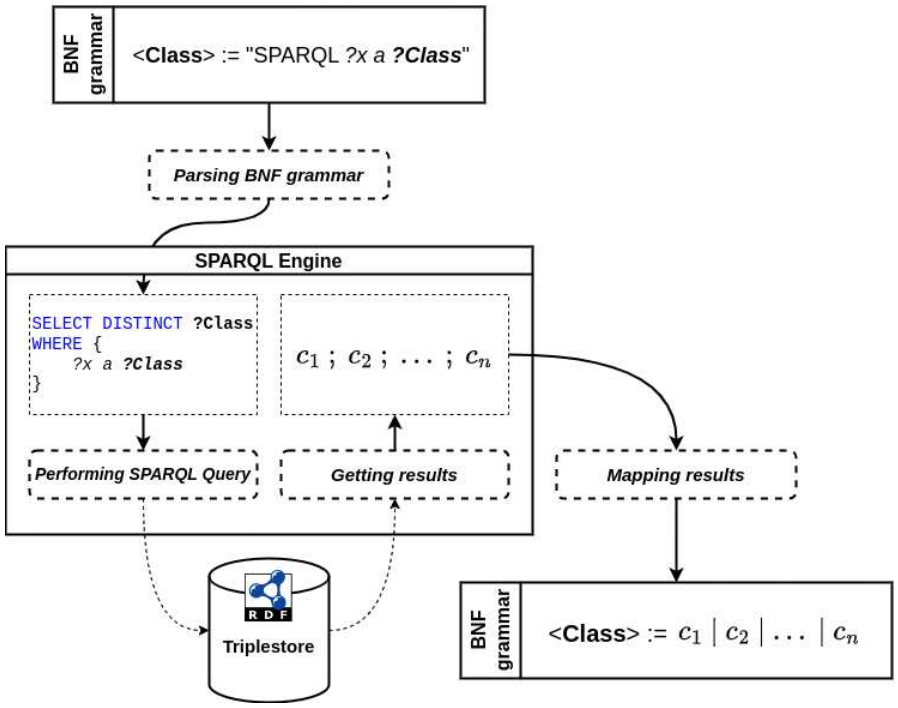


Figure 5.2: Dynamic rules process based on the BNF grammar presented in Fig. 5.1

```
<ShapeBody> := <ClassTarget> | <SubjOfTarget> |
                <ObjOfTarget> | <NodeTarget>
<ClassTarget> := "sh:targetClass " <Class> "; " <ShapeProp>
<SubjOfTarget> := "sh:targetSubjectsOf " <Property> "; " <ShapeProp>
<ObjOfTarget> := "sh:targetObjectsOf " <Property> "; " <ShapeProp>
<NodeTarget> := "sh:targetNode " <Node> "; " <ShapeProp>
```

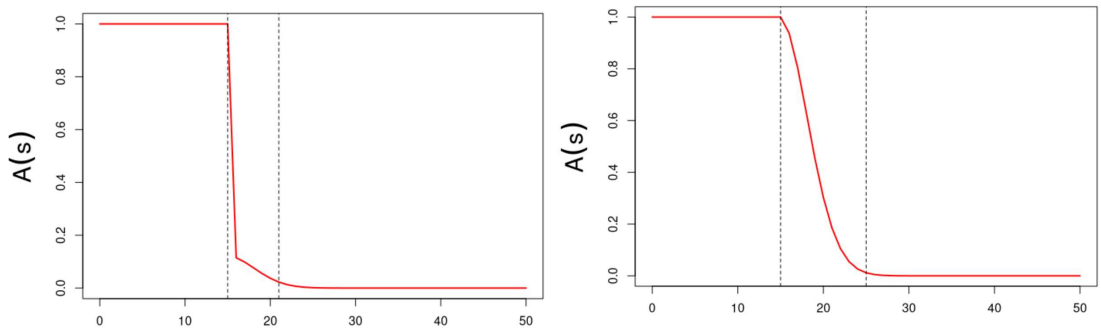
Figure 5.3: Extended BNF grammar: build candidate shapes with different types of targeting

based on an acceptability measure of a shape combined with the probabilistic framework discussed in Chapter 4 to assess the credibility of candidate shapes using RDF facts. The **acceptability** of a SHACL shape s, denoted $A(s)$, depends on the observed error proportion $\hat{p}_s$ (Definition 4.1) when validating an RDF dataset against s: $A(s) \in [0, 1]$ is defined in Eq. (5.1).

Equation 5.1: Acceptance of a candidate shape

$$A(s) = \begin{cases} 1 & \text{if } \hat{p_s} \leq p \text{ or } X_s^2 \in I_a \quad \text{(Definition 4.7)} \\ \frac{L_{|\upsilon_s^-|}}{\mathbb{P}(X=|\upsilon_s| \times p)} & \text{otherwise} \quad \text{(Definition 4.4)} \end{cases}$$

In the case where the null hypothesis is rejected: $A(s) \neq 1$, which means that s is not acceptable, but it may be considered in the grammatical evolution algorithm for crossover or mutation operations. In this context, $A(s)$ represents the likelihood of s ($L_{|\upsilon_s^-|}$) normalised by the maximal value of the probability mass function for a binomial distribution $X \sim B(|\upsilon_s|, p)$. This normalisation ensures a more balanced distribution of $A(s)$ values between 0 and 1, in contrast to the sole likelihood value $L_{|\upsilon_s^-|}$ and therefore avoids excessively penalising individuals who are "*close*" to being acceptable but for whom the likelihood is very low, as depicted in Fig. 5.4.



(a) with $A(s) = L_{|\upsilon_s^-|}$ (b) with $A(s) = \frac{L_{|\upsilon_s^-|}}{\mathbb{P}(X=p \times |\upsilon_s|)}$ (normalisation)

Figure 5.4: Example of the acceptance $A(s)$ values definition

5.3.2 Fitness Function of Candidate Shapes

The fitness function of a SHACL shape s, $F(s)$ combines its acceptability $A(s)$ (see Eq. (5.1)) and the cardinality of its confirmations $|\upsilon_s^+|$ as a "support": it is defined in

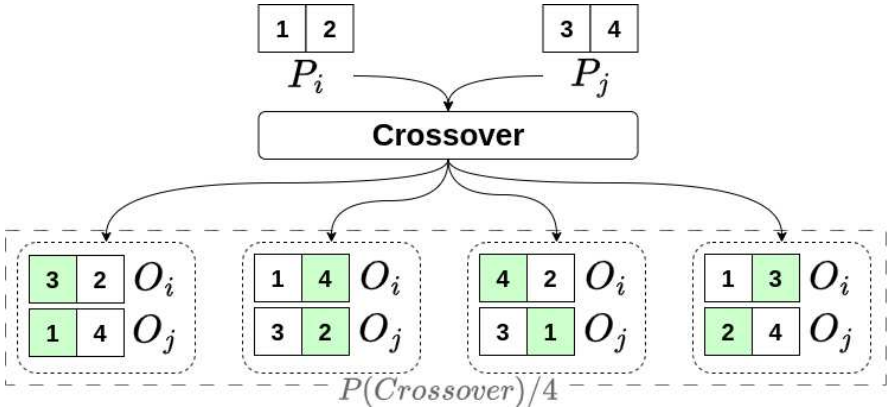
Eq. (5.2). Consequently, the best individuals are those that have been accepted with many RDF facts confirming the candidate shape. Fig. C.2 presents a candidate shape's overall fitness computation process.

Equation 5.2: Fitness function of a candidate shape

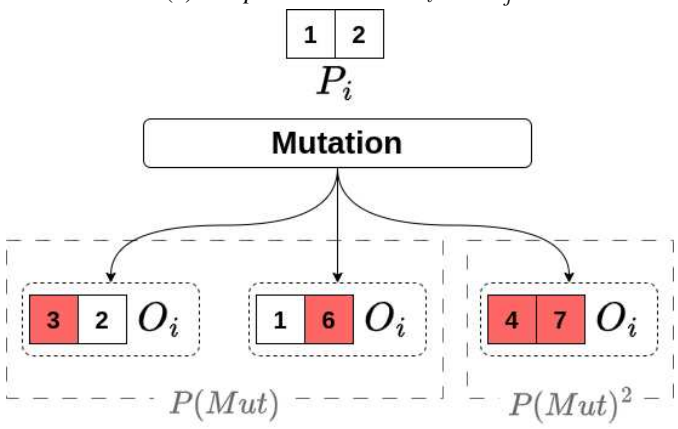
$$F(s) = |\upsilon_s^+| \times A(s)$$

5.4 Variation and Recombination Operators

In this paper, we adapt the main components of the GE variation operators to discover SHACL shapes over RDF facts to consider the issues of the *redundancy* and *low locality*.



(a) *Swap Crossover* of P_i and P_j



(b) *Int Flip Mutation* of parent P_i

Figure 5.5: Representation of GE operators and their probabilities of occurrence

The *redundancy* is observed when many genotypes map the same phenotype expression [LFPC17]. Based on this fact, we adapt the *recombination* phase to filter every offspring by performing a **phenotypic comparison**: Algorithm 4 presents the recombination of selected individuals $\mathcal{S}$ among the whole population $\mathcal{P}$ ($\mathcal{E}$ represents the elite individuals). Line 12 of the Algorithm 4 depicts the criteria: an offspring i is integrated into $\mathcal{R}$ if the phenotypic expression of i is not already observed among the elitist individuals $\mathcal{E}$ and the replacement individuals $\mathcal{R}$. As a consequence, we overcome the reflection of the *redundancy* issue in the final population: $\forall i \in \mathcal{P}, \nexists j \in \mathcal{P} : i \equiv j$.

Algorithm 4 Recombination of a population $\mathcal{P}$

Input: *elite individuals* $\mathcal{E}$ **and** *selected individuals* $\mathcal{S}$
Output: *replacement population* $\mathcal{R}$

1: $\mathcal{R} \leftarrow \{\}$
2: **while** $|\mathcal{R}| \neq |\mathcal{P}| - |\mathcal{E}|$ **do**
3: $\quad \mathcal{C} \leftarrow \{\}$
4: $\quad p_1 \leftarrow \mathcal{S}[random() * |\mathcal{S}|]$
5: $\quad p_2 \leftarrow \mathcal{S}[random() * |\mathcal{S}|]$ *# $random() * |\mathcal{S}|$ as integer*
6: $\quad$ **if** $p_1 \neq p_2$ **then**
7: $\quad\quad \mathcal{C} \leftarrow p_1 \cup p_2$
8: $\quad\quad \mathcal{C} \leftarrow crossover(\mathcal{C})$ *# Figure 5.5a*
9: $\quad\quad \mathcal{C} \leftarrow mutation(\mathcal{C})$ *# Figure 5.5b*
10: $\quad\quad$ **for** $i \in \mathcal{C}$ **do**
11: $\quad\quad\quad$ ***# Phenotypic** comparison*
12: $\quad\quad\quad$ **if** $i \notin \mathcal{E} \cup \mathcal{R}$ **and** $|\mathcal{R}| \neq |\mathcal{P}| - |\mathcal{E}|$ **then**
13: $\quad\quad\quad\quad \mathcal{R} \leftarrow \mathcal{R} \cup i$
14: $\quad\quad\quad$ **end if**
15: $\quad\quad$ **end for**
16: $\quad$ **end if**
17: **end while**
18: **return** $\mathcal{R}$

The *locality* issue relies on the relationship between the selected rules from parent to offspring. As illustrated in Fig. 5.5, certain results from the variation operators may result in a low locality, meaning a significantly different offspring, while others may lead to a relatively strong locality. Considering the grammar in Fig. 5.1, a change in the first codon, impacting the target class value (`sh:targetClass`) c_i, can have a significant impact on the meaning of the phenotypic trait. This is because SHACL validation is carried out on the nodes instantiating c_i: replacing c_i with a new production rule (*e.g.*, c_i') leads to a locality as low as the proximity between c_i and c_i' (*e.g.*, common instances). On the other hand, modifying the last codon has a lower impact on SHACL validation,

as the targeted nodes remain the same, leading to a reasonably strong locality, even if the meaning of the phenotype differs.

To measure the evolution of the population through each recombination phase, we propose the population development rate metric $\mathcal{P}_i^{dev}$: Let $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ the population at generations $i-1$ and i (*i.e.*, measurable after the first recombination of the evolutionary process), the population development rate $\mathcal{P}^{dev}$ is the rate of different individuals (*phenotypic comparison*) between $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ (see Eq. (5.3)).

Equation 5.3: Population development rate

$$\mathcal{P}_i^{dev} = \frac{|\mathcal{P}_i \setminus \mathcal{P}_{i-1}|}{|\mathcal{P}_i|}, \ \mathcal{P}_i^{dev} \in [0, 1]$$

5.5 Experiments

To validate the proposed approach, we consider the discovery of credible candidate shapes representing **association rules** (see Section 4.3.2) between *Wikidata* named entities, *i.e.*, rules of the form $\mathcal{X} \rightarrow \mathcal{Y}$, from the *Covid-on-the-Web* RDF data graph presented in Section 4.3.1. We use the BNF grammar presented in Fig. 5.1 to generate candidate shapes: each candidate involves a first *Wikidata* entity, *i.e.*, the *antecedent* called $\mathcal{X}$, and target nodes n (referring to scientific articles) instances of $\mathcal{X}$ using the `sh:targetClass` property. The proposed constraint verifies if these nodes are also typed by a second Wikidata entity, *i.e.*, the *consequent*, called $\mathcal{Y}$, using the `sh:hasValue` constraint applied on the `rdf:type` property. A concrete example of candidate shape is presented in Fig. 4.4.

In these experiments, our main focus was on achieving various acceptable shapes through the discovery process. Although it is clear that using resource-intensive parameters (like large population size and high effort) would provide the best results, we opted for a more balanced set of parameters to minimize computation time.

We used an implementation of the presented algorithm combined with the probabilistic SHACL validation engine implemented in the *Corese* semantic Web factory [Cé23]. We considered a theoretical error proportion $p = 0.5$ (*i.e.*, *physiological* error) according to the experimental results discussed in Chapter 4: this value p maximises the average value of the likelihood measure $\overline{L}$.

The experiments have been performed on a server equipped with an Intel(R) Xeon(R) CPU E5-2637 v2 processor at 3.50GHz clock speed, with 172 GB of RAM, 1 TB of disk space running under the Ubuntu 20.06.4 LTS 64-bit operating system.

