RDF Explorer: A Visual Query Builder for Semantic Web Knowledge Graphs

Hernán Vargas[1], Carlos Buil-Aranda[1], Aidan Hogan[2], and Claudia López[1]

[1] Universidad Técnica Federico Santa María, Valparaíso, Chile
{hvargas,cbuil,claudia}@inf.utfsm.cl
[2] DCC, Universidad de Chile, Santiago, Chile ahogan@dcc.uchile.cl

Abstract Despite the growing popularity of knowledge graphs for managing diverse data at large scale, users who wish to pose expressive