



Original software publication

Rhizomer: Interactive semantic knowledge graphs exploration

Roberto García^{a,*}, Juan-Miguel López-Gil^b, Rosa Gil^a^a Universitat de Lleida, Jaume II 69, 25001 Lleida, Spain^b University of the Basque Country, Paseo Manuel de Lardizabal 1, 20018 Donostia-San Sebastián, Spain

ARTICLE INFO

Article history:

Received 27 June 2022

Received in revised form 19 September 2022

Accepted 14 October 2022

Keywords:

Knowledge graph

Semantic data

Visualization

User interface

ABSTRACT

Rhizomer helps researchers and practitioners explore knowledge graphs available as Semantic Web data by performing the three data analysis tasks: overview, zoom and filter, and details-on-demand. This approach makes it easier for users to get an idea about the overall structure and intricacies of a dataset, when compared to existing approaches and even without prior knowledge. Rhizomer is helpful for data reusers, who want to know about the reuse opportunities of a given dataset, and for knowledge graph creators, who can check if the generated data follow their expectations. Rhizomer has been applied in many scenarios, from research and commercial projects to teaching.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Code metadata

Current code version

Permanent link to code/repository used for this code version

Permanent link to Reproducible Capsule

Legal Code License

Code versioning system used

Software code languages, tools, and services used

Compilation requirements, operating environments & dependencies

If available Link to developer documentation/manual

Support email for questions

RhizomerEye v0.2.1 and RhizomerAPI v0.2.1

<https://github.com/ElsevierSoftwareX/SOFTX-D-22-00168>

GPL-3.0, <https://github.com/rhizomik/rhizomerAPI/blob/master/LICENSE> and <https://github.com/rhizomik/rhizomerEye/blob/master/LICENSE>

git

RhizomerEye: TypeScript, Angular. RhizomerAPI: Java, Spring.

RhizomerEye: Node 12 and RhizomerAPI: Java 11

<https://github.com/rhizomik/rhizomerAPI/blob/master/README.md>contact@rhizomik.net

Software metadata

Current software version

Permanent link to executables of this version

Permanent link to Reproducible Capsule

Legal Software License

Computing platforms/Operating Systems

Installation requirements & dependencies

If available, link to user manual — if formally published include a reference to the publication in the reference list

Support email for questions

RhizomerEye v0.2.1 and RhizomerAPI v0.2.1

Docker images available, RhizomerEye

<https://hub.docker.com/repository/docker/rhizomik/rhizomer-eye> andRhizomerAPI: <https://hub.docker.com/repository/docker/rhizomik/rhizomer-api>

GPL-3.0, <https://github.com/rhizomik/rhizomerAPI/blob/master/LICENSE> and <https://github.com/rhizomik/rhizomerEye/blob/master/LICENSE>

Web based

Recommended to use Docker, as detailed in the installation instructions

<https://github.com/rhizomik/rhizomerAPI/blob/master/README.md#install>contact@rhizomik.net

* Corresponding author.

E-mail address: roberto.garcia@udl.cat (Roberto García).

Motivation and significance

The Semantic Web provides the tools and mechanisms to build a Web of data that facilitates data integration at the Web scale while providing access, in a unified way, through a standard query language. Compared to the so-called Web 2.0, the main distinction is that rather than working with a sizable amount of scattered data that needs some level of human interpretation to process it, data is integrated and conceptualised in a way that computers themselves can “understand” it and extract new knowledge [1].

However, neither the Semantic Web nor the software agents are currently entirely functional for the following tasks targeted by this initiative. Extracting meaningful information from research documents [2], providing tools for computing and extracting ideas from the Semantic Web [3], semantic aware graph-based partitioning [4], and effective query performance in terms of query execution time and data scalability by the use of data storage solutions [5] remain research topics in this field. From a trends perspective, topics such as linked data, open data, and data sources have increased in importance over the years. On the other hand, its interactions with other research areas highlight the cross-disciplinary nature of the Semantic Web [6].

Accessing the Semantic Web is still mostly a human activity. Even if some degree of integration is achieved, managing and effectively updating such an integrated framework may require significant human resources, be extremely time-consuming and difficult to fully automate. This access is through the user interfaces and is primarily used for information-seeking tasks. A popular study area is the interactive Semantic Web, where technologies are created using a human-centred approach that attempts to improve human engagement with semantic data, rather than simply enhancing the efficacy and accuracy of automated algorithms and processes. The distinctive characteristics that a Semantic Web user interface needs to support were analysed in [7], while [8] reviews research on user interfaces, visualizations, and interaction methods from relevant Semantic Web venues.

Structured data called knowledge graphs, which are frequently defined using the Semantic Web standard Resource Description Framework [1], depict entities and the interactions between them in the form of a graph. In particular, when graphs from several data sources need to be combined, it could be challenging for lay users to examine existing knowledge graphs [9]. However, there are still issues with efficiency, usability, and scalability when extending the semantic search to document structure and external, formal knowledge sources [10]. Efforts towards interactive and intuitive Semantic Web explorers are being made in multiple areas, such as life sciences, geosciences, digital humanities, healthcare, or defence [11].

This work presents Rhizomer, a tool to easily explore knowledge graphs available as semantic data that provides a head-start for researchers, practitioners, or even lay users, without requiring prior knowledge of the data structure or the underlying Semantic Web technologies.