5.5.1 A Recall Measure for Acceptable Shapes Coverage

The recall R of our algorithm is defined to assess the ability of our approach to find acceptable candidates in the solution space of the defined problem. The recall provides the rate of *distinct and acceptable solutions found* by our algorithm over the total number of acceptable solutions, denoted $\mathcal{A}$. Let Ω be the set of all possible pairs $(\mathcal{X}, \mathcal{Y})$ of distinct named entities extracted from the *Covid-on-the-web* dataset. Considering the dataset characteristics presented in Table 4.1, the total number of possible solutions is defined as:

$$|\Omega| = 6,331 \times 6,330 = 40,075,230$$

We sample a random subset of Ω, denoted Ω' (*i.e.*, $\Omega' \subseteq \Omega$), to estimate the number of acceptable shapes in Ω. The *Cochran* formula is used to compute the minimal cardinality $|\Omega'|$ while ensuring the representativity of the subset:

$$|\Omega'| = \frac{z^2 \times p \times (1-p)}{m^2} = \frac{2.58^2 * 0.5^2}{0.02^2} \approx 4,161$$

where $z \approx 2.58$ is the standard normal *z-table* with a confidence level of 99%, $m = 0.02$ (2%) is the tolerated margin of error and $p = 0.5$[3] the probability that the candidate shape is acceptable. $4,161$ distinct candidate shapes have been randomly generated and evaluated against the *Covid-on-the-Web* subgraph using the probabilistic SHACL validation with a physiological error rate $p = 0.5$: the results show that only 2 shapes in Ω' have been accepted, *i.e.*, 0.05% of the total. The total number of acceptable shapes $|\mathcal{A}|$ is thus estimated:

$$|\mathcal{A}| \cong |\Omega| \times 0.0005 = 20,037.6$$

The recall of our algorithm, denoted $R(x)$, measures how effectively the acceptable shapes x cover the set of acceptable solution space $\mathcal{A}$, as defined in Eq. (5.4).

Equation 5.4: Recall measure

$$R(x) = \frac{x}{|\mathcal{A}|} \times 100, \;\; R(x) \in [0, 1]$$

[3] which is unknown in this context, so $p = 0.5$

5.6 Results

5.6.1 $|\mathcal{P}|/E$ choice

We assessed our approach with manually defined small population sizes (*i.e.*, $|\mathcal{P}|$) and quite low *effort* values E, and we analysed the effects of the ratio $|\mathcal{P}|/E$. This corresponds to verifying if our algorithm can find credible and surprising candidate shapes using a minimum investment of CPU time. Consequently, we performed 10 executions of our algorithm using the different parameter settings presented in Table 5.1 (*i.e.*, 90 in total) and analyzed the final whole population $\mathcal{P}$ and the final elitist subset $\mathcal{E}$ ($\mathcal{E} \subseteq \mathcal{P}$). Each configuration has been assessed regarding the following metrics:

- the average fitness value: $\overline{F}$
- the average rate of accepted shapes: $\overline{\%A}$
- the average CPU time (in ms) for evaluating an individual: $\overline{T}$
- the average recall: $\overline{R}$

Table 5.1: Used parameters to analyse the impact of $|\mathcal{P}|$/E choice.

Parameters	**Value(s)**
GE	
$\|P\|$	{100; 200; 500}
E	{5,000; 10,000; 20,000}
% Selection ($\mathcal{E}$)	20%
% Selection ($\mathcal{R}$)	40%
Selection type	Tournament
% Tournament	25%
Crossover type - P	Swap (Fig. 5.5a) - 75%
Mutation type - P	Int Flip (Fig. 5.5b) - 5%
Probabilistic SHACL validation	
Confidence level $1-\alpha$	95%
Physiological error rate p	50%

The results presented in Table 5.2 show that a gradual increase of the effort E tends to enhance the global quality of candidate shapes into $\mathcal{P}$: This is evident regarding the metrics related to individual quality ($\overline{F}$, $\overline{\%A}$, $\overline{L}$ and $\overline{R}$) regardless of the population size $|P|$. This trend is clearer regarding the elitist part $\mathcal{E}$. The evolution of the population

development rate at each generation $\mathcal{P}_i^{dev}$, presented in Fig. C.3, suggests that its value tends to stabilise as the population size increases (regardless of effort E).

Table 5.2: Results obtained using the parameters presented in Table 5.1. The best result for each metric is in **bold**, and the second best is underlined. Highlighted columns are the best.

		$\lvert P\rvert = 100$			$\lvert P\rvert = 200$			$\lvert P\rvert = 500$		
		E = 5,000	E = 10,000	E = 20,000	E = 5,000	E = 10,000	E = 20,000	E = 5,000	E = 10,000	E = 20,000
From $\mathcal{P}$	$\overline{F}$	0.79 ± 1.4	0.64 ± 1.05	$\mathbf{1.52 \pm 1.7}$	0.53 ± 0.81	1.24 ± 1.83	1.51 ± 1.14	0.9 ± 0.97	1.07 ± 1	1.3 ± 0.87
	$\overline{\%A}$	1.9 ± 2.77	4 ± 2.71	$\mathbf{10.1 \pm 4.18}$	2 ± 0.75	3.3 ± 2.15	8.05 ± 3.11	1.56 ± 0.76	2.36 ± 1.05	4.98 ± 2.33
	$\overline{L}$	2.55 ± 1.77	4.86 ± 1.66	$\mathbf{6.74 \pm 2.07}$	2.42 ± 0.97	4.41 ± 1.85	6.18 ± 1.13	1.25 ± 0.29	2.07 ± 0.59	4.47 ± 1.29
	$\overline{T}$	18 ± 2.73	17.98 ± 2.96	21.94 ± 3.73	$\mathbf{16.84 \pm 2.4}$	19.22 ± 3.03	21.14 ± 3.65	19.21 ± 2.28	18.39 ± 3.71	19.31 ± 3.71
	$\overline{R}$	0.01 ± 0.01	0.02 ± 0.01	0.05 ± 0.02	0.02 ± 0.01	0.03 ± 0.02	0.08 ± 0.03	0.04 ± 0.02	0.06 ± 0.03	$\mathbf{0.13 \pm 0.06}$
From $\mathcal{E}$	$\overline{F}$	3.91 ± 7	2.77 ± 5.28	$\mathbf{7.41 \pm 8.48}$	2.65 ± 4.06	6.11 ± 9.15	$\mathbf{7.41 \pm 5.74}$	4.41 ± 4.9	5.29 ± 4.99	6.36 ± 4.39
	$\overline{\%A}$	9.5 ± 13.83	18.5 ± 12.92	$\mathbf{49.5 \pm 19.64}$	9.5 ± 4.22	15.75 ± 11.31	39.5 ± 15.27	6.7 ± 3.13	11.2 ± 4.94	24.1 ± 10.99
	$\overline{L}$	9.36 ± 6.74	16.18 ± 4.47	$\mathbf{21.51 \pm 4.88}$	8.21 ± 3.23	14.96 ± 5.84	20.73 ± 2.25	4.03 ± 1.1	7.22 ± 1.86	15.19 ± 3.67
	$\overline{T}$	10.6 ± 2.45	9.35 ± 1.43	7.68 ± 1.26	9.22 ± 1.28	7.53 ± 1.11	$\mathbf{6.65 \pm 0.65}$	10.64 ± 2.83	7.81 ± 0.94	7.21 ± 2.66
	$\overline{R}$	0.01 ± 0.01	0.02 ± 0.01	0.05 ± 0.02	0.02 ± 0.01	0.03 ± 0.02	0.08 ± 0.03	0.03 ± 0.02	0.06 ± 0.02	$\mathbf{0.12 \pm 0.06}$

Table 5.3: *Mann-Whitney-Wilcoxon* test: comparison between the results obtained for ($|P| = 100$; $E = 20,000$) and ($|P| = 200$; $E = 20,000$) with $\alpha = 5\%$.

From $\mathcal{P}$		From $\mathcal{E}$	
Metrics	P-value	Metrics	P-value
$\overline{F}$	0.528	$\overline{F}$	0.529
$\overline{\%A}$	0.198	$\overline{\%A}$	0.210
$\overline{L}$	0.684	$\overline{L}$	0.796
$\overline{T}$	0.631	$\overline{T}$	0.076
$\overline{R}$	**0.037**	$\overline{R}$	**0.028**

Globally, it appears that smaller values of $|\mathcal{P}|$ with higher effort lead to the best results. When comparing the results obtained with ($|\mathcal{P}| = 100$; $E = 20,000$) and ($|\mathcal{P}| = 200$; $E = 20,000$), there are many similarities, except for the proportion of acceptable shapes in $\mathcal{E}$ (49.5% and 39.5% respectively). This is why a *Mann-Withney-Wilconox* test was conducted for each metric to identify any differences in the results (see Table 5.3): the test demonstrated that only the average recall $\overline{R}$ values from $\mathcal{P}$ and $\mathcal{E}$ were significantly different (< 0.05), indicating that the choice of ($|\mathcal{P}| = 200$; $E = 20,000$) is the best one for this measure.

5.6.2 Selection ($\mathcal{R}$) pressure

We believe that analyzing selective pressure and understanding its impact on metrics is best achieved by examining the population with the smallest size and the highest effort, i.e. ($|\mathcal{P}| = 100$; $E = 20,000$). As a result, we have explored various selection types, including *Scaled Roulette Wheel* and *Tournament*, with the settings outlined in Table 5.4.

The results obtained through the use of the *Scaled Roulette Wheel* selection are shown in Table 5.5 and demonstrate that the metrics improve with a high selection rate ($S = 60\%$), despite the scarcity of highly promising candidates. A high selection rate expands the exploration of the solution space, leading to a notable disparity between the results from $\mathcal{P}$ and the elitist subset $\mathcal{E}$.

On the other hand, the results obtained with the *Tournament* selection, presented in Table 5.6, highlight the same trend. With a selection rate of $S = 20\%$, there is a minimal overall difference in results between populations $\mathcal{P}$ and $\mathcal{E}$, enhancing the *homogeneity* of the population $\mathcal{P}$. Conversely, a higher selection rate ($S = 60\%$) signifies a more diverse

Table 5.4: Used parameters to analyse the impact of the selective pressure on $\mathcal{R}$.

Parameters	**Value(s)**
GE	
$\|P\|$	100
E	$20,000$
% Selection ($\mathcal{E}$)	20%
Selection type	{Scaled Roulette Wheel; Tournament}
% Selection ($\mathcal{R}$)	{20%; 40%; 60%}
% Tournament (*Tour*)	{10%; 25%; 50%}
Crossover type - P	Swap (Fig. 5.5a) - 75%
Mutation type - P	Int Flip (Fig. 5.5b) - 5%
Probabilistic SHACL validation	
Confidence level $1-\alpha$	95%
Physiological error rate p	50%

population (*i.e.*, more *heterogeneous*), as the overall difference in results between $\mathcal{P}$ and $\mathcal{E}$ is substantial. Comparing the average CPU time for the whole population $\mathcal{P}$ with the average CPU time for the elite $\mathcal{E}$ demonstrates that a higher selection rate S leads to significant population diversity while maintaining an outstanding elite population. This ensures high-quality elite shapes and a broad exploration of the overall population, enabling the discovery of diverse and potentially interesting shapes.

Table 5.5: Results obtained using the *Scaled Roulette Wheel* selection and parameters presented in Table 5.4: best result for each metric is in **bold** and second best underlined.

		$S = 20\%$	$S = 40\%$	$S = 60\%$
From $\mathcal{P}$	$\overline{F}$	0.86 ± 1.06	$\mathbf{1.78 \pm 2.99}$	0.56 ± 0.16
	$\overline{\%A}$	8.7 ± 3.83	7.3 ± 5.33	$\mathbf{11.7 \pm 3.06}$
	$\overline{L}$	7.85 ± 2.46	$\mathbf{8.85 \pm 1.89}$	8.25 ± 2.6
	$\overline{T}$	20.25 ± 6.26	19.86 ± 8.61	$\mathbf{19.83 \pm 5.78}$
	$\overline{R}$	0.04 ± 0.02	0.04 ± 0.03	$\mathbf{0.06 \pm 0.02}$
From $\mathcal{E}$	$\overline{F}$	4.27 ± 5.29	$\mathbf{8.87 \pm 14.97}$	2.74 ± 0.8
	$\overline{\%A}$	43.5 ± 19.16	36 ± 25.47	$\mathbf{58 \pm 14.94}$
	$\overline{L}$	24.33 ± 4.36	$\mathbf{25.25 \pm 5.5}$	23.94 ± 4.31
	$\overline{T}$	8.49 ± 1.29	7.77 ± 1.46	$\mathbf{7.38 \pm 1.19}$
	$\overline{R}$	0.04 ± 0.02	0.04 ± 0.03	$\mathbf{0.06 \pm 0.02}$

Table 5.6: Results obtained using the *Tournament* selection and parameters presented in Table 5.4: best result for each metric is in **bold**, second best underlined. The Highlighted column corresponds to the *reference* results presented in Table 5.2.

