RDFminer: an Interactive Tool for the Evolutionary Discovery of SHACL Shapes

Rémi Felin, Pierre Monnin, Catherine Faron, Andrea G. B. Tettamanzi

RDFminer: an Interactive Tool for the Evolutionary Discovery of SHACL Shapes

Rémi Felin[1 [0000−0003−2532−7555]], Pierre Monnin[1 [0000−0002−2017−8426]], Catherine Faron[1 [0000−0001−5959−5561]], and Andrea G. B. Tettamanzi[1 [0000−0002−8877−4654]]

Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France
{name.surname}@inria.fr

Abstract. RDFminer is an open source Web application to automatically discover SHACL shapes through an evolutionary process. It takes an RDF data graph as input, from which shapes are mined and assessed using a probabilistic validation framework. The user can interact with RDFminer through a dashboard where they can launch and monitor the mining of shapes, and analyse the results in real time.

Keywords: RDF · SHACL · Shape Mining · Evolutionary Algorithm

1 Introduction

The continuous intensive production of RDF facts on the Web contributes to the availability of large knowledge graphs. Subsequently, the problem of inconsistencies in RDF data resulting from these efforts has emerged, which directly impacts the RDF data graph quality, validity and actionability. To identify inconsistencies in RDF data, the SHACL W3C recommendation allows to express constraints as *shapes* that RDF data must conform to. This shifts the problem to determining the domain constraints to be checked: it is well-known that acquiring SHACL shapes from large RDF data graphs is a tedious task [5]. In this paper, we present `RDFminer`, a Web application that makes it possible to discover SHACL shapes from an RDF graph. It implements an evolutionary approach and provides an interactive interface enabling the user to launch, monitor and analyse their shape discovery projects. Fig. 1 presents the whole architecture of `RDFminer`.

2 Evolutionary Discovery of SHACL Shapes

`RDFminer` is a framework implementing the evolutionary approach based on Grammatical Evolution described in [3] to discover relevant SHACL shapes from an RDF data graph. The principle of this approach is to generate and manage a population of candidate shapes that evolve through mutation and crossover, with the aim of improving their fitness, *i.e.*, their adequacy to the data graph, over time. The main steps of the algorithm are presented in Fig. 2

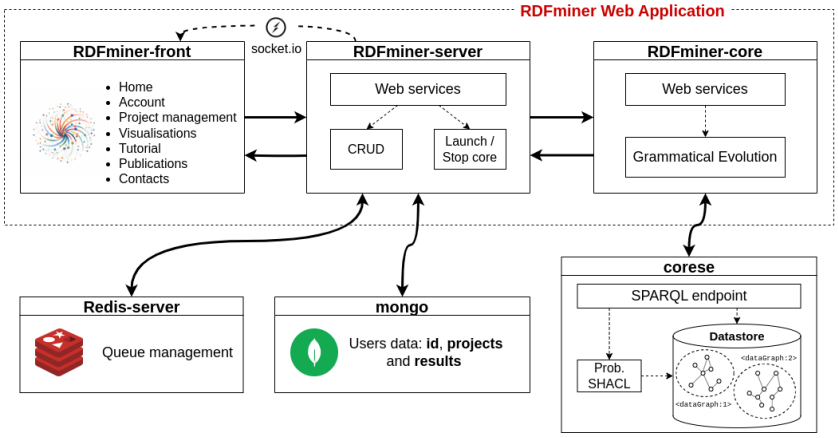

Fig. 1: Global architecture of *RDFminer*

The fitness of the shapes is calculated using a probabilistic framework for SHACL validation presented in [2]. That framework extends standard SHACL validation by declaring RDF graphs as valid w.r.t. a shape if they contain less than a given threshold of triples that do not conform to this shape. SHACL validation reports are extended accordingly with probabilistic metrics, using an extended vocabulary.[1]

`RDFminer` relies on the `GEVA 2.0` [4] implementation of Grammatical Evolution for the generation of candidate SHACL shapes and on the `Corese` [1] semantic Web factory to query RDF data and validate RDF data against candidate shapes. We implemented a multi-threading system to assess candidate shapes as this is the most time-consuming task in the overall evolutionary discovery process.

3 A Web Application to Discover SHACL Shapes

Exploiting the `RDFminer` core engine to discover SHACL shapes is essentially a trial-and-error process. That is why we developed a Web application to provide users with an interface that allows them to control the mining process interactively: it enables to parameterize and launch the discovery process, monitor its execution, inspect and analyze its results.

3.1 Monitoring Dashboard

The connected user can discover shapes from a given RDF data graph by creating a project and defining the parameters of the mining process: the data graph,

[1] Probabilistic SHACL vocabulary: http://ns.inria.fr/probabilistic-shacl/

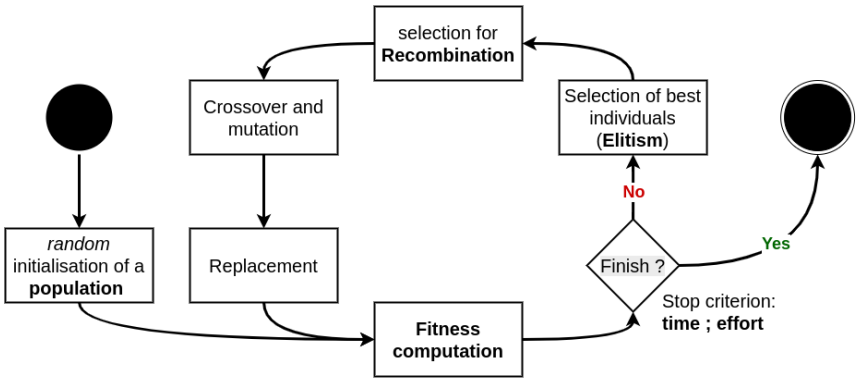

Fig. 2: Main steps of the Grammatical Evolution algorithm

the SHACL constructs to be considered, and the hyper-parameters of the Grammatical Evolution algorithm. The status of the running project is updated in real time and can be interrupted if needed. This dashboard is presented in Fig. 3*a*.