Software description

Through Rhizomer, a knowledge graph can be explored without requiring prior knowledge about the dataset structure or the underlying semantic data and query languages. Rhizomer is available as an open-source web application with a backend (RhizomerAPI) and a frontend (RhizomerEye).

To facilitate the exploration, Rhizomer supports the three classical data analysis tasks proposed by Shneiderman [12]: getting an overview of the data, zooming and filtering, and viewing

details on demand. Each of these tasks, further detailed in the following subsections, is supported by the following set of features implemented by Rhizomer:

- **Overview**
 - **Word Cloud:** overview the classes in a dataset through a word cloud with the names of the classes and where their size is proportional to the number of instances of each class.
 - **Network:** an overview of the main classes, and relationships among them, using a network representation which includes classes and relationships names.
- **Zoom and Filter**
 - **Classes Autocomplete:** input field, with autocomplete based on the name of the classes instantiated in the dataset, to choose the class to focus on.
 - **Global Text Search:** input field to search across the dataset for instances related to literals containing the typed text, or resource whose label contains that text.
 - **Facet Values Filter:** facet view showing the 10 most common values for the property corresponding to the facet and constrained to the focused class. Clicking any of them filters the displayed instances to those featuring that value for the property.
 - **Facet Values Autocomplete:** input field to filter the list of instances for the focused class using any of the values for the corresponding property. Typed text is autocompleted to the available values for the facet.
 - **Numeric Range Facet:** for facets with numeric ranges (like integer, decimal or year), the minimum and maximum values are shown together with a slider to filter instances based on a user-defined range of values.
 - **Class Text Search:** input field for text search among all facet values for the class currently the focus.
- **Details-on-demand**
 - **Instance Metadata View:** once a particular instance is selected, all the metadata describing it is presented. This includes all direct properties and values. Labels are used instead of URI identifiers when they are available.
 - **Linked Data Browser:** despite labels being shown instead of URIs for relationships, they are displayed as links. When clicked, it is checked if the URI can resolve to RDF data and thus exploration can continue by rendering it using the Instance Metadata View. This way, Rhizomer also works as a Linked Data browser. For instance, if a DBpedia URI for a city is used in the data being explored, it can be clicked to retrieve the metadata describing the city.
 - **Inverse Facet:** facet corresponding to an inverse property for the currently displayed instance. For instance, for a person, a facet to explore the publications that person has authored.

Other Rhizomer’s features, which are not specific to a particular task are:

- **Multilingual Support:** Rhizomer uses labels, whenever they are available, instead of their identifiers (URIs or fragments of them) to refer to classes, properties and resources in the data being explored. Moreover, it gives preference to those in the language selected by the user if labels have a language tag. Otherwise, it defaults to untagged labels or those in English.

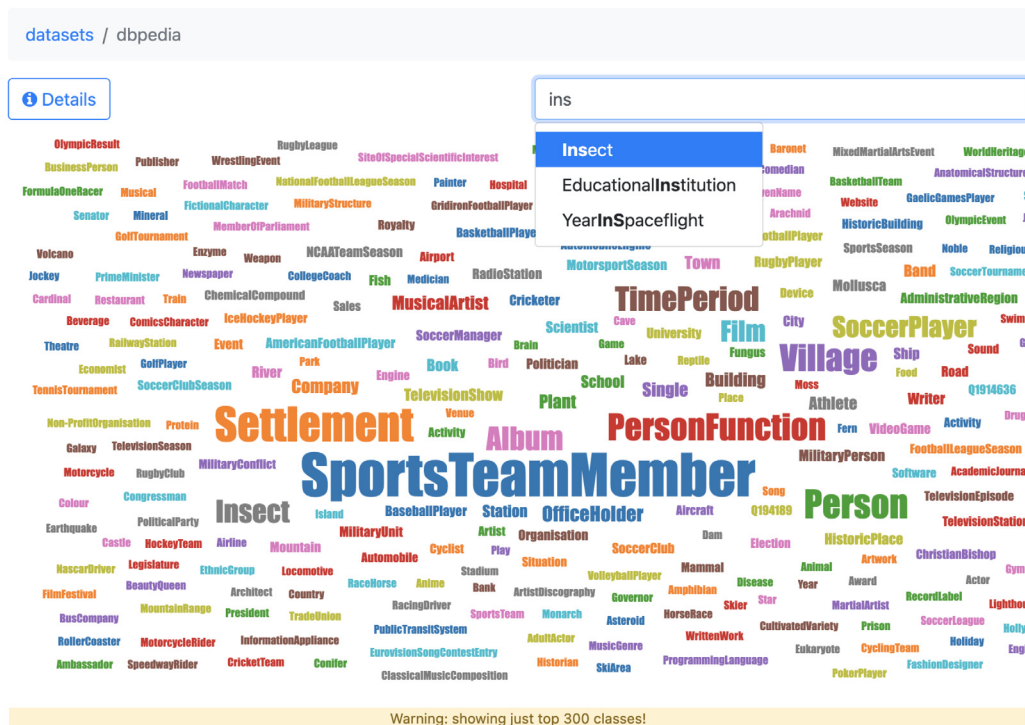


Fig. 1. Overview as a word cloud, online interactive version <https://rhizomer.rhizomik.net/datasets/dbpedia>.