		$S = 20\%$			$S = 40\%$			$S = 60\%$		
		$Tour = 10\%$	$Tour = 25\%$	$Tour = 50\%$	$Tour = 10\%$	$Tour = 25\%$	$Tour = 50\%$	$Tour = 10\%$	$Tour = 25\%$	$Tour = 50\%$
From $\mathcal{P}$	$\overline{F}$	1.78 ± 1.93	1.82 ± 2.44	1.08 ± 1.98	1.2 ± 1.54	1.52 ± 1.7	1.71 ± 2.42	0.99 ± 0.83	$\mathbf{2.32 \pm 2}$	1.78 ± 2.33
	$\overline{\%A}$	$\mathbf{11.1 \pm 5.2}$	8.9 ± 5.45	8.1 ± 5.43	9.9 ± 4.51	10.1 ± 4.18	9.7 ± 4.27	8.3 ± 3.97	10.7 ± 4.99	8.4 ± 3.27
	$\overline{L}$	5.55 ± 2.6	6.84 ± 2.81	6.98 ± 2.58	6.11 ± 1.14	6.74 ± 2.07	6.03 ± 2.02	6.12 ± 1.32	6.36 ± 2.1	$\mathbf{7.1 \pm 1.32}$
	$\overline{T}$	22.92 ± 9.12	$\mathbf{16.4 \pm 3.4}$	18.88 ± 6.41	23.93 ± 5.93	21.94 ± 3.73	21.66 ± 7.37	28.44 ± 7.5	27.89 ± 6.46	27.9 ± 6.62
	$\overline{R}$	$\mathbf{0.06 \pm 0.03}$	0.04 ± 0.03	0.04 ± 0.03	0.05 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.04 ± 0.02	0.05 ± 0.03	0.04 ± 0.02
From $\mathcal{E}$	$\overline{F}$	8.83 ± 9.64	9.02 ± 12.21	5.34 ± 9.94	5.93 ± 7.69	7.41 ± 8.48	8.47 ± 12.13	4.9 ± 4.14	$\mathbf{11.55 \pm 10.02}$	8.86 ± 11.68
	$\overline{\%A}$	$\mathbf{55.5 \pm 25.98}$	43.5 ± 26.46	39.5 ± 25.65	49 ± 21.96	49.5 ± 19.64	48.5 ± 21.35	41 ± 19.12	52.5 ± 24.41	42 ± 16.36
	$\overline{L}$	18.96 ± 6.49	19.1 ± 3.73	21.6 ± 4.95	21.22 ± 2.5	21.51 ± 4.88	19.77 ± 3.69	22.68 ± 4.24	24.25 ± 5.67	$\mathbf{24.46 \pm 3.09}$
	$\overline{T}$	7.54 ± 1	8.09 ± 1.07	7.76 ± 0.79	7.4 ± 1.52	7.68 ± 1.26	7.25 ± 1.51	$\mathbf{6.93 \pm 0.43}$	7.06 ± 1.18	8.02 ± 1.38
	$\overline{R}$	$\mathbf{0.06 \pm 0.03}$	0.04 ± 0.03	0.04 ± 0.03	0.05 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.04 ± 0.02	0.05 ± 0.02	0.04 ± 0.02

Table 5.7: Overview of the distinct and acceptable shapes discovered from all the performed experiments.

Metrics	$\overline{F}$	$\overline{L}$	$\overline{T}$	$\overline{R}$
Values	19.49	19.14	7.93	8.91

5.6.3 Acceptable shapes

We discovered a set of 1,766 distinct and acceptable shapes among all the experiments we conducted: an overview of these results is presented in Table 5.7. Some of these shapes have been accepted despite a high violation rate ($> 50\%$), but they can still be easily validated: *e.g.*, the candidate shape implying the following rule (`gene expression profiling` $\rightarrow$ `gene expression`)[4] is easily understandable and acceptable, despite having a violation rate of 52.6%. Furthermore, 46.38% of the whole acceptable shapes has been accepted after performing hypothesis testing, translating a significant impact on the acceptance of shapes and the mining process. However, some acceptable shapes require validation from experts due to their complexity: *e.g.*, the following rule (`chemokine` $\rightarrow$ `cytokine`) has been *manually* validated after some research: "*Chemokines [...] are a family of small cytokines*"[5]. However, some of them require an in-depth domain knowledge to be validated: *e.g.*, the rule (`tlr9` $\rightarrow$ `toll-like receptor`).

Some discovered "*very well fit*" candidate shapes impacts the standard deviation of many values, *e.g.*, $\overline{F}$ and $\overline{\%A}$ (some of these are higher than the mean value): this is correlated to some trivial shapes discovered with *identical* classes for the `sh:targetClass` and the constraint `sh:hasValue`, which implies a perfect acceptance of these trivial candidates (*i.e.*, no possible violations), impacting their fitness values. These shapes can be generated because of the selection of production rules using a *modulo* operator and a *quasi-infinite* range for codon definition. However, this is a fairly rare occurrence: we observe it among (only) 132 candidates from the 1,766 acceptable shapes, which is approximately 7.47%. This is why we suggest accepting a low occurrence of these shapes being discovered (even if they are meaningless) to avoid any negative impact on exploring the solution space.

The $\overline{T}$ value presented in Table 5.7 and the correlation between the number of violations and the CPU time presented in Fig. 5.6 suggest that the CPU time required to invest in an evolutionary process is maximal at the beginning, then decreases as the

[4] Obtained results are SHACL shapes similar to the one presented in Fig. 4.4, this notation is used to simplify the reading.

[5] https://en.wikipedia.org/wiki/Chemokine

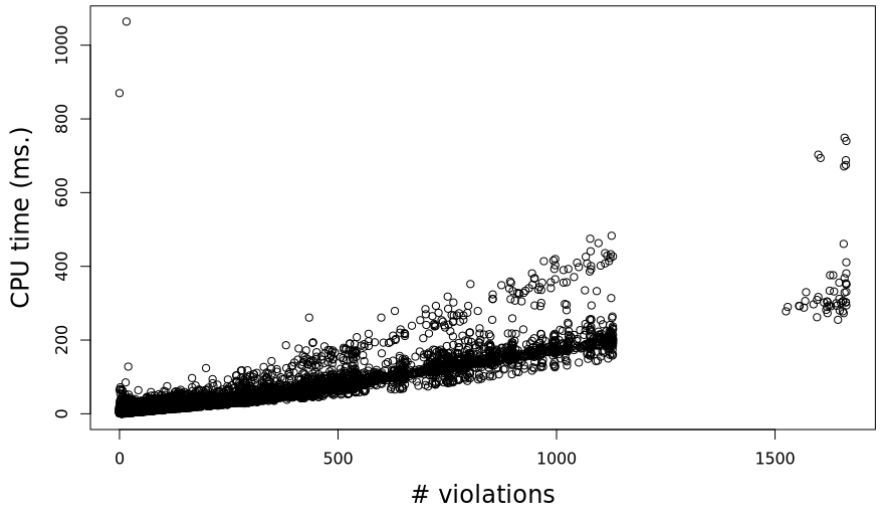


Figure 5.6: CPU time spent for the probabilistic SHACL validation of each discovered shape considering the number of violations

average number of violations decreases (and therefore when the shapes become more and more acceptable). Considering this expected evolution and the low average time, this evolution demonstrates the relevance of this evolutionary approach to the discovery of SHACL shapes over an RDF data graph and appears suitable for scalability.

5.7 Conclusion

In this chapter, we have presented a framework using an evolutionary algorithm based on Grammatical Evolution to discover candidate SHACL shapes from a real-world RDF data graph based on a manually defined BNF grammar. The proposed algorithm effectively responds to the *redundancy* issue by proposing an adaptation of the recombination phase, but the *low locality* issue appears to be *problem-dependent*. The proposed approach aims to tackle the requirement of a broad exploration of the wide search space of possible SHACL shapes (discussed in Section 5.5.1) to discover acceptable ones. The framework uses a probabilistic SHACL validation process with an *acceptability* measure and a fitness function to assess candidate shapes and retain the best ones through the evolutionary process while considering the physiological error proportion. The conducted experiments have led to the discovery of a large set of acceptable candidate shapes while considering a high physiological error rate. The first experiments show that a low population size

and a high effort provide the best statistics on the discovered candidate shapes (average fitness, average likelihood, ...). Based on these results, we analysed the selection pressure by varying the type and the proportion of selected individuals for the recombination phase. We have observed that these variations significantly impact the *homogeneity* of the population and its elite. Finally, our approach is able to capture credible SHACL shapes, describing domain constraints from the *Covid-on-the-Web* RDF dataset.

In the next chapter, we present the web application we developed to set up and manage the process of SHACL shape mining.

Chapter 6

RDFminer: a Tool to Automatically Discover Knowledge From RDF Data Graph

6.1 Introduction

RDFminer is an open-source Web application that enables the automatic discovery of SHACL shapes and OWL axioms through an evolutionary process. It takes an RDF data graph and a BNF grammar as input, from which candidate individuals (*i.e.*, axioms or shapes) are randomly generated and assessed using the possibilistic framework presented in Chapter 3 for candidate axiom assessment or the probabilistic framework presented in Chapter 5 for candidate shape assessment. `RDFminer` provides an interactive interface enabling users to launch, monitor and analyse their shape discovery projects in real-time.

This chapter is organized as follows: Section 6.2 presents the evolutionary discovery of candidate shapes and axioms implemented in the `RDFminer-core` component. In Section 6.3, we present the `RDFminer` Web application for monitoring the evolutionary process. We conclude this chapter in Section 6.4.

6.2 Evolutionary Discovery of SHACL Shapes or OWL axioms

`RDFminer-core` is an API that exploits the evolutionary approach based on Grammatical Evolution presented in Chapter 5: its implementation relies on the `GEVA 2.0` [OHG+11] Java library to exploit GE basic features and operators and on the

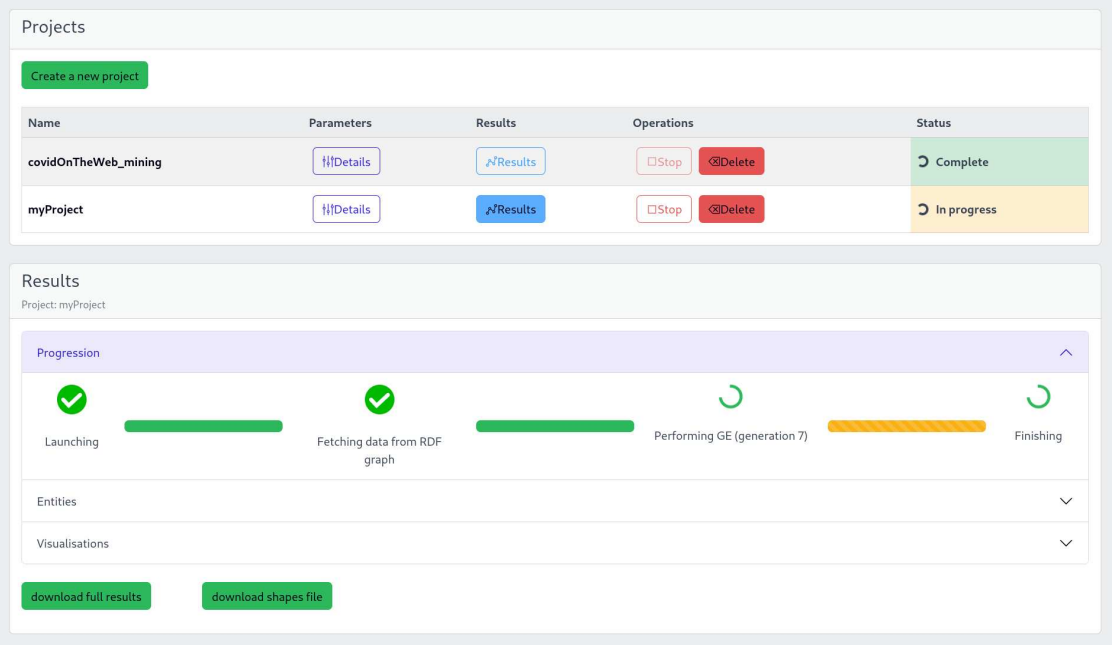


Figure 6.1: `RDFminer` dashboard overview

`Corese` [Cé23] semantic Web factory to query and exploit RDF data throughout the evolutionary process. Additionally, we use the multi-threading system presented in Section 3.3.1 to assess individuals simultaneously.

`GEVA` is an open-source implementation of GE developed by UCD's Natural Computing Research & Applications group.[1] Along with the typical genotype-phenotype mapping feature, `GEVA` also includes a search engine and a basic GUI.[2]

The experiments presented in Section 3.4.2 and Section 5.5 have been performed with `RDFminer`. Although the discovery of axioms is limited to subsumption `SubClassOf` and disjointness `DisjointClasses` axioms (that can involve complex class expressions), `RDFminer` can discover a wide range of constraints (*i.e.*, shapes): it is only limited by the type of constraints that the probabilistic framework can support.

[1] Documentation: `http://ncra.ucd.ie/GEVA/geva.pdf`
[2] GEVA GUI is not used.

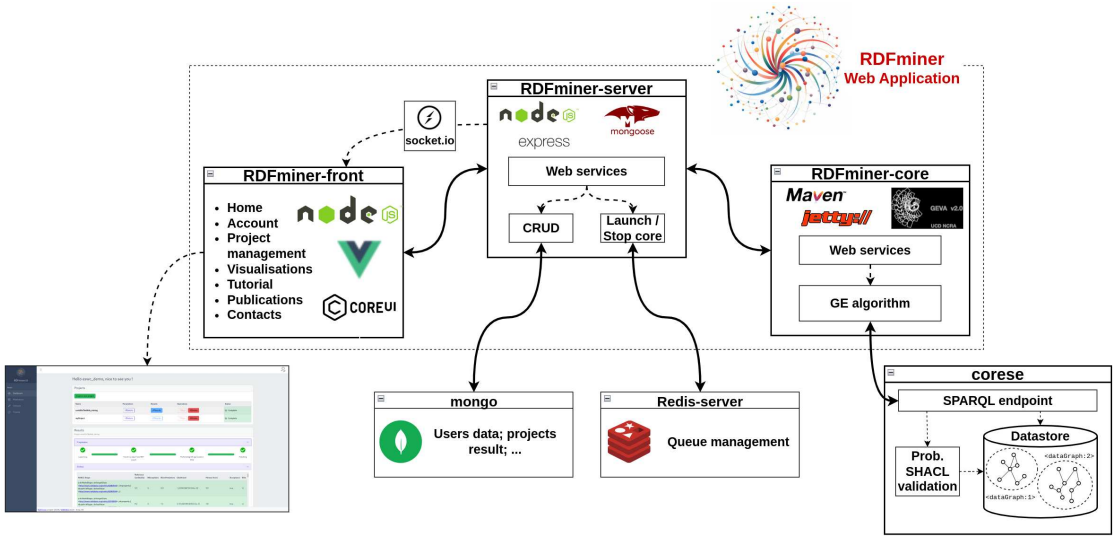


Figure 6.2: Global architecture of `RDFminer`

6.3 A Web Application to Discover SHACL Shapes and OWL Axioms

Exploiting the `RDFminer-core` engine to discover SHACL shapes or OWL axioms is essentially a "*trial-and-error*" process. That is why we developed a Web application to provide the user with an interface that allows them to control the mining process interactively: it enables them to parameterize and launch the discovery process, monitor its execution, and inspect and analyze its results. The global view is depicted in Fig. 6.1.

The overall architecture of the application is depicted in Fig. 6.2. The components are *docker services* that operate together within a *docker-compose* ecosystem.