3.2 Result Analysis Dashboard

Due to the nature of the evolutionary mining process, the population of candidate shapes evolves continuously. This dashboard thus allows the user to consult and analyze results both in real-time when evolution is underway or at the end of the process.

In more detail, every running project generates in real-time results which can be analysed through this dashboard. Completed projects are accessible as well. The user accesses the Results view and can analyse the current execution status (Fig. 3b), the discovered SHACL shapes and their characteristics (Fig. 3c) and the execution statistics as charts (Fig. 3d). The *population evolution* chart describes the rate of individuals (candidate shapes) that differ from one generation to the next one: this should be interpreted together with the *individuals with non-null fitness* and *fitness evolution* charts to determine if the chosen hyper-parameters of the evolutionary algorithm lead to the discovery of relevant shapes. The *characteristics of the entities* chart provides information on the quality of the shapes: a colour gradient from red to green indicates the degree to which RDF data conform to the shapes. This real-time analysis of the mining process is an effective way of supervising its execution. For instance, the user can decide to stop it (Fig. 3a) if it appears to be stuck in a local optimum. At the end of an execution, the user can download the shapes graph in Turtle format and/or the

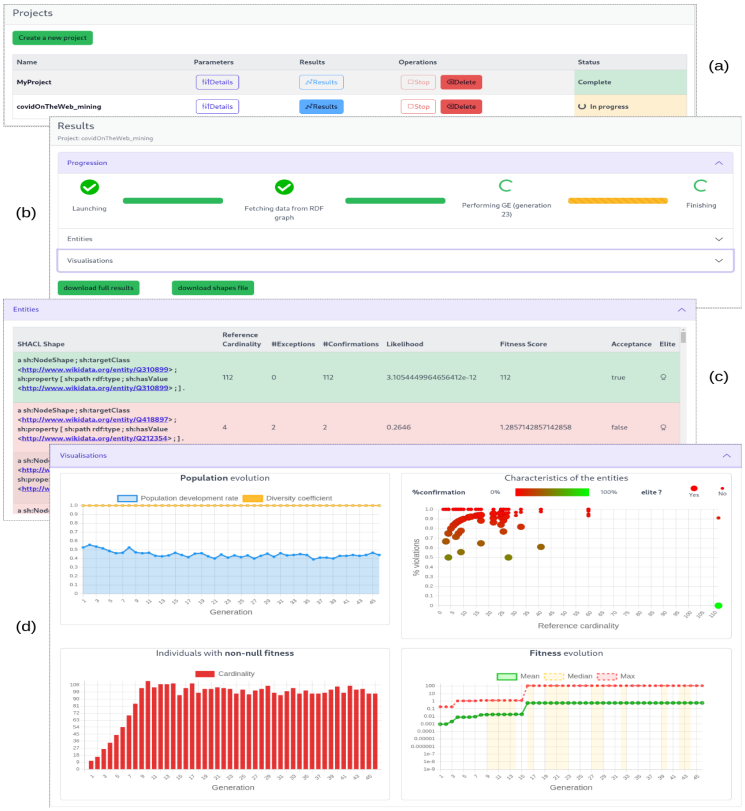

Fig. 3: Visualization dashboard of a shape mining project in progress

complete results file (including individuals, their statistics and the algorithm's statistics) in JSON format for post-processing.

It should also be noted that `RDFminer` core can be used independently of the other components through its API.[2] The source code is available in a public repository[3] and an `RDFminer` service is available online.[4]

4 Proposed Demonstration

The demonstration will be as follows: We will connect to the `eswc_demo` account and show the results of a completed project aiming to discover shapes from the covid-on-the-web RDF data graph [3]. Then we will create and launch a new similar mining project but with less demanding hyper-parameters so that it can

[2] User guide: https://github.com/Wimmics/RDFminer/tree/main/RDFminer-core
[3] Source code: https://github.com/Wimmics/RDFminer
[4] Web application: https://ns.inria.fr/rdfminer/

complete within a few minutes and we will show how the user can visualize in real-time the current state of the project: the current list of shapes discovered, the development rate, the proportion of individuals with a non-zero fitness score and the evolution of the fitness score. On completion of the project, we will download the results file and the shapes graph. A tutorial video corresponding to this demonstration is available on the `RDFminer` website.[5]

As future work, we aim to conduct a user evaluation of both the quality of the generated shapes considered as valid, with a special focus on shapes with a little support (*i.e.*, few triples that confirm them), and the usability of the RDFminer dashboard to monitor and tune the shape mining process.

Acknowledgements This work has been partially funded by the 3IA Côte d'Azur "Investments in the Future" project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

References

1. Cérès, R., et al.: Corese (2023), https://project.inria.fr/corese/
2. Felin, R., et al.: A Framework to Include and Exploit Probabilistic Information in SHACL Validation Reports. ESWC (2023)
3. Felin, R., et al.: An Algorithm Based on Grammatical Evolution for Discovering SHACL Constraints. EuroGP (2024)
4. O'Neill, M., et al.: GEVA - Grammatical Evolution in Java (2011)
5. Rabbani, K., Lissandrini, M., Hose, K.: SHACL and shex in the wild: A community survey on validating shapes generation and adoption. In: WWW (Companion Volume) (2022)

[5] Tutorial video: https://ns.inria.fr/rdfminer/tutorial