- **Endpoint Graphs Management:** when the endpoint storing the data for a dataset is writable, Rhizomer facilitates its management through its user interface. It is possible to create graphs and load data into them as further detailed in the Usage section.
- **Inference:** Rhizomer provides inference capabilities for datasets with ontologies with rich class hierarchies available through endpoints that do not offer this capability. Based on the data and ontologies to be explored, Rhizomer materialises all inferences regarding subclass hierarchies plus domain and range restrictions. The materialised data is then used during the exploration so, for instance, all instances of a class can be explored together even if that information is not captured explicitly in the original data.

The interactive exploration facilitated by the combination of all these features is useful to discover the structure of an existing dataset, for instance, to reuse it. Moreover, for users generating new datasets with graph data, Rhizomer is also a valuable tool to check if the data structure satisfies the expectations, including outliers, missing values, or issues with the relationships among items. The following subsections detail how Rhizomer supports each of the exploration tasks.

Overview

With an overview, users get the full picture of the dataset. Rhizomer automatically generates a word cloud to provide an overview of the kinds of things in the dataset, as shown in [Fig. 1](#). This is the default overview mechanism because it works even for really big datasets like DBpedia, with more than 100 million statements, as the 300 most common classes are displayed.

For a more informative overview that also includes how the main classes relate among them, there is also the option of a network representation. In this case, the 30 most instantiated classes are shown as nodes together with the most frequent properties connecting them as labelled edges. Fig. 2 shows an example of the network overview.

Listing 1: SPARQL query to retrieve all instantiated classes and their number of instances. It excludes anonymous classes, those without a URI to identify them

```
1 SELECT ?class (COUNT(DISTINCT ?instance) AS ?
2         n)
3 WHERE {
4     ?instance a ?class
5     FILTER ( !isBlank(?class) )
6 } GROUP BY ?class
```

Both overview features are completely data-driven, derived from querying the underlying data with SPARQL queries like the one shown in Listing 1, being SPARQL the standard for graph database querying [13]. This approach facilitates the exploration of schemaless data, like that generated by directly transforming existing data to RDF, or to verify that the explored data conforms to the intended schemas and ontologies.

It is also possible to configure the schemas and ontologies that accompany the data to be explored, in case they are available. In this case, Rhizomer will use the ontologies to retrieve the labels for the classes and properties, so more user-friendly presentations can be built, based on labels instead of fragments of the corresponding URIs. Future plans include leveraging the ontologies, when available, to generate overviews that make use of the hierarchical organisation of classes in the ontology, like Treemaps [14].

Zoom and filter

After a class is selected from the overview, Rhizomer generates a faceted view. It zooms in and allows filtering of resources of the chosen type based on their properties, as shown in Fig. 3. As with the previous step, this view is generated automatically, driven by the underlying data even if it lacks a schema. This feature also allows exploring data that does not fully comply with an existing

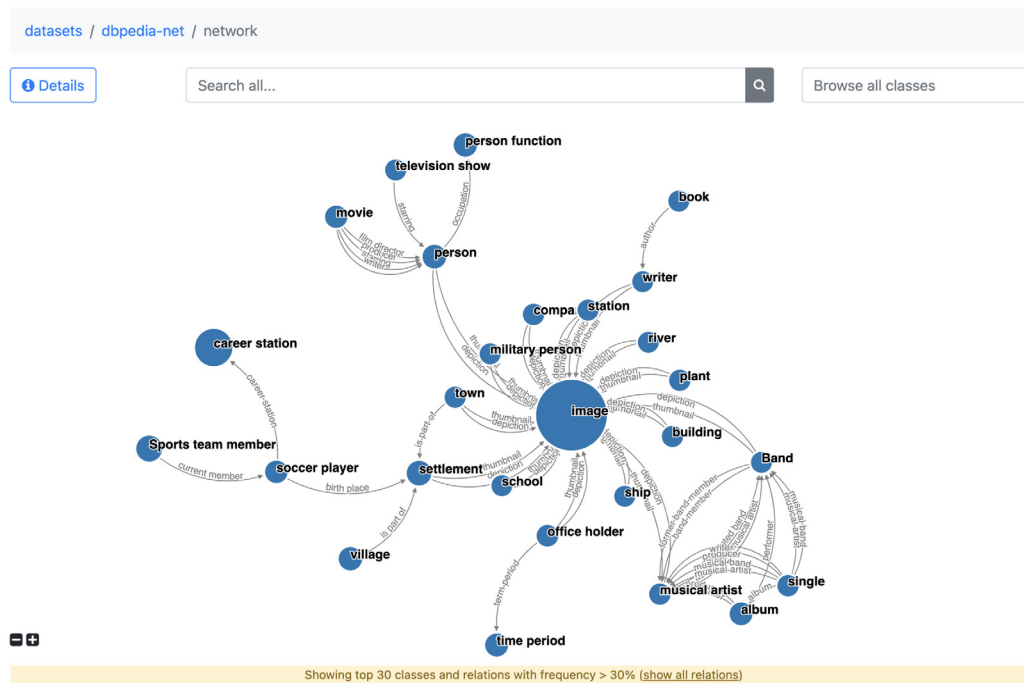


Fig. 2. Overview as a network, online interactive version <https://rhizomer.rhizomik.net/datasets/dbpedia-net/network>.