* **`RDFminer-core`**[3] is an API that exploits the evolutionary algorithm implemented in *Java*. The server relies on implementing a RESTful web service using the *JAX-RS* framework (*i.e.*, *Jetty*[4] server).

* **`RDFminer-front`** is built with *VueJS*[5], a *Javascript* framework, allowing for interactive control of the mining process. Users can customize and initiate the discovery process, supervise its progress, and examine results.

[3] `RDFminer-core` can be used independently of the other components through its API: https://github.com/Wimmics/RDFminer/tree/main/RDFminer-core

[4] https://jetty.org/

[5] https://vuejs.org/

* **`RDFminer-server`** provides web services for interaction between `front` and `core` (with `socket.io` server for WebSockets transport that allows interactions in real-time), and interactions with databases:
  - `mongo`: a MongoDB[6] instance to store users data, projects settings and results
  - `Redis-server`: a Redis[7] DB instance to manage the project execution queue on the production server, *i.e.*, **one run** at a time (for performance reasons)

The architecture has been deployed on a server equipped with an Intel(R) Xeon(R) CPU E5-2637 v2 processor at 3.50GHz clock speed, with 172 GB of RAM, 1 TB of disk space running under the Ubuntu 20.06.4 LTS 64-bit operating system.

6.3.1 Monitoring Dashboard

The connected user can discover axioms or shapes from a given RDF data graph by creating a project and defining the parameters of the mining process: the data graph, the BNF grammar to be considered, and the hyper-parameters of the Grammatical Evolution algorithm. The form[8] is presented in Fig. 6.3, and the configuration is inspired by conducted experiments discussed in Section 5.5. As depicted in Fig. 6.1, the status of the running project is updated in real-time and can be interrupted if needed, and the user can access the *Results* view as well. At the end of the execution, the user can download the SHACL shapes or OWL axioms in Turtle format and/or the complete results file (including individuals, their statistics and the algorithm's statistics) in JSON format for post-processing.

6.3.2 Result Analysis Dashboard

Due to the nature of the evolutionary mining process, the population of candidate shapes or axioms evolves continuously. This dashboard enables users to consult and analyse results in real-time, whether the evolution is ongoing or completed. In more detail, each active project produces real-time results that can be examined using this dashboard. Additionally, users can access completed projects through the dashboard. Throughout each generation, the individuals and their statistics are displayed in a table (see Fig. 6.5), which is composed of the following columns:

[6] https://www.mongodb.com/
[7] https://redis.io/
[8] the form parameters are rigorously detailed in the readme:
https://github.com/Wimmics/RDFminer/tree/main/RDFminer-core

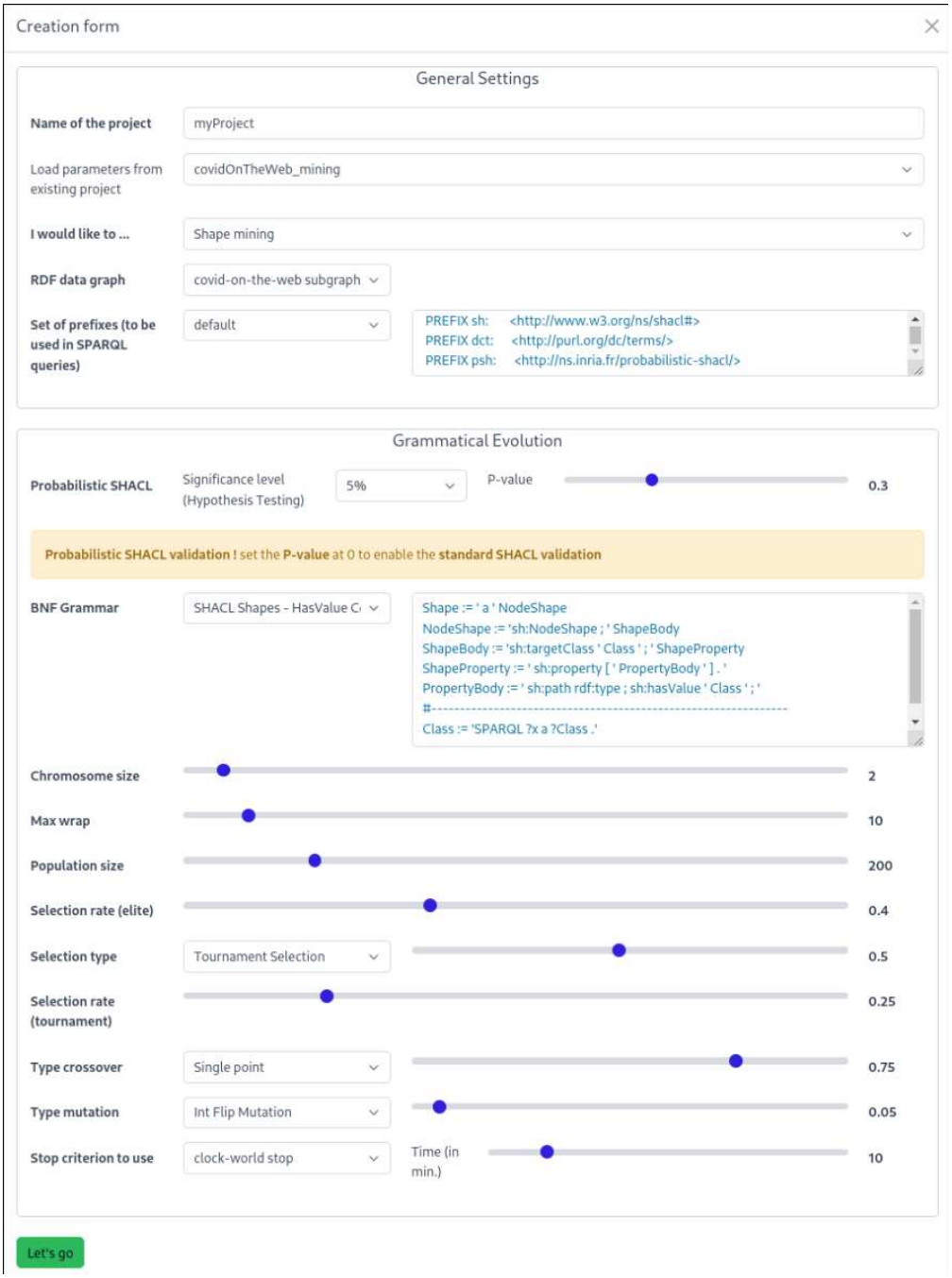


Figure 6.3: `RDFminer` project creation form popup

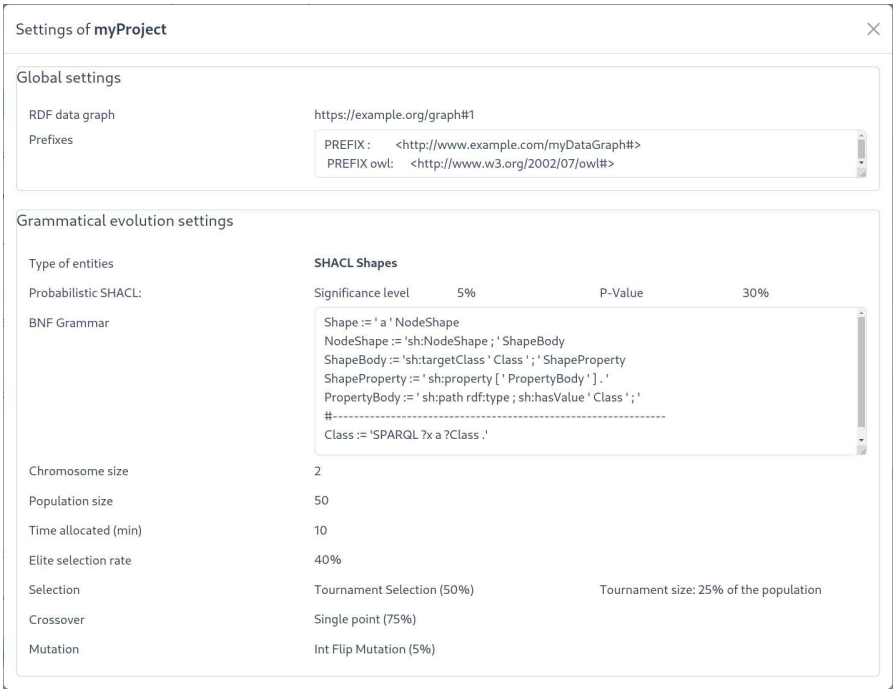


Figure 6.4: `RDFminer` project details popup

- the individual's *phenotype*: **Entity**
- its **reference cardinality**
- its number of violations (or **exceptions**)
- its number of **confirmations**
- its **likelihood** measure *if and only if* the algorithm is used to discover SHACL shapes
- its **fitness** score: based on Eq. (3.6) for axioms and Eq. (5.2) for shapes
- the **acceptance** decision: based on ARI measure (Eq. (3.5)) for axioms and A measure (Eq. (5.1)) for shapes

The visualisations that enable an in-depth analysis of the algorithm's performance are shown in Fig. 6.6 and are organised as follows:

(A) The **population evolution** line chart describes the rate of individuals that differ from one generation to the next one, *i.e.*, the population development rate $\mathcal{P}^{dev}$

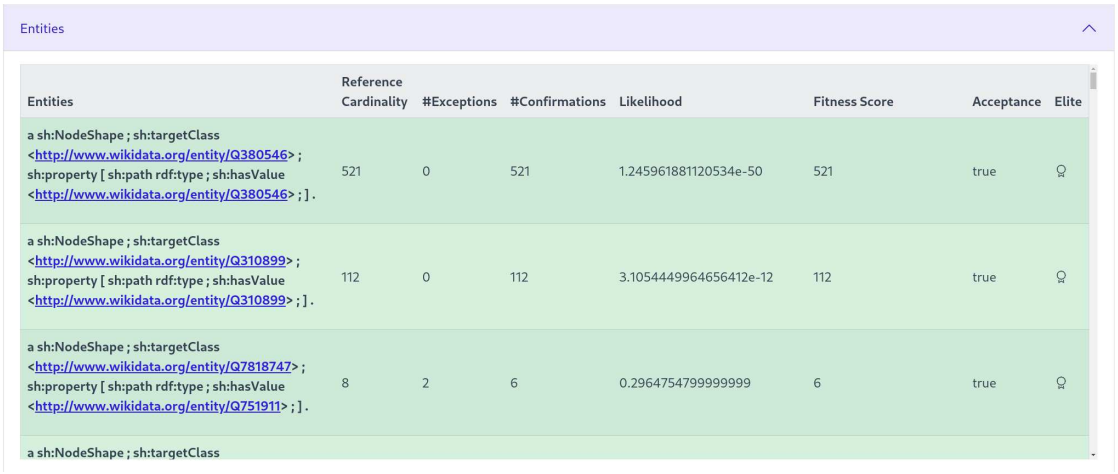


Figure 6.5: Entities (axioms or shapes) table

(see Eq. (5.3)), and the proportion of distinct individuals (*i.e.*, distinct phenotypes) within the population, respectively in blue and in yellow.

(B) The **characteristics of the entities** bubble chart provides information on the quality of the individuals: a colour gradient from red to green indicates the degree to which RDF data conforms to the candidate shape or axiom.

(C) The **individuals with non-null fitness** bar chart enables checking the number of individuals with non-zero fitness over the generations.

(D) The **fitness evolution** line chart shows the average fitness value (in green) for each generation and the median (in yellow) and maximum (in red) fitness values as well: this makes it a more detailed analysis of the individual fitness evolution (and its dispersion).

By providing these metrics to the user on demand, RDFMiner enables a real-time analysis of the mining process, and therefore an effective way of supervising its execution: *e.g.*, the user can decide to stop it if it appears to be stuck in a local optimum, which means that the chosen hyper-parameters (*e.g.*, those chosen in Fig. 6.3) of the evolutionary algorithm do not lead to the discovery of a large set of relevant individuals.

6.4 Conclusion

In this chapter, we presented the `RDFminer` software for the evolutionary discovery of OWL axioms and SHACL shapes from an RDF data graph. Its architecture enables the

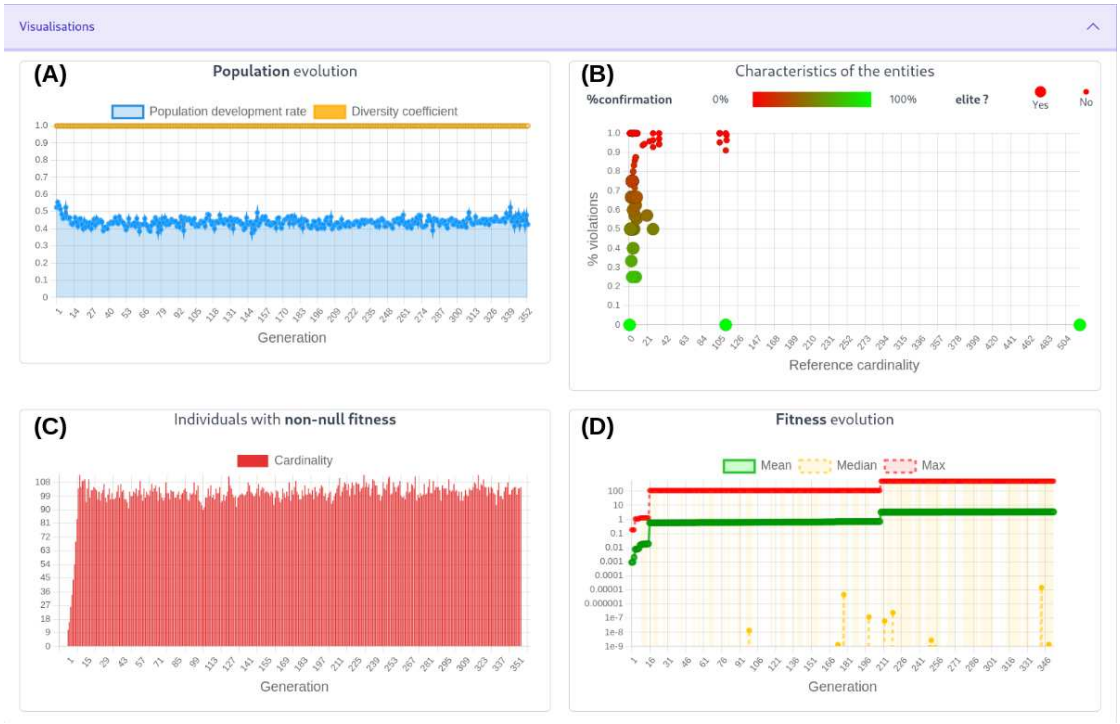


Figure 6.6: Visualisation of the algorithm's results

configuration of the discovery process, launching a project, supervising its progress, and inspecting the results in real-time. The presented charts provide a clear idea of the state of progress of the algorithm in real-time, which can be further developed: *e.g.*, bar chart of recombination statistics for each generation (#crossover, #mutation, ...).