Listing 2: SPARQL query to retrieve all facets for class *Insect*, which correspond to the properties used to describe its instances. The result includes how many instances use each facet and the number of different values per facet

```
1 SELECT ?property (COUNT(?instance) AS ?uses)
2   (COUNT(DISTINCT ?object) AS ?values)
3 WHERE {
4   ?instance a <http://dbpedia.org/ontology/
5     Insect> ;
6     ?property ?object
7 } GROUP BY ?property
```

schema and highlights these inconsistencies during exploration to help users spot them.

For instance, Listing 2 shows the SPARQL query to retrieve all facets for class *Insect* in DBpedia. Similar queries are also used to implement the other features supporting the Zoom and Filter task. More details are available from the RhizomerAPI GitHub repository.

Like in the case of Overview, Rhizomer's features supporting the Zoom and Filter tasks make use of the ontologies the data is based on if they are available. In this case, for the moment, they are just used to retrieve properties, ranges and values labels.

Details-on-demand

After zooming and filtering, the user arrives at the resources of interest. All properties and values are shown for every selected resource. Users can also browse resources linked directly or through reverse facets, as shown in Fig. 4.

Regarding linked resources, it is also important to note that Rhizomer goes beyond the exploration of a closed dataset. If the retrieved graph data uses external URLs from which additional data can be retrieved, Rhizomer also behaves as what is called

a Linked Data browser [15]. This makes it easy to reuse and integrate existing data.

An example of this is the reuse of the URLs identifying the Game of Thrones books in DBpedia¹ in a custom Game of Thrones dataset. For instance, to capture in which of the books a given character appears. In addition to exploring through Rhizomer all the dataset data, the user can click these URLs for the books and, transparently, all the data available from these URLs will be presented to the user. Just like if it was captured in the custom dataset, but without requiring the dataset creators to copy or write themselves the details about each book. And the benefits of Linked Data do not end there, the exploration process can continue by following links, for instance, the book's authors' details, details about their birthplace, etc.²

Architecture

To provide the previous functionality while keeping a good User Experience, Rhizomer is based on a client/server architecture shown in Fig. 5. The frontend, called RhizomerEye, is based on the Angular framework, which provides a higher level of interactivity on the user side as it is based on the Single Page Application paradigm [16].

On the other hand, the backend, RhizomerAPI, is based on the Spring framework and takes care of the resource-intensive tasks of dealing with the graph databases through their SPARQL endpoints. Its API isolates the frontend from the generation of the required SPARQL queries to extract the classes or facets from the underlying graph data. This includes taking care of the particularities of some graph stores, enriching queries with labels in different languages when they are available or caching the

¹ DBpedia is the Linked Data version of Wikipedia, <https://dbpedia.org>.

² <https://rhizomer.rhizomik.net/datasets/got/dbo:Book>

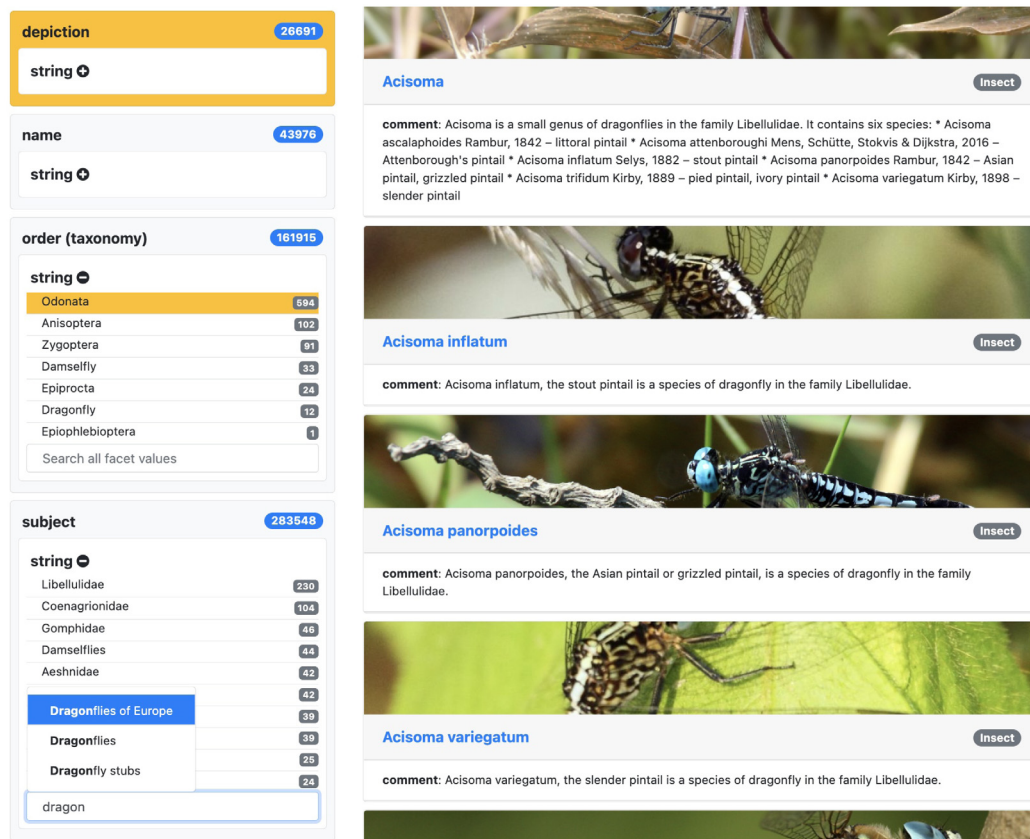


Fig. 3. Faceted view, online interactive version <https://tinyurl.com/ycktrcka>.

results in a relational database to speed up future interactions. The documentation of the Rhizomer API is available online.³

It is also important to note that Rhizomer goes beyond the exploration of datasets available from one or more SPARQL endpoints. It is also possible to fetch graph data from the Web if it is available as Linked Data. As pointed out previously, this makes it easy to reuse and integrate existing data.