In addition, user tests are to be carried out to assess the software's ergonomics and how well the tool works in the main use cases, enabling us to identify improvement needs and new functionalities arising from discussions.

The source code is available in a public repository[9] and an `RDFminer` service is available online.[10] A tutorial video is available on the `RDFminer` website.[11]

[9] Source code: https://github.com/Wimmics/RDFminer
[10] Web application: https://ns.inria.fr/rdfminer/
[11] Tutorial video: https://ns.inria.fr/rdfminer/tutorial

Chapter 7

Conclusions & Perspectives

7.1 Conclusions

In this thesis, we studied knowledge discovery from RDF data graphs, assuming that this task could be done using a bottom-up approach. To this end, we have proposed an evolutionary approach based on the use of standard semantic Web technologies combined with evolutionary algorithms to:

* *construct* the new knowledge as individuals representing **candidate solutions** to the given problem,

** fairly *assess* these candidate solutions using RDF facts,

*** *discover* a large set of credible solutions by retaining the best individuals over iterations (or generations).

The proposed approach demonstrated that the algorithms based on Grammatical Evolution were suited to the study, and their adaptation to semantic web standards was a relevant working direction. We have defined the research scope of this knowledge in two specific categories: **(1)** Subsumption `SubClassOf` axioms, for which their class expressions can be *complex*, and **(2)** SHACL shapes, composed of *one* `hasValue` constraint. It is important to note that the expressiveness of BNF grammars, a fundamental component of GE, allows the framework of this knowledge to be extended much further.

Chapter 3 focuses on the evolutionary discovery of `SubClassOf` axioms. We started by addressing the central issue of computing exceptions to a subsumption axiom under the "*open-world*" assumption: this assumption significantly limits the use of this heuristic in assessing candidate subsumption axioms and, as a consequence, the evolutionary

discovery of candidate subsumption axioms from large RDF data graph. First, we demonstrated that the computation time issue for subsumption axiom exceptions is mainly due to a large number of duplicate computations during the SPARQL query processing for computing exceptions to a subsumption axiom: the higher the number of instances computed by the SPARQL query, the greater the duplication can be. To respond efficiently to this problem, we have proposed *(1)* a multi-threading system to reduce the global computation time for axiom assessment, *(2)* an extended heuristic to avoid redundant computations, and *(3)* an optimised SPARQL query chunking technique to iterate on a query service call and, by doing so, divide the computation into several tasks. The comparative study we have carried out demonstrates that this approach significantly reduces the computation time for axiom assessment, both globally and locally. This opened up new perspectives for the evolutionary discovery of subsumption axioms. We have, therefore, proposed a BNF grammar for producing candidate subsumption axioms whose class expressions can be *complex*. The evolutionary algorithm used is presented in Chapter 5 and uses the optimisations presented before to assess candidate axioms. The experiments have led to the discovery of a large set of candidate subsumption axioms acceptable through the evolutionary process across a very large RDF data graph. We also observed that some of these candidate axioms are poorly supported by RDF data (*i.e.*, few instances confirm them), which suggests that our algorithm is robust in discovering axioms with few RDF facts to confirm them.

Chapter 4 opens up the prospect of discovering SHACL shapes from RDF facts through a similar evolutionary process. We proposed a framework to include and exploit probabilistic information in SHACL validation reports to take into account inherent errors and incompleteness in RDF data graphs when validating them against SHACL constraints. First, we proposed a probabilistic model based on the assumption that the SHACL validation process follows a *binomial* distribution: from this model, we defined the *likelihood* measure, *i.e.*, the likelihood of the validation results of nodes against a SHACL shape, taking into account the physiological error rate estimated beforehand. Second, we extended the SHACL validation report model to include this likelihood measure and additional metrics computed while validating nodes against SHACL shape(s) for subsequent use through an extended validation report. Lastly, we proposed an acceptance decision model of RDF data based on hypothesis testing to validate or not the null hypothesis H_0, *i.e.*, "*the RDF data* υ_s *follow a binomial distribution with the given error rate*", through the testing for Goodness of Fit. The experiments carried out enabled the physiological error rate of a real-world RDF data graph to be estimated empirically. Moreover, performance testing suggests a slight increase in computation time, which ap-

pears linear. This led to research on the evolutionary discovery of SHACL shapes using the probabilistic framework for assessing candidate shapes.

Regarding the SHACL shapes discovery task, we proposed an algorithm based on Grammatical Evolution in Chapter 5 to discover credible SHACL shapes regarding an RDF data graph. First, we presented an extended method to write BNF grammars that dynamically inject RDF facts as productions by writing SPARQL queries into rules to produce representative candidate shapes[1] as individuals. Second, we define *(1)* an acceptance function of a shape $A(s)$ based on the likelihood measure defined in Chapter 4, and *(2)* a fitness function $F(s)$ that uses the acceptance measure and the number of confirmations as "support" to fairly assess candidate shapes while taking into account the physiological errors in the data graph. Finally, we have presented GE operators and a recombination algorithm to guarantee the diversity of individuals in the population by filtering individuals (based on a phenotypic comparison) and the population development rate to measure their impact on the discovery of candidate shapes. The experiments show that the proposed approach enables the discovery of a large set of acceptable candidate shapes from a real-world RDF data graph, considering a physiological error rate. Moreover, the approach can highlight *very fine-grained* candidate shapes that bring relevant conclusions on RDF facts despite low support and some violations (for most of them).

Finally, Chapter 6 focuses on the `RDFminer` software, which is the main tool used to perform every evolutionary task presented in Chapter 3 and Chapter 5. We presented the architecture of the software and how users can use it to discover candidate shapes or axioms by creating projects through an interactive dashboard, analysing the results of their projects on the dashboard in real-time, and controlling their execution.

Our work shows that an evolutionary approach is relevant for discovering new knowledge from an RDF data graph, considering the scalability, errors, and incompleteness issues. Our results show that the generated candidate axioms or shapes are diverse, from very general individuals to individuals with limited representation, leading to some *surprising* conclusions. It is essential to remember that the results require domain expert validation to rigorously validate the discovered shapes that have been automatically accepted.

[1] This approach is also suitable for the evolutionary discovery of axioms

7.2 Perspectives

Our research work opens up many perspectives, which we present according to three different perspectives: related to *evolutionary algorithms*, *SHACL shapes assessment*, and the `RDFminer` software.

7.2.1 On the Evolutionary Discovery of Knowledge

Novelty Search

Considering our ambition to discover a wide set of credible knowledge from the RDF data graph, the **Novelty Search** technique appears essential to explore. Novelty Search aims to reward candidate solutions for their credibility and *originality*, *i.e.*, the characteristics distinguishing them from others in a population, resulting in different conclusions about individuals among generations [DLC19]. In the context of subsumption axiom discovery, a fitness function which measures the credibility of an axiom (see Eq. (3.6)) and the distance of an axiom from the other candidate axioms in the current population appears to be the most relevant idea. Moreover, the distance between two candidates *SubClassOf* axioms can be designed in different way [MdCPT20], and we suggest that a data-driven based approach provides the most credible similarity explanation between subsumption axioms.

Assess by Predicting the Score of Candidate Axioms or Shapes

Despite the contributions presented in Chapter 3 to reduce the computation time of subsumption axiom's exceptions, some axioms unavoidably involve high computation times, which is related to the number of nodes in an RDF data graph. In the same way, this issue affects the evolutionary discovery of SHACL shapes when the number of nodes to be tested for candidate shapes is very large. Consequently, we assume that another heuristic should perform the individual's assessment task: *e.g.*, Ballout et al. proposed a scalable heuristic to predict the acceptability (*i.e.*, *ARI*) of atomic subsumption axioms [BdCPT24]. A promising perspective is to use their model to automatically assess the *ARI* score of candidate axioms and use the possibilistic heuristics in combination to assess the final population. The search for models to predict the acceptability and fitness of candidate shapes is an interesting perspective.

Advanced Grammatical Evolution Heuristics

To conclude, we believe that it would be interesting to explore an adaptation of our approach with other extended Grammatical Evolution algorithms discussed in Section 2.3: *e.g.*, the structured GE [LPC16b] and/or (more recent) the probabilistic structured GE [MLM22b]. Moreover, searching for RDF data-driven-based heuristics for crossover and mutation operators could significantly improve the exploratory capabilities.

7.2.2 On the Assessment of SHACL Shapes

Shape Assessment at the Constraint Level

The probabilistic measure applies at the level of the shape. When a given shape expresses more than one constraint, this measure considers the violations of nodes for each constraint of the shape and expresses a "global" measure. We believe that a likelihood measure expressed at the constraint level is a promising perspective: *e.g.*, it should enable a more precise understanding of the impact of constraints in shape.

Probabilistic or Possibilistic Models

When analysing the limitations concerning the probabilistic assessment framework, it becomes evident that validating nodes against cardinality constraints implies a dependence between RDF facts: this contradicts the data independence principle within a binomial distribution. Therefore, modelling other probabilistic models (or a possibilistic model) appears crucial to extend its use for cardinality constraints and (overall) to improve the quality of the expressed probabilistic information.

7.2.3 On the Evolution of the `RDFminer` software

Discovering new Knowledge on distant SPARQL Endpoints

The proposed architecture allows the user to discover candidate axioms or shapes from RDF data graphs that are loaded in the Corese semantic Web factory [Cé23] data store. We believe that one of the future challenges is to enable the discovery of axioms or shapes from *remote* data graphs available from SPARQL endpoints, using SPARQL Federated queries to assess axioms against RDF facts [FCFT22]. Concerning the candidate shapes assessment, Corman et al. proposed a method to assess RDF facts from SPARQL endpoints against shapes [CFRS19b, CFRS19a], and the `Trav-SHACL` engine can be used to evaluate remote graphs [FRV21], but does not implement a probabilistic framework.

Towards an implementation of `RDFminer-core` in Python

Considering the proposals for future work, the question of a technological change, *i.e.*, an implementation of our evolutionary algorithm and therefore of the module `RDFminer-core` appears natural and becomes important. First, we noticed that `GEVA 2.0` is no longer maintained, in favour of the `PonyGE2` [FMF+17] Grammatical Evolution toolkit[2], developed in Python. Second, most of the advanced GE algorithms, *e.g.*, Dynamic Structured Grammatical Evolution [LAP+18][3], have been developed in Python. Finally, most of the advanced AI toolkits (*e.g.*, TensorFlow toolkit for Machine Learning[4]) are available in Python: this would open up new perspectives in the development of heuristics using these models, in particular, the possibility of using models to predict the score of candidate axioms [BdCPT24].

[2] Source code: https://github.com/PonyGE/PonyGE2
[3] Source code: https://github.com/nunolourenco/dsge
[4] https://www.tensorflow.org/

Bibliography

[ACO+20] Medina Andresel, Julien Corman, Magdalena Ortiz, Juan L. Reutter, Ognjen Savkovic, and Mantas Simkus. Stable model semantics for recursive shacl. In *Proceedings of The Web Conference 2020*, WWW '20, page 1570–1580, New York, NY, USA, 2020. Association for Computing Machinery.

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. *Mining Association Rules Between Sets of Items in Large Databases, SIGMOD Conference*, volume 22, pages 207–. ACM, 06 1993.

[AKNR21] Muhammad Sarmad Ali, Meghana Kshirsagar, Enrique Naredo, and Conor Ryan. Autoge: A tool for estimation of grammatical evolution models. In *International Conference on Agents and Artificial Intelligence*, 2021.

[AWK+18] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. A survey of ontology learning techniques and applications. *Database*, 2018:bay101, 10 2018.

[BdCPT24] Ali Ballout, Célia da Costa Pereira, and Andrea G. B. Tettamanzi. Scalable Prediction of Atomic Candidate OWL Class Axioms Using a Vector-Space Dimension Reduced Approach. In Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik, editors, *ICAART 2024 - 16th International Conference on Agents and Artificial Intelligence*, volume 3 of *16th ICAART 2024, vol 3*, pages 347–357, Rome, Italy, February 2024. SCITEPRESS - Science and Technology Publications.

[BDFAG19] Iovka Boneva, Jérémie Dusart, Daniel Fernández Alvarez, and Jose Emilio Labra Gayo. Shape Designer for ShEx and SHACL Constraints. ISWC 2019 - 18th International Semantic Web Conference, October 2019. Poster.

[BFH+12] Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, and Michael Smith. Owl 2 web ontology language structural specification and functional-style syntax (second edition). W3C recommendation, W3C, 12 2012. https://www.w3.org/TR/owl2-syntax/.

[BFM00] T. Baeck, D.B. Fogel, and Z. (Eds.). Michalewicz. *Evolutionary Computation 1: Basic Algorithms and Operators (1st ed.)*. CRC Press, 2000.

[BGE14] Dan Brickley and R.V. Guha (Editors). Rdf schema 1.1. W3C recommendation, W3C, 02 2014. https://www.w3.org/TR/rdf-schema/.

[BJVdB22] Bart Bogaerts, Maxim Jakubowski, and Jan Van den Bussche. Expressiveness of shacl features. In *ICDT*, 2022.

[BL12] Lorenz Bühmann and Jens Lehmann. Universal owl axiom enrichment for large knowledge bases. In Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, editors, *Knowledge Engineering and Knowledge Management*, pages 57–71, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[CD22] Andrew Cropper and Sebastijan Dumančić. Inductive logic programming at 30: a new introduction, 2022.

[CF15] Olivier Corby and Catherine Faron. STTL: A SPARQL-based Transformation Language for RDF. In *11th International Conference on Web Information Systems and Technologies*, Lisbon, Portugal, May 2015.

[CFG20] Andrea Cimmino, Alba Fernández-Izquierdo, and Raúl García-Castro. Astrea: Automatic generation of SHACL shapes from ontologies. In *ESWC*, volume 12123 of *Lecture Notes in Computer Science*, pages 497–513. Springer, 2020.

[CFG+21] Olivier Corby, Catherine Faron, Fabien Gandon, Damien Graux, and Franck Michel. Beyond Classical SERVICE Clause in Federated SPARQL Queries: Leveraging the Full Potential of URI Parameters. In *WEBIST 2021 - 17th International Conference on Web Information Systems and Technologies*, Online, Portugal, October 2021.

[CFRS19a] Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savkovic. Shacl2sparql: Validating a sparql endpoint against recursive shacl constraints. In *International Workshop on the Semantic Web*, 2019.

[CFRS19b] Julien Corman, Fernando Florenzano, Juan L. Reutter, and Ognjen Savković. Validating shacl constraints over a sparql endpoint. In Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtěch Svátek, Isabel Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon, editors, *The Semantic Web – ISWC 2019*, pages 145–163, Cham, 2019. Springer International Publishing.

[CKXS17] Cheng-Hao Cai, Dengfeng Ke, Yanyan Xu, and Kaile Su. Symbolic manipulation based on deep neural networks and its application to axiom discovery. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2136–2143, 2017.

[CMRRI24] Leonardo Lucio Custode, Chiara Camilla Migliore Rambaldi, Marco Roveri, and Giovanni Iacca. Comparing large language models and grammatical evolution for code generation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '24 Companion, page 1830–1837, New York, NY, USA, 2024. Association for Computing Machinery.

[CRS18] Julien Corman, Juan L. Reutter, and Ognjen Savković. Semantics and validation of recursive shacl. In Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee, and Elena Simperl, editors, *The Semantic Web – ISWC 2018*, pages 318–336, Cham, 2018. Springer International Publishing.

[CT20] Lucie Cadorel and Andrea Tettamanzi. Mining rdf data of covid-19 scientific literature for interesting association rules. *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 145–152, 2020.

[Cé23] Cérès, Rémi and Corby, Olivier and Gandon, Fabien. Corese software (version 4.5.0). https://github.com/Wimmics/corese, 2023.

[DCFD23] Xuemin Duan, David Chaves-Fraga, and Anastasia Dimou. Xsd2shacl: Capturing rdf constraints from xml schema. In *Proceedings of the 12th*

Knowledge Capture Conference 2023, K-CAP '23, page 214–222, New York, NY, USA, 2023. Association for Computing Machinery.

[DCFDD24] Xuemin Duan, David Chaves-Fraga, Olivier Derom, and Anastasia Dimou. SCOOP all the Constraints' Flavours for your Knowledge Graph. In *ESWC 2024 - 21st International Conference on Semantic Web*, May 2024.

[DDSMO+21] Thomas Delva, Birte De Smedt, Sitt Min Oo, Dylan Van Assche, Sven Lieber, and Anastasia Dimou. Rml2shacl: Rdf generation taking shape. In *Proceedings of the 11th Knowledge Capture Conference*, K-CAP '21, page 153–160, New York, NY, USA, 2021. Association for Computing Machinery.

[DLC19] Stephane Doncieux, Alban Laflaquière, and Alexandre Coninx. Novelty search: a Theoretical Perspective. In *GECCO '19: Genetic and Evolutionary Computation Conference*, pages 99–106, Prague Czech Republic, France, July 2019. ACM.

[DW22] Grant Dick and Peter A. Whigham. Initialisation and grammar design in grammar-guided evolutionary computation, 2022.

[DZP10] Euthymios Drymonas, Kalliopi Zervanou, and Euripides G. M. Petrakis. Unsupervised ontology acquisition from plain texts: The ontogain system. In Christina J. Hopfe, Yacine Rezgui, Elisabeth Métais, Alun Preece, and Haijiang Li, editors, *Natural Language Processing and Information Systems*, pages 277–287, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[EL16] Fajar J. Ekaputra and Xiashuo Lin. Shacl4p: Shacl constraints validation within protégé ontology editor. In *2016 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6, 2016.

[FÀLGGA22] Daniel Fernandez-Àlvarez, Jose Emilio Labra-Gayo, and Daniel Gayo-Avello. Automatic extraction of shapes using shexer. *Knowledge-Based Systems*, 238:107975, 2022.

[FAT98] Katerina T. Frantzi, Sophia Ananiadou, and Jun-ichi Tsujii. The c-value/nc-value method of automatic recognition for multi-word terms. In *Proceedings of the Second European Conference on Research and Advanced*

Technology for Digital Libraries, ECDL '98, page 585–604, Berlin, Heidelberg, 1998. Springer-Verlag.

[FBMR17] Michael Faerber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, 9:1–53, 03 2017.

[FCFT22] Rémi Felin, Olivier Corby, Catherine Faron, and Andrea G. B. Tettamanzi. Optimizing the Computation of a Possibilistic Heuristic to Test OWL SubClassOf Axioms Against RDF Data. In *W-IAT 2022 - IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, Niagara Falls, Canada, November 2022.

[FdE08] Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. Dl-foil concept learning in description logics. In Filip Železný and Nada Lavrač, editors, *Inductive Logic Programming*, pages 107–121, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[FFT23a] Rémi Felin, Catherine Faron, and Andrea G. B. Tettamanzi. A Framework to Include and Exploit Probabilistic Information in SHACL Validation Reports. In *ESWC 2023 - 20th International European Semantic Web Conference*, volume LNCS-13870 of *The Semantic Web 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28–June 1, 2023, Proceedings*, pages 91–104, Hersonissos, Greece, May 2023. Springer Nature Switzerland.

[FFT23b] Rémi Felin, Catherine Faron, and Andrea G. B. Tettamanzi. Un cadre pour inclure et exploiter des informations probabilistes dans les rapports de validation SHACL. In *IC 2023 - 34es Journées francophones d'Ingénierie des Connaissances @ Plate-Forme Intelligence Artificielle (PFIA 2023)*, IC2023 : 34es Journées francophones d'Ingénierie des Connaissances, Strasbourg, France, July 2023.

[FMF+17] Michael Fenton, James McDermott, David Fagan, Stefan Forstenlechner, Erik Hemberg, and Michael O'Neill. Ponyge2: grammatical evolution in python. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, page 1194–1201, New York, NY, USA, 2017. Association for Computing Machinery.

[FMFT24a] Rémi Felin, Pierre Monnin, Catherine Faron, and Andrea G. B. Tettamanzi. An Algorithm Based on Grammatical Evolution for Discovering SHACL Constraints. In *EuroGP 2024 - 27th European Conference on Genetic Programming*, volume 14631 of *Genetic Programming – 27th European Conference, EuroGP 2024*, pages 176–191, Aberystwyth, United Kingdom, April 2024. Springer Nature Switzerland.

[FMFT24b] Rémi Felin, Pierre Monnin, Catherine Faron, and Andrea G. B. Tettamanzi. Extraction probabiliste de formes SHACL à l'aide d'algorithmes évolutionnaires. In *EGC 2024 - Extraction et Gestion de la Connaissance*, volume RNTI-E-40, Dijon, France, January 2024.

[FMFT24c] Rémi Felin, Pierre Monnin, Catherine Faron, and Andrea G. B. Tettamanzi. RDFminer: an Interactive Tool for the Evolutionary Discovery of SHACL Shapes. In *ESWC 2024 - 21th International European Semantic Web Conference*, Hersonissos, Greece, May 2024. Springer Nature Switzerland.

[FRV21] Mónica Figuera, Philipp D. Rohde, and Maria-Esther Vidal. Trav-shacl: Efficiently validating networks of shacl constraints. In *Proceedings of the Web Conference 2021*, WWW '21, page 3337–3348, New York, NY, USA, 2021. Association for Computing Machinery.

[FT21] Rémi Felin and Andrea G. B. Tettamanzi. Using Grammar-Based Genetic Programming for Mining Subsumption Axioms Involving Complex Class Expressions. In *WI-IAT 2021 - 20th IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Melbourne, Australia, December 2021.

[FV11] Daniel Fleischhacker and Johanna Völker. Inductive learning of disjointness axioms. In Robert Meersman, Tharam Dillon, Pilar Herrero, Akhil Kumar, Manfred Reichert, Li Qing, Beng-Chin Ooi, Ernesto Damiani, Douglas C. Schmidt, Jules White, Manfred Hauswirth, Pascal Hitzler, and Mukesh Mohania, editors, *On the Move to Meaningful Internet Systems: OTM 2011*, pages 680–697, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[GTHS13] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in

ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 413–422, New York, NY, USA, 2013. Association for Computing Machinery.

[GTHS15] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Fast Rule Mining in Ontological Knowledge Bases with AMIE+. *The VLDB Journal*, 2015.

[Har10] Robin Harper. Ge, explosive grammars and the lasting legacy of bad initialisation. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

[HPSB+04] Ian Horrocks, Peter Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. W3C member submission, W3C, 05 2004.

[HSP13] Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. Sparql 1.1 query language. W3C recommendation, W3C, 03 2013.

[KA15] Hyun-Tae Kim and Chang Wook Ahn. A new grammatical evolution based on probabilistic context-free grammar. In Hisashi Handa, Hisao Ishibuchi, Yew-Soon Ong, and Kay-Chen Tan, editors, *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems - Volume 2*, pages 1–12, Cham, 2015. Springer International Publishing.

[KAB06] Lobna Karoui, Marie-aude Aufaure, and Nacera Bennacer. Context-based hierarchical clustering for the ontology learning. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pages 420–427, 2006.

[KdSF21] Aljosha Köcher, Luis Miguel Vieira da Silva, and Alexander Fay. Constraint checking of skills using shacl. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pages 1–6, 2021.

[Kee] Alexis Keely. shaclgen 0.2.5.2. https://github.com/uwlib-cams/shaclgen.

[KK17] Dimitris Kontokostas and Holger Knublauch. Shapes constraint language (SHACL). W3C recommendation, W3C, 07 2017. https://www.w3.org/TR/2017/REC-shacl-20170720/.

[Knu17] Holger Knublauch. Shacl and owl compared. Technical report, W3C, 08 2017. https://spinrdf.org/shacl-and-owl.html.

[KS19] Ranjith K Soman. Modelling construction scheduling constraints using shapes constraint language (shacl). In *2019 European Conference on Computing in Construction*, pages 351–358, 07 2019.

[LAP+18] Nuno Lourenço, Filipe Assunção, Francisco B. Pereira, Ernesto Costa, and Penousal Machado. *Structured Grammatical Evolution: A Dynamic Approach*, pages 137–161. Springer International Publishing, Cham, 2018.

[LDV20] Sven Lieber, Anastasia Dimou, and Ruben Verborgh. Statistics about data shape use in RDF data. In Kerry Taylor, Rafael Gonçalves, Freddy Lecue, and Jun Yan, editors, *Proceedings of the 19th International Semantic Web Conference: Posters, Demos, and Industry Tracks*, volume 2721 of *CEUR Workshop Proceedings*, pages 330–335, November 2020.

[LEF18] Rinaldo Lima, Bernard Espinasse, and Fred Freitas. Ontoilper: an ontology- and inductive logic programming-based system to extract entities and relations from text. *Knowl. Inf. Syst.*, 56(1):223–255, jul 2018.

[Leh09] Jens Lehmann. Dl-learner: Learning concepts in description logics. *J. Mach. Learn. Res.*, 10:2639–2642, dec 2009.

[LEO+13] Rinaldo Lima, Bernard Espinasse, Hilário Oliveira, Rafael Ferreira, Luciano Cabral, Dimas Filho, Fred Freitas, and Renê Gadelha. An inductive logic programming-based approach for ontology population from the web. In Hendrik Decker, Lenka Lhotská, Sebastian Link, Josef Basl, and A. Min Tjoa, editors, *Database and Expert Systems Applications*, pages 319–326, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[LFPC17] Nuno Lourenço, Joaquim Ferrer, Francisco B. Pereira, and Ernesto Costa. A comparative study of different grammar-based genetic programming approaches. In James McDermott, Mauro Castelli, Lukas Sekanina, Evert Haasdijk, and Pablo García-Sánchez, editors, *Genetic Programming*, pages 311–325, Cham, 2017. Springer International Publishing.

[LPC16a] Nuno Lourenço, Francisco Pereira, and Ernesto Costa. Unveiling the properties of structured grammatical evolution. *Genetic Programming and Evolvable Machines*, 17, 09 2016.

[LPC16b] Nuno Lourenço, Francisco B. Pereira, and Ernesto Costa. Sge: A structured representation for grammatical evolution. In Stéphane Bonnevay, Pierrick Legrand, Nicolas Monmarché, Evelyne Lutton, and Marc Schoenauer, editors, *Artificial Evolution*, pages 136–148, Cham, 2016. Springer International Publishing.

[LS11] Joel Lehman and Kenneth O. Stanley. *Novelty Search and the Problem with Objectives*, pages 37–56. Springer New York, New York, NY, 2011.

[LSR+20] Martin Leinberger, Philipp Seifer, Tjitze Rienstra, Ralf Lämmel, and Steffen Staab. Deciding shacl shape containment through description logics reasoning. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 366–383, Cham, 2020. Springer International Publishing.

[M+92] Samir W Mahfoud et al. Crowding and preselection revisited. In *PPSN*, volume 2, pages 27–36, 1992.

[MCP24] Huu Tan Mai, Cuong Xuan Chu, and Heiko Paulheim. Do llms really adapt to domains? an ontology learning perspective, 2024.

[MdCPT20] Dario Malchiodi, Célia da Costa Pereira, and Andrea G. B. Tettamanzi. Classifying Candidate Axioms via Dimensionality Reduction Techniques. In Vicenç Torra, Yasuo Narukawa, Jordi Nin, and Núria Agell, editors, *MDAI 2020 - 17th International Conference on Modeling Decisions for Artificial Intelligence*, pages 179–191, Sant Cugat, Spain, September 2020. Springer.

[Med17] Eric Medvet. A comparative analysis of dynamic locality and redundancy in grammatical evolution. In James McDermott, Mauro Castelli, Lukas Sekanina, Evert Haasdijk, and Pablo García-Sánchez, editors, *Genetic Programming*, pages 326–342, Cham, 2017. Springer International Publishing.