Another features of Rhizomer, facilitated by its client/server architecture, are that deployment is not tied to a particular graph database and is a multiuser tool. Each user can interactively configure the datasets they want to explore. This way, anyone can try Rhizomer by requesting a user account for an existing deployment. To make it even easier for users without semantic graph technologies experience, it is also possible to create graphs and load data into them interactively through Rhizomer Web user interface.

Usage

The first step to use Rhizomer, after installing both frontend and backend or if using an existing deployment like <https://rhizomer.rhizomik.net>, is to create a new dataset and configure it. First, in addition to the dataset name and if it can be explored publicly or remains private, the user should choose between “Detailed” or “Optimized” exploration. “Detailed” is recommended because it takes into account different facet ranges and enables

the Network overview. However, for big datasets like DBpedia or to simplify the user experience, it might be better to use “Optimized” exploration, which treats all facets ranges as text literals and used Word Cloud to generate the overview.

The next step in the configuration process is to define the SPARQL endpoint where the data to be explored is located. A form like the one shown on the left side of Fig. 6 is used to define the kind of SPARQL server, the URL of the endpoint and if it is password protected or writable. In the latter case, separate endpoint addresses or credentials are supported for write operations. Additionally, it is possible to enable inferencing for the explored data based on the corresponding ontologies, as previously detailed for the Inference feature in the Software description section.

The final step of the configuration process is to select the data graphs available from the SPARQL endpoint to be explored. It is possible to combine many graphs to explore all of them together. This is very useful to retain data provenance or to integrate different data sources. The form used for managing dataset graphs is shown on the right side of Fig. 6. If the SPARQL endpoint is writable, it also allows the creation of new graphs in the corresponding endpoint and the loading of existing RDF data files into them. Each graph can be selected to be used as data to be explored, as a source for ontological knowledge or both, if the ontology is also to be explored. Being able to define ontologies, which should be loaded into the corresponding graph, enables features like Multilingual Support or Inference.

Once the configuration is completed, the exploration process starts by generating the corresponding Overview. This might take from just seconds for datasets with thousands of statements

³ <https://rhizomer-api.rhizomik.net/swagger-ui.html>

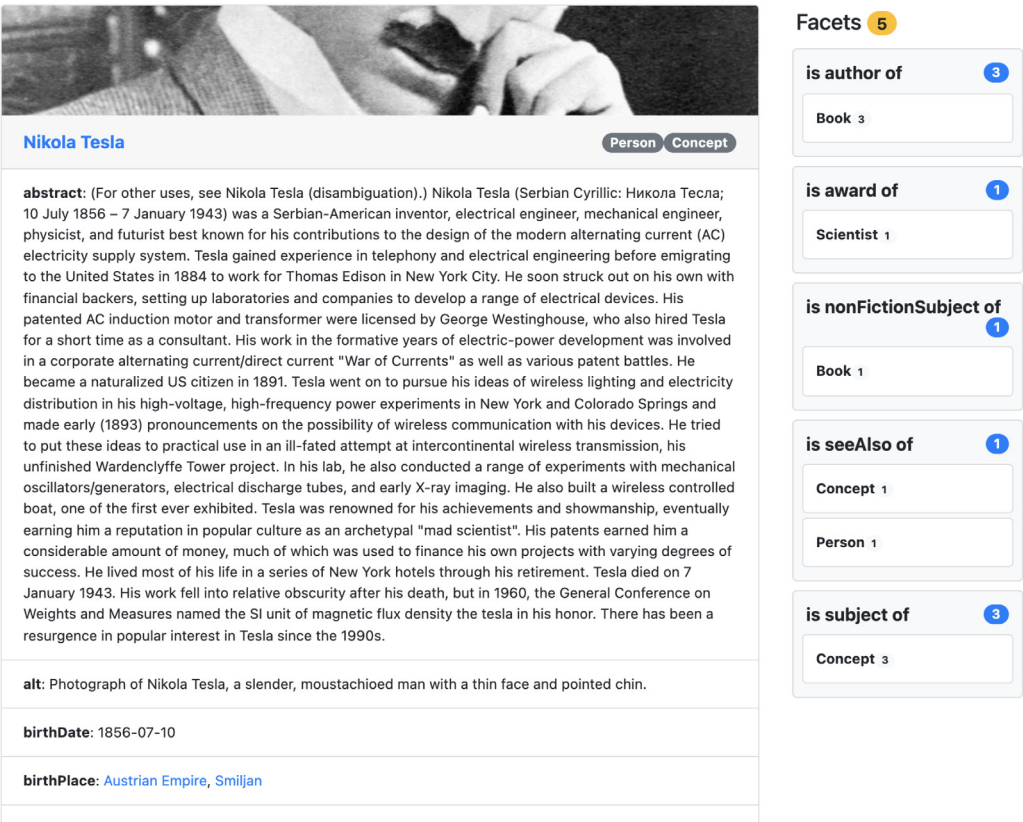


Fig. 4. Details view, online interactive version https://rhizomer.rhizomik.net/datasets/dbpedia/dbpedia:Person/resource?uri=http:%2F%2Fdbpedia.org%2Fresource%2FNikola_Tesla.