[MGAK+20] Franck Michel, Fabien Gandon, Valentin Ah-Kane, Anna Bobasheva, Elena Cabrio, Olivier Corby, Raphaël Gazzotti, Alain Giboin, Santiago Marro, Tobias Mayer, Mathieu Simon, Serena Villata, and Marco Winckler. Covid-on-the-Web: Knowledge Graph and Services to Advance

COVID-19 Research. In *ISWC 2020 - 19th International Semantic Web Conference*, Athens / Virtual, Greece, November 2020.

[MLM22a] Jessica Mégane, Nuno Lourenço, and Penousal Machado. Co-evolutionary probabilistic structured grammatical evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, jul 2022.

[MLM22b] Jessica Mégane, Nuno Lourenço, and Penousal Machado. Probabilistic structured grammatical evolution. In *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, jul 2022.

[MRR+18] Nandana Mihindukulasooriya, Mohammad Rifat Ahmmad Rashid, Giuseppe Rizzo, Raúl García-Castro, Oscar Corcho, and Marco Torchiano. Rdf shape induction using knowledge base profiling. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, page 1952–1959, New York, NY, USA, 2018. Association for Computing Machinery.

[MS04] Alexander Maedche and Steffen Staab. *Ontology Learning*, pages 173–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[MvHE04] Deborah L. McGuinness and Frank van Harmelen (Editors). Owl web ontology language overview. W3C recommendation, W3C, 02 2004. https://www.w3.org/TR/2004/REC-owl-features-20040210/.

[NA18] Miguel Nicolau and Alexandros Agapitos. *Understanding Grammatical Evolution: Grammar Design*, pages 23–53. Springer International Publishing, Cham, 2018.

[NdSL16] Farzad Noorian, Anthony M. de Silva, and Philip H. W. Leong. gramevol: Grammatical evolution in r. *Journal of Statistical Software*, 71(1):1–26, 2016.

[NOB12] Miguel Nicolau, Michael O'Neill, and Anthony Brabazon. Termination in grammatical evolution: grammar design, wrapping, and tails. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[NT19a] Thu Huong Nguyen and Andrea G. B. Tettamanzi. An Evolutionary Approach to Class Disjointness Axiom Discovery. In Payam M. Barnaghi, Georg Gottlob, Yannis Manolopoulos, Theodoros Tzouramanis,

and Athena Vakali, editors, *WI 2019 - IEEE/WIC/ACM International Conference on Web Intelligence*, pages 68–75, Thessaloniki, Greece, October 2019. ACM.

[NT19b] Thu Huong Nguyen and Andrea G B Tettamanzi. Learning Class Disjointness Axioms Using Grammatical Evolution. In Lukás Sekanina, Ting Hu, Nuno Lourenço, Hendrik Richter, and Pablo García-Sánchez, editors, *EuroGP 2019 - 22nd European Conference on Genetic Programming*, Genetic Programming - 22nd European Conference, EuroGP 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24–26, 2019, Proceedings, pages 278–294, Leipzig, Germany, April 2019. Springer.

[NT19c] Thu Huong Nguyen and Andrea G.B. Tettamanzi. An evolutionary approach to class disjointness axiom discovery. In *IEEE/WIC/ACM International Conference on Web Intelligence*, WI '19, page 68–75, New York, NY, USA, 2019. Association for Computing Machinery.

[NT20a] Thu Huong Nguyen and Andrea G. B. Tettamanzi. A Multi-Objective Evolutionary Approach to Class Disjointness Axiom Discovery. In *WI-IAT 2020 - IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, Melbourne/ Virtual, Australia, December 2020.

[NT20b] Thu Huong Nguyen and Andrea G. B. Tettamanzi. Grammatical Evolution to Mine OWL Disjointness Axioms Involving Complex Concept Expressions. In *CEC 2020 - IEEE Congress on Evolutionary Computation*, pages 1–8, Glasgow, United Kingdom, July 2020. IEEE.

[NT20c] Thu Huong Nguyen and Andrea G. B. Tettamanzi. Grammatical Evolution to Mine OWL Disjointness Axioms Involving Complex Concept Expressions. In *CEC 2020 - IEEE Congress on Evolutionary Computation*, pages 1–8, Glasgow, United Kingdom, July 2020. IEEE.

[NT20d] Thu Huong Nguyen and Andrea G. B. Tettamanzi. Using Grammar-Based Genetic Programming for Mining Disjointness Axioms Involving Complex Class Expressions. In Mehwish Alam, Tanya Braun, and Bruno Yun, editors, *ICCS 2020 - 25th International Conference on Conceptual Structures*, volume 12277 of *Lecture Notes in Computer Science*, pages 18–32, Bozen-Bolzano, Italy, September 2020. Springer.

[NVG03] Roberto Navigli, Paola Velardi, and Aldo Gangemi. Ontology learning and its automated terminology translation. *Intelligent Systems, IEEE*, 18:22– 31, 02 2003.

[OBN+04] Michael O'Neill, Anthony Brabazon, Miguel Nicolau, Sean Mc Garraghy, and Peter Keenan. πgrammatical evolution. In Kalyanmoy Deb, editor, *Genetic and Evolutionary Computation – GECCO 2004*, pages 617–629, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[OHG+11] Michael O'Neill, Erik Hemberg, Conor Gilligan, Eliott Bartley, James McDermott, and Anthony Brabazon. GEVA - Grammatical Evolution in Java (v2.0). http://ncra.ucd.ie/GEVA/geva.pdf, 2011.

[OR01] Michael O'Neill and Conor Ryan. Grammatical evolution. *IEEE Trans. Evol. Comput.*, 5(4):349–358, 2001.

[OR04] Michael O'Neill and Conor Ryan. Grammatical evolution by grammatical evolution: The evolution of grammar and genetic code. In Maarten Keijzer, Una-May O'Reilly, Simon Lucas, Ernesto Costa, and Terence Soule, editors, *Genetic Programming*, pages 138–149, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[OTRMH22] Pouya Omran, Kerry Taylor, Sergio Rodríguez Méndez, and Armin Haller. Learning shacl shapes from knowledge graphs. *Semantic Web*, 14:1–21, 09 2022.

[PGR16] Giulio Petrucci, Chiara Ghidini, and Marco Rospocher. Using recurrent neural network for learning expressive ontologies. *ArXiv*, abs/1607.04110, 2016.

[PK22] Paolo Pareti and George Konstantinidis. *A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs*, pages 115–144. Springer International Publishing, Cham, 2022.

[PKM22] Paolo Pareti, George Konstantinidis, and Fabio Mogavero. Satisfiability and containment of recursive SHACL. *J. Web Semant.*, 74:100721, 2022.

[PLGS14] Eric Prud'hommeaux, Jose Emilio Labra Gayo, and Harold Solbrig. Shape expressions: an rdf validation and transformation language. In *Proceedings of the 10th International Conference on Semantic Systems*, SEM '14, page 32–40, New York, NY, USA, 2014. Association for Computing Machinery.

[PLLW15] Baolin Peng, Zhengdong Lu, Hang Li, and Kam-Fai Wong. Towards neural network-based reasoning, 2015.

[PMP+21] Renzo Principe, Andrea Maurino, Matteo Palmonari, Michele Ciavotta, and Blerina Spahiu. Abstat-hd: a scalable tool for profiling very large knowledge graphs. *The VLDB Journal*, 31, 09 2021.

[POL18] H.J. Pandit, D. O'Sullivan, and D. Lewis. Using ontology design patterns to define shacl shapes. In *WOP@ISWC*, pages 67–71, Monterey California, USA, 2018.

[POP01] Raffaele Perego, Salvatore Orlando, and P. Palmerini. Enhancing the apriori algorithm for frequent set counting. In Yahiko Kambayashi, Werner Winiwarter, and Masatoshi Arikawa, editors, *Data Warehousing and Knowledge Discovery*, pages 71–82, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[Qui90] J. R. Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, sep 1990.

[RIV23] Philipp D. Rohde, Enrique Iglesias, and Maria-Esther Vidal. Shacl-acl: Access control with shacl. In Catia Pesquita, Hala Skaf-Molli, Vasilis Efthymiou, Sabrina Kirrane, Axel Ngonga, Diego Collarana, Renato Cerqueira, Mehwish Alam, Cassia Trojahn, and Sven Hertling, editors, *The Semantic Web: ESWC 2023 Satellite Events*, pages 22–26, Cham, 2023. Springer Nature Switzerland.

[RKCJ20] Conor Ryan., Meghana Kshirsagar., Purva Chaudhari., and Rushikesh Jachak. Gets: Grammatical evolution based optimization of smoothing parameters in univariate time series forecasting. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 595–602. INSTICC, SciTePress, 2020.

[RKV+22] Conor Ryan, Meghana Kshirsagar, Gauri Vaidya, Andrew Cunningham, and R. Sivaraman. Design of a cryptographically secure pseudo random number generator with grammatical evolution. *Scientific Reports*, 12:8602, 05 2022.

[RLH22] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. SHACL and shex in the wild: A community survey on validating shapes generation and adoption. In *WWW (Companion Volume)*, pages 260–263. ACM, 2022.

[RLH23a] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Extraction of validating shapes from very large knowledge graphs. *Proc. VLDB Endow.*, 16(5):1023–1032, 2023.

[RLH23b] Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Shactor: Improving the quality of large-scale knowledge graphs with validating shapes. In *Companion of the 2023 International Conference on Management of Data*, SIGMOD '23, page 151–154, New York, NY, USA, 2023. Association for Computing Machinery.

[RO06] Franz Rothlauf and Marie Oetzel. On the locality of grammatical evolution. In Pierre Collet, Marco Tomassini, Marc Ebner, Steven Gustafson, and Anikó Ekárt, editors, *Genetic Programming*, pages 320–330, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[ROC18] Conor Ryan, Michael O'Neill, and JJ Collins. *Introduction to 20 Years of Grammatical Evolution*, pages 1–21. Springer International Publishing, Cham, 2018.

[SKR+19] Xavier Schmitt, Sylvain Kubler, Jérémy Robert, Mike Papadakis, and Yves LeTraon. A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 338–343, 2019.

[SMP18] Blerina Spahiu, Andrea Maurino, and Matteo Palmonari. Towards improving the quality of knowledge graphs with data-driven ontology patterns and shacl. In *WOP@ISWC*, 2018.

[SPP+16] Blerina Spahiu, Riccardo Porrini, Matteo Palmonari, Anisa Rula, and Andrea Maurino. Abstat: Ontology-driven linked data summaries with pattern minimalization. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenić, Sören Auer, and Christoph Lange, editors, *The Semantic Web*, pages 381–395, Cham, 2016. Springer International Publishing.

[SVV08] Vassilis Spiliopoulos, Alexandros G. Valarakos, and George A. Vouros. Csr: Discovering subsumption relations for the alignment of ontologies. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *The Semantic Web: Research and Applications*, pages 418–431, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[TDdNT17] Minh Tran Duc, Claudia d'Amato, Binh Thnanh Nguyen, and Andrea Tettamanzi. An Evolutionary Algorithm for Discovering Multi-Relational Association Rules in the Semantic Web. In Peter A. N. Bosman, editor, *Genetic and Evolutionary Computation Conference (GECCO 2017)*, Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017, pages 513–520, Berlin, Germany, July 2017. ACM SIGEVO, ACM.

[TdTN19] Minh Duc Tran, Claudia d'Amato, Andrea G. B. Tettamanzi, and Binh Thanh Nguyen. Constructing Metrics for Evaluating Multi-Relational Association Rules in the Semantic Web from Metrics for Scoring Association Rules. In Marc Bui, Nhan Le Thanh, and Hung Võ Trùng, editors, *IEEE-RIVF 2019 - International Conference on Computing and Communication Technologies*, pages 65–70, Da Nang, Vietnam, March 2019. IEEE.

[TFG15] Andrea G.B. Tettamanzi, Catherine Faron, and Fabien L. Gandon. Dynamically time-capped possibilistic testing of subclassof axioms against rdf data to enrich schemas. In Ken Barker and José Manuel Gómez-Pérez, editors, *K-CAP*, number 7 in Proceedings of the 8th International Conference on Knowledge Capture, Palisades, NY, United States, October 2015.

[TFG17] Andrea Tettamanzi, Catherine Faron, and Fabien Gandon. Possibilistic testing of OWL axioms against RDF data. *International Journal of Approximate Reasoning*, 2017.

[TKS02] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 32–41, New York, NY, USA, 2002. Association for Computing Machinery.

[VIK+22] Gauri Vaidya, Luise Ilg, Meghana Kshirsagar, Enrique Naredo, and Conor Ryan. Hyperestimator: Evolving computationally efficient cnn models with grammatical evolution. In *ICSBT*, pages 57–68, 2022.

[VKR23] Gauri Vaidya, Meghana Kshirsagar, and Conor Ryan. Grammatical evolution-driven algorithm for efficient and automatic hyperparameter optimisation of neural networks. *Algorithms*, 16(7), 2023.

[VN11] Johanna Völker and Mathias Niepert. Statistical schema induction. In Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Pan, editors, *The Semantic Web: Research and Applications*, pages 124–138, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[WRMH+20] Jesse Wright, Sergio José Rodríguez Méndez, Armin Haller, Kerry Taylor, and Pouya G. Omran. Schímatos: A shacl-based web-form generator for knowledge graph editing. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 65–80, Cham, 2020. Springer International Publishing.

[XHGI20] Ziwei Xu, Mounira Harzallah, Fabrice Guillet, and Ryutaro Ichise. Towards a term clustering framework for modular ontology learning. In Ana Fred, Ana Salgado, David Aveiro, Jan Dietz, Jorge Bernardino, and Joaquim Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 178–201, Cham, 2020. Springer International Publishing.

[YC06] Yanbin Ye and Chia-Chu Chiang. A parallel apriori algorithm for frequent itemsets mining. In *Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)*, pages 87–94, 2006.

[Zad99] Lotfi A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 100:9–34, 1999.

[ZPP14] Qiang Zeng, Jignesh M. Patel, and David Page. Quickfoil: scalable inductive logic programming. *Proc. VLDB Endow.*, 8(3):197–208, nov 2014.

[ZVFD24] Tsitsi Zengeya and Jean Vincent Fonou-Dombeu. A review of state of the art deep learning models for ontology construction. *IEEE Access*, 12:82354–82383, 2024.

[ZY22] Alebachew Zewdu and Betselot Yitagesu. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9, 01 2022.