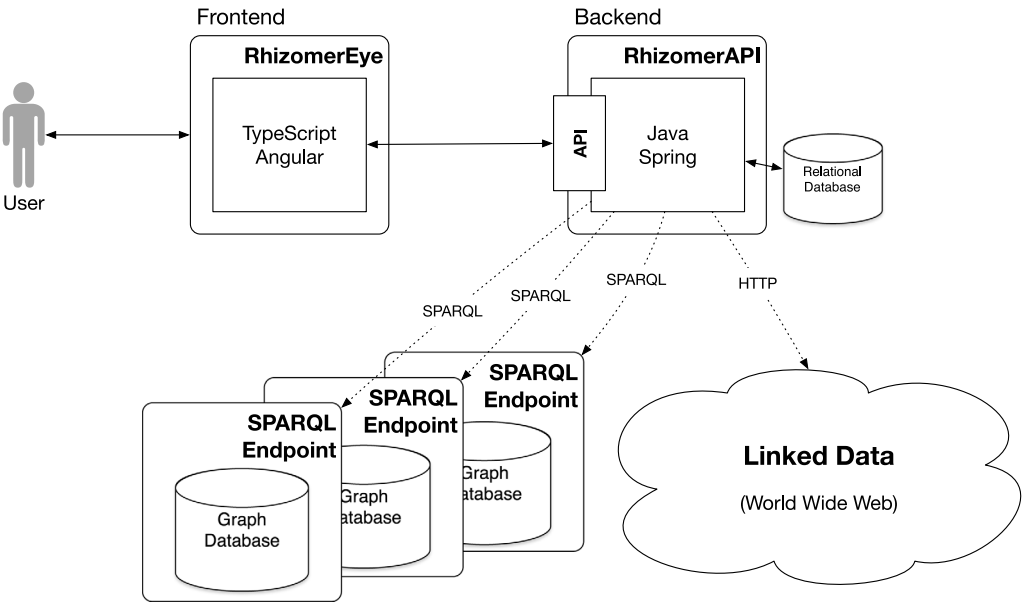


Fig. 5. Rhizomer's architecture.

(triples) to some minutes for bigger datasets, with millions of triples, especially if a “Detailed” exploration is intended. It will largely depend on the dataset size and the SPARQL endpoint performance. In any case, this is done just the first time the dataset is explored after configuration. The data structures required to build the overviews are cached in RhizomerAPI. The same happens for

classes’ faceted views, they are computed the first time they are required and then cached.

Impact

Rhizomer helps researchers discover the structure of existing graph datasets they might be interested in reusing, as long as

Fig. 6. Configuring the SPARQL endpoint for a dataset (left) and the endpoint's graphs to explore as data or ontology.

they are available or converted into the RDF standard. Moreover, Rhizomer also assists researchers in generating data modelled as a graph using RDF. In this case, researchers use Rhizomer to inspect the data they are generating, its shape, potential missing values, existing outliers, etc.

Without Rhizomer, users interested in discovering the structure of an RDF dataset have to rely on SPARQL [13], the standard query language for graph databases. However, this is a cumbersome process that provides a fragmented dataset overview, made by the results of each query used to inspect the dataset. For instance, the list of main classes or the properties used to describe a particular class.

There are other tools that, like Rhizomer, also provide visual interfaces to explore graph data and hide the particularities of SPARQL. However, their focus is on helping explore the data building easier but not trivial query notations, including graphical notations like RDF Explorer [17], or through faceted views combined with a list of all available classes, like RDF Surveyor [18].

To the best of our knowledge, none of them, despite providing detailed or faceted views, combine both with usable overviews of the dataset like a word cloud or a network diagram. Without them, users require some a priori knowledge of the data structure to know where to start. If the entry point is a search form that allows looking for a specific class of entities in the dataset, for instance, *Insect*, the user needs to know what to type, even if there is autocomplete assistance. On the other hand, presenting the list of all available classes can be overwhelming for most users, especially for big datasets and even for experts.

Another important feature of Rhizomer is that it is easier to use than most alternatives. Usually, similar tools are deployed and tied to a specific dataset. End-users are not allowed to configure their datasets. With Rhizomer, they do not even need their deployment. Users can request a user account and interactively configure their dataset through Web forms.

For more advanced users, it is also possible to deploy your copy of Rhizomer. There are prebuilt Docker images for both the frontend and the backend available from the Docker Hub public registry:

- RhizomerEye's Docker image for the frontend, it has more than 1200 pulls (downloads) from Docker Hub.⁴

- RhizomerAPI's Docker image for the backend, more than 1100 pulls from Docker Hub.⁵

Rhizomer is the result of many years of research and development and promising results along this path. In 2013, Rhizomer won the Intelligent Exploration of Semantic Data (IESD) Challenge.⁶ In 2015, Rhizomer won the VIVO Linked Open Data Contest.⁷ with its application to facilitate the exploration of scholarly data based on the VIVO Ontology and available online⁸

In addition to these prizes, we are aware of the following past research projects that used Rhizomer:

- MediaMixer⁹ aim was to facilitate media reuse. Rhizomer provided the means to explore the semantic data generated after media segmentation and annotation.
- InVID¹⁰ targeted the news industry, facilitating social media verification and reuse negotiation for journalistic purposes. The reuse conditions were modelled using semantic technologies and available for system administrators through Rhizomer.
- MediSys monitors the media for potential plant health threats using knowledge captured from domain experts and represented using semantic technologies [19]. Rhizomer facilitated the interaction with the captured knowledge which is available online.¹¹
- The Diabetes Dataset is an ontology that models the diabetes domain from an existing open dataset of around 70,000 diabetic patients [20]. The dataset is published and can be explored using Rhizomer.¹²
- Wikidata Subsetting is a project that deals with the enormous size of the Wikidata dataset (about 1.5 billion statements) by providing mechanisms to extract subsets [21]. Rhizomer is proposed to explore the quality of the generated subsets, for instance for COVID19-related data.¹³

⁵ <https://hub.docker.com/repository/docker/rhizomik/rhizomer-api>

⁶ <http://imash.leeds.ac.uk/event/2013/challenge.html>

⁷ <https://twitter.com/rogargon/status/649162786510086144>

⁸ <https://rhizomer.rhizomik.net/datasets/linkedudl>

⁹ <https://www.mediaboxer.eu>

¹⁰ <https://www.invid-project.eu>

¹¹ <https://rhizomer.rhizomik.net/datasets/PlantHealthThreats>

¹² <https://rhizomer.rhizomik.net/datasets/diabetes>

¹³ <https://rhizomer.rhizomik.net/datasets/covid19>

⁴ <https://hub.docker.com/repository/docker/rhizomik/rhizomer-eye>

Additionally, Rhizomer is also used for teaching as it makes it easier for students to explore and interact with the data that they generate. For instance, in the Web of Data subject, groups of students develop their projects based on semantic data. Semantic technologies facilitate the integration of different data sources and the final results are made available through Rhizomer.

Regarding commercial projects, Rhizomer has been used in collaboration with Amazon Web Services to illustrate the features of their graph database offering, the Neptune database. The results of this collaboration are detailed in an AWS Database Blog post.¹⁴

Before that, Rhizomer was promoted in collaboration with the IT Services and Consulting firm GFT at CODE_n,¹⁵ a global cross-industry innovation platform for digital pioneers, startups, and corporations. Additionally, it has been explored with the media services company NueMeta to help communicate to its clients the value of the semantic data generated for them [22].

The ongoing research projects using Rhizomer are:

- ANGLIRU (Applying kNowledge Graphs to research data interoperability and ReUsability) aims to make research data easier to find, access, integrate, and reuse by using semantic knowledge graph technologies. Once converted into semantic graph form, researchers will use Rhizomer to explore the data.
- UdL Experts is the experts portal used by Universitat de Lleida to promote the expertise of its researchers. After mapping and integrating legacy data into the semantic form, interested parties can use Rhizomer to search for experts and get to know their research topics and outputs.¹⁶

Conclusions and future work

As detailed in the Impact section, Rhizomer has been used in a wide range of scenarios, from research or commercial projects to teaching. It has shown its usefulness when exploring knowledge graphs available as semantic data. Users can get both the overall structure and intricacies of the dataset, even if they do not have prior knowledge about the dataset at hand or the underlying semantic technologies.

Rhizomer's versatility, being capable of exploring data both at the global and detailed level, is attained by performing the three typical data analysis tasks: get an overview, zoom and filter, and details-on-demand. These capabilities make it useful for both data reusers, who want to know about the reuse opportunities of a given dataset, and for semantic knowledge graph creators, who can check if the generated dataset follows their expectations.

Future work focuses on including additional mechanisms for data exploration that make use of the underlying ontologies, like Treemaps [14], or tailored to specific kinds of data, like charts for numeric data, timelines for chronological information or maps for geolocated resources. Additional efforts are planned to improve the performance of Rhizomer when exploring big datasets, especially involving features involving text search. To this end, Rhizomer is being tailored to the mechanisms provided by different SPARQL engines regarding text indexing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the data and code is available from GitHub through the provided links.

Acknowledgements

This work was partially supported by project “ANGRU: Applying kNowledge Graphs to research data ReUsability” with reference PID2020-117912RB-C22 and funded by MCIN/AEI/10.13039/501100011033. Additionally, we would like to thank everyone who has participated in the development of Rhizomer, and its current or previous versions, namely: Oriol Aguilar, Gil Grau, Juan Manuel Gimeno, David Castellà, Josep Maria Brunetti, and Joan Manel Giménez.

Illustrative examples

Explanatory videos showing how Rhizomer can be used to explore different semantic knowledge graphs are available from https://www.youtube.com/playlist?list=PLJ0YJaEOtqlkgZv7Ogl_0kBttWqTfIGDC.