List of Figures

List of Tables

List of Algorithms

Appendix

Appendix

A Candidate Subsumption Axioms

Table A.1: Discovered subsumption axioms where $|\upsilon_\phi^+| > 0$ sorted by their ARI value

Axiom (ϕ)	ARI_ϕ	$\lvert\upsilon_\phi\rvert$	$\lvert\upsilon_\phi^+\rvert$	$\lvert\upsilon_\phi^-\rvert$	existing ?
SubClassOf(dbo:GridironFootballPlayer dbo:Athlete)	1	6174	6174	0	Yes
SubClassOf(dbo:BaseballTeam dbo:SportsTeam)	1	370	370	0	Yes
SubClassOf(dbo:Cheese dbo:Food)	1	325	325	0	Yes
SubClassOf(dbo:Brewery dbo:Company)	1	62	62	0	Yes
SubClassOf(dbo:Baronet dbo:Person)	0.978	741	587	0	**No**
SubClassOf(dbo:Historian dbo:Agent)	0.882	710	376	0	**No**
SubClassOf(dbo:ScreenWriter dbo:Agent)	0.59	598	115	0	**No**
SubClassOf(ObjectAllValuesFrom(dbo:artist dbo:TelevisionShow) dbo:Work)	0.22	82	2	0	**No**
SubClassOf(ObjectSomeValuesFrom(dbp:platforms dbo:Company) dbo:Work)	0.206	93	2	0	**No**
SubClassOf(dbo:RugbyClub dbo:Agent)	0.196	2171	42	0	**No**
SubClassOf(dbo:SnookerPlayer dbo:Agent)	0.139	310	3	0	**No**
SubClassOf(ObjectSomeValuesFrom(dbo:managerClub dbo:Organisation) dbo:Person)	0.129	841	7	0	**No**
SubClassOf(dbo:Astronaut dbo:Agent)	0.125	635	5	0	**No**
SubClassOf(ObjectAllValuesFrom(dbo:artist dbo:Group) dbo:Work)	0.094	909	4	0	**No**
SubClassOf(dbo:Amphibian dbo:Species)	0.074	4752	13	0	**No**
SubClassOf(dbo:Reptile dbo:Species)	0.072	5422	14	0	**No**
SubClassOf(dbo:SumoWrestler dbo:Athlete)	0.066	463	1	0	**No**
SubClassOf(ObjectSomeValuesFrom(dbp:platforms dbo:Software) dbo:Work)	-0.042	6775	56	6	**No**
SubClassOf(ObjectAllValuesFrom(dbp:platforms dbo:Software) dbo:Work)	-0.056	3224	22	5	**No**
SubClassOf(ObjectAllValuesFrom(dbp:platforms dbo:Software) dbo:Software)	-0.157	3224	23	40	**No**
SubClassOf(ObjectSomeValuesFrom(dbp:platforms dbo:Software) dbo:Software)	-0.188	6775	58	121	**No**
SubClassOf(ObjectSomeValuesFrom(dbp:builder dbo:MilitaryUnit) dbo:Bridge)	-0.558	135	2	23	**No**
SubClassOf(ObjectAllValuesFrom(dbp:builder dbo:MilitaryUnit) dbo:Bridge)	-0.562	133	2	23	**No**
SubClassOf(ObjectSomeValuesFrom(dbp:seasonTopscorer dbo:SoccerPlayer) dbo:SportsTeamMember)	-0.909	2628	37	1534	**No**
SubClassOf(ObjectAllValuesFrom(dbp:seasonTopscorer dbo:SoccerPlayer) dbo:SportsTeamMember)	-0.909	2533	36	1478	**No**

B Extension of the SHACL Validation Report Model

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix psh: <http://ns.inria.fr/probabilistic-shacl/> .
@prefix : <http://www.example.com/myDataGraph#> .

:v1 a sh:ValidationResult ;
    sh:focusNode :n1 ;
    [...]
    sh:sourceShape :s1 .

:v2 a sh:ValidationResult ;
    sh:focusNode :n2 ;
    [...]
    sh:sourceShape :s1 .

[ a sh:ValidationReport ;
    sh:conforms false ;
    sh:result :v1 ;
    sh:result :v2 ;
    [...]
    # SHACL Extension
    # shape s1
    psh:summary [
        a psh:ValidationSummary ;
        psh:generality "0.2"^^xsd:decimal ;
        psh:numConfirmation 178 ;
        psh:numViolation 22 ;
        psh:likelihood "0.081"^^xsd:decimal ;
        psh:referenceCardinality 200 ;
        psh:focusShape :s1
    ] ;
] .
```

Figure B.1: Example of an extended SHACL validation report for a shape `:s1` with $|\upsilon| = 1,000$ and $p = 0.1$

C An algorithm based on Grammatical Evolution for Discovering SHACL Constraints

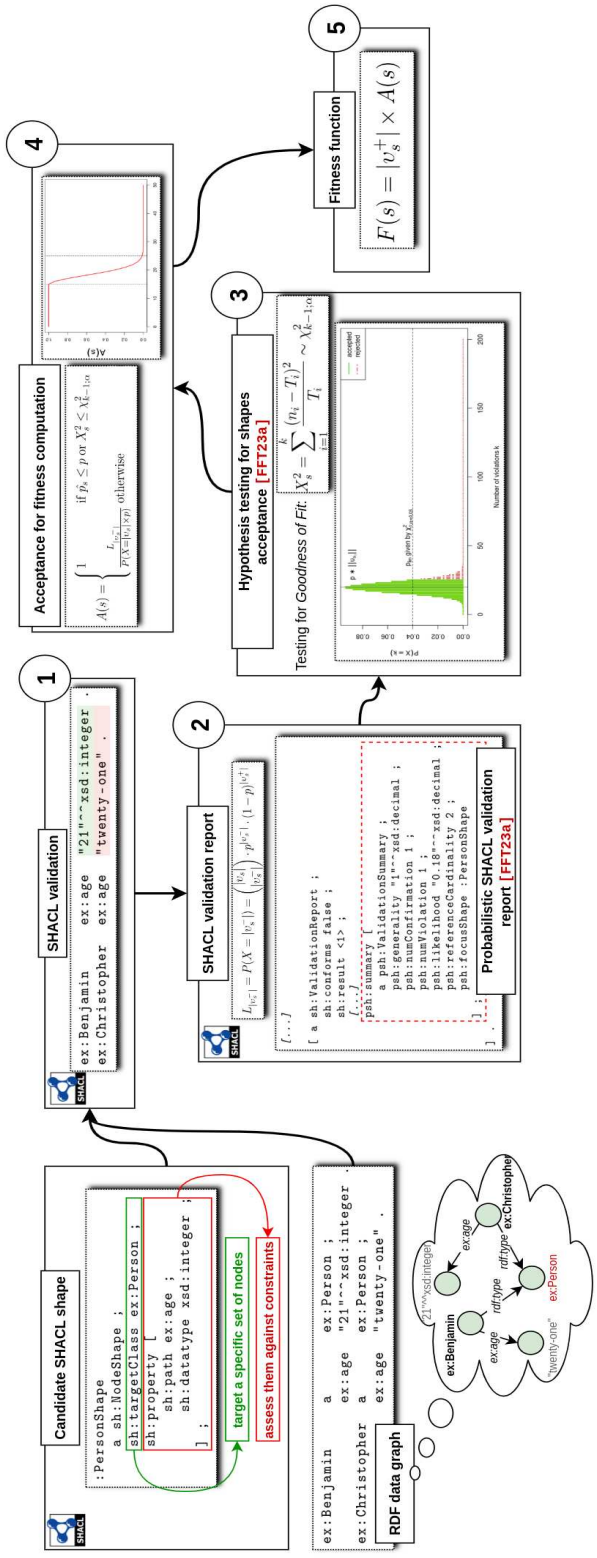


Figure C.2: Overview of the fitness computation process for candidate shapes

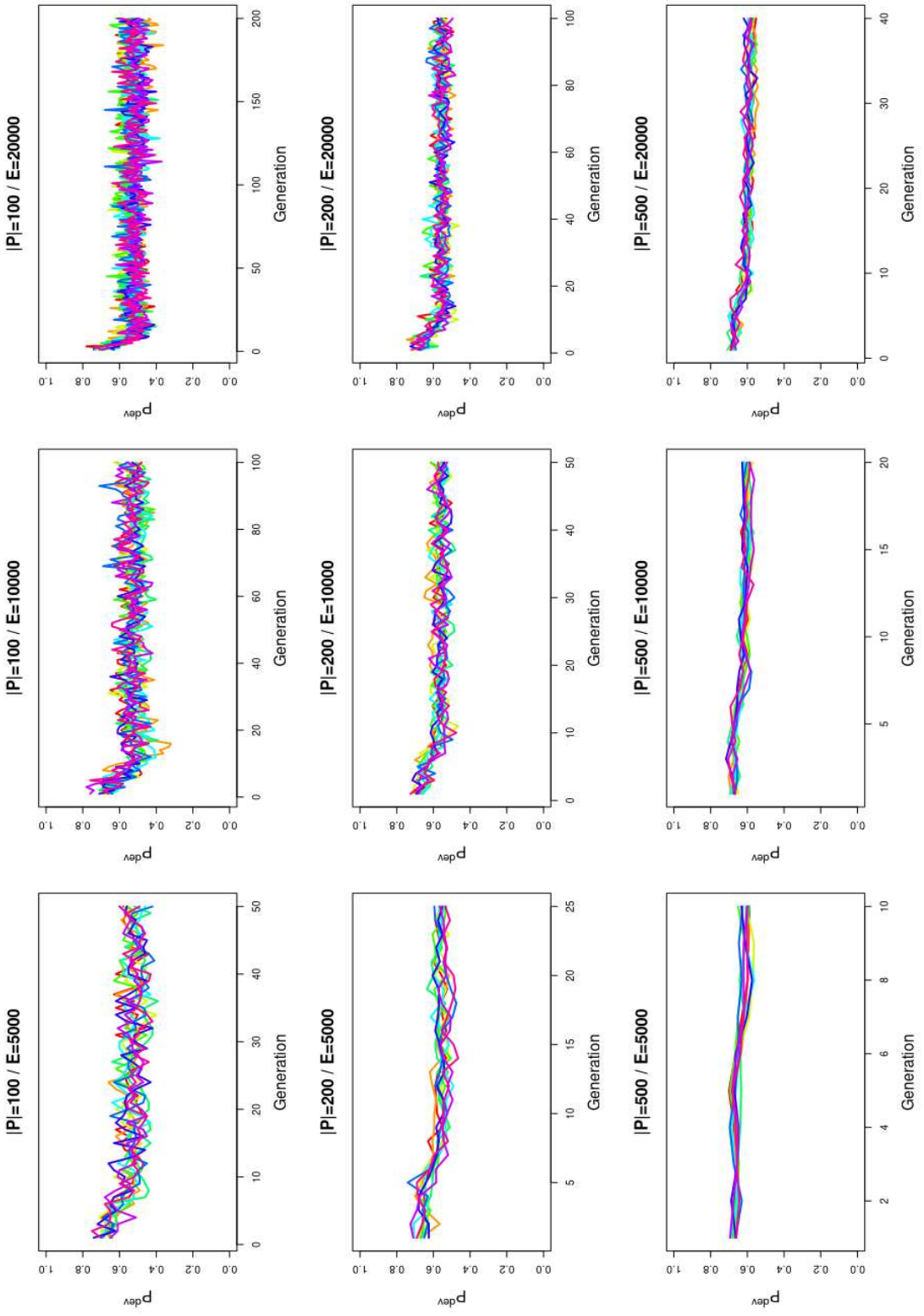


Figure C.3: Evolution of the population development rate P^{dev} over 10 executions

Découverte évolutive de connaissance à partir de graphes de données RDF

Rémi Felin

Résumé

Les graphes de connaissance sont des collections de descriptions interconnectées d'entités (objets, événements ou concepts). Ils mettent les données en contexte par le biais de liens sémantiques, fournissant ainsi un cadre pour l'intégration, l'unification, l'analyse et le partage des données. Aujourd'hui, nous disposons d'un grand nombre de graphes de connaissance riches en données factuelles, dont la construction et l'enrichissement est une tâche relativement bien maîtrisée. Ce qui est plus difficile et plus coûteux, c'est de doter ces graphes de schémas, règles et contraintes qui permettent de vérifier leur cohérence et de déduire des connaissances implicites par raisonnement. Cette thèse présente une approche basée sur la technique d'évolution grammaticale pour la découverte automatique de nouvelles connaissances à partir d'un graphe de données représenté en RDF. Cette approche repose sur l'idée que les connaissances candidates sont générées à partir d'un mécanisme heuristique (exploitant les données du graphe), testés contre les données du graphe, et évoluent à travers un processus évolutionnaire de sorte à ce que seules les connaissances candidates les plus crédibles soient conservées. Dans un premier temps, nous nous sommes concentrés sur la découverte d'axiomes OWL qui permettent, par exemple, d'exprimer des relations entre concepts et d'inférer, à partir de ces relations, de nouvelles informations factuelles. Les axiomes candidats sont évalués à partir d'une heuristique existante basée sur la théorie des possibilités, permettant de considérer l'incomplétude des informations d'un graphe de données. Cette thèse présente les limites de cette heuristique et une série de contributions permettant une évaluation significativement moins coûteuse en temps de calcul. Cela a permis l'évaluation efficace d'axiomes candidats lors du processus évolutif, nous menant ainsi à la découverte d'un grand nombre d'axiomes candidats pertinents vis-à-vis d'un graphe de données RDF. Dans un second temps, nous avons proposé une approche pour la découverte de shapes SHACL qui expriment des contraintes que les données RDF doivent respecter. Elles sont utiles pour contrôler la cohérence (par exemple, structurelle) des données du graphe et facilitent l'intégration de nouvelles données. L'évaluation de shapes candidates repose sur l'évaluation SHACL des données vis-à-vis de ces formes, à laquelle nous ajoutons un cadre probabiliste pour prendre en compte les erreurs et l'incomplétude inhérente des graphes de données lors de l'évaluation de shapes candidates. Enfin, nous présentons `RDFminer`, une application Web open-source permettant d'exécuter notre approche pour découvrir des axiomes OWL ou des formes SHACL à partir d'un graphe de données RDF. L'utilisateur peut contrôler l'exécution et analyser les résultats en temps réels à travers une interface graphique interactive. Les résultats obtenus montrent que l'approche proposée permet de découvrir un large ensemble de nouvelles connaissances crédibles et pertinentes à partir de graphes de données RDF volumineux.