References

- [1] Berners-Lee T, Hendler J. Publishing on the Semantic Web. *Nature* 2001;410(6832):1023–4.
- [2] Angrosch M, Cranefield S, Stanger N. Contextual information retrieval in research articles: Semantic publishing tools for the research community. *Semantic Web* 2014;5(4):261–93.
- [3] Kirchberg M, Leonardi E, Tan YS, Link S, Ko RKL, Lee BS. Formal concept discovery in Semantic Web Data. In: Domenach F, Ignatov DI, Poelmans J, editors. *Formal concept analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012, p. 164–79.
- [4] Pandat A, Gupta N, Bhise M. Load balanced Semantic aware distributed RDF graph. In: 25th international database engineering & applications symposium. New York, NY, USA: Association for Computing Machinery; 2021, p. 127–33. <http://dx.doi.org/10.1145/3472163.3472167>.
- [5] Padiya T, Bhise M, Vasani S, Pandey M. Query execution for RDF data on row and column store. In: Natarajan R, Barua G, Patra MR, editors. *Distributed computing and internet technology*. Cham: Springer International Publishing; 2015, p. 403–8.
- [6] Kirrane S, Sabou M, Fernández JD, Osborne F, Robin C, Buitelaar P, et al. A decade of Semantic Web research through the lenses of a mixed methods approach. *Semantic Web* 2020;11(6):979–1005.
- [7] Charalampidis CC, Keramopoulos EA. Semantic Web user interfaces – A model and a review. *Data Knowl Eng* 2018;115:214–27. <http://dx.doi.org/10.1016/j.datak.2018.04.003>.
- [8] Pesquita C, Ivanova V, Lohmann S, Lambrix P. A framework to conduct and report on empirical user studies in Semantic web contexts. In: Faron Zucker C, Ghidini C, Napoli A, Toussaint Y, editors. *Knowledge engineering and knowledge management*. Cham: Springer International Publishing; 2018, p. 567–83.
- [9] Nečaský M, Stenclák Štěpán. Interactive and iterative visual exploration of knowledge graphs based on shareable and reusable visual configurations. *J Web Semant* 2022;73:100713. <http://dx.doi.org/10.1016/j.websem.2022.100713>.
- [10] Tablan V, Bontcheva K, Roberts I, Cunningham H. Mimir: An open-source Semantic search framework for interactive information seeking and discovery. *J Web Semant* 2015;30:52–68. <http://dx.doi.org/10.1016/j.websem.2014.10.002>, Semantic Search.
- [11] Hitzler P. A review of the Semantic Web field. *Commun ACM* 2021;64(2):76–83.
- [12] Shneiderman B. The eyes have it: A task by data type taxonomy for information visualizations. In: *Proceedings of the IEEE symposium on visual languages*. Boulder, CO, USA: IEEE; 1996, p. 336–43. <http://dx.doi.org/10.1109/VL.1996.545307>.
- [13] DuCharme B. *Learning SPARQL: querying and updating with SPARQL 1.1*. 2nd ed.. Sebastopol, CA: O'Reilly Media; 2013.
- [14] Shneiderman B. Tree visualization with tree-maps: 2-D space-filling approach. *ACM Trans Graph* 1992;11(1):92–9. <http://dx.doi.org/10.1145/102377.115768>.
- [15] Heath T, Bizer C. *Linked data: evolving the web into a global data space*. 1st ed.. Morgan & Claypool Publishers; 2011.

¹⁴ [AWSBlogDatabaseRhizomer](https://awsblogdatabaseRhizomer)

¹⁵ <https://vimeo.com/60635390>

¹⁶ <https://experts.udl.cat>

- [16] Jadhav MA, Sawant BR, Deshmukh A. Single page application using AngularJS. *Int J Comput Sci Inf Technol* 2015;6(3):2876–9.
- [17] Vargas H, Buil-Aranda C, Hogan A, López C. RDF explorer: A visual SPARQL query builder. In: Ghidini C, Hartig O, Maleshkova M, Svátek V, Cruz I, Hogan A, et al., editors. *The Semantic Web. Lecture notes in computer science*, Cham: Springer International Publishing; 2019, p. 647–63. http://dx.doi.org/10.1007/978-3-030-30793-6_37.
- [18] Vega-Gorgojo G, Slaughter L, Von Zernichow BM, Nikolov N, Roman D. Linked data exploration with RDF surveyor. *IEEE Access* 2019;7:172199–213. <http://dx.doi.org/10.1109/ACCESS.2019.2956345>.
- [19] Alomar O, Batlle A, Brunetti J, García R, Gil R, Granollers A, Jiménez S, et al. Development and testing of the media monitoring tool MedISys for early identification and reporting of existing and emerging plant health threats. *EPPPO Bull* 2015;45(2):288–93. <http://dx.doi.org/10.1111/epp.12209>.
- [20] Subirats L, Gil R, García R. Personalization of Ontologies Visualization: Use Case of Diabetes. In: Alor-Hernández G, Sánchez-Cervantes JL, Rodríguez-González A, Valencia-García R, editors. *Current trends in semantic web technologies: theory and practice. Studies in computational intelligence*, Cham, Switzerland: Springer International Publishing; 2019, p. 3–24. http://dx.doi.org/10.1007/978-3-030-06149-4_1.
- [21] Labra-Gayo JE, Hevia AG, Álvarez DF, Ammar A, Brickley D, Gray AJG, et al. Knowledge graphs and Wikidata subsetting. Tech. rep., BioHackrXiv; 2021, <http://dx.doi.org/10.37044/osf.io/wu9et>.
- [22] García R, Sincaglia N. Semantic Web Technologies for User Generated Content and Digital Distribution Copyright Management. In: Polleres A, Garcia A, Benjamins R, editors. *Proceedings of the industry track at the international semantic web conference 2014*, vol. 1383. Riva del Garda, Italy: CEUR Workshop Proceedings; 2014, p. 2, URL <http://ceur-ws.org/Vol-1383/paper14.pdf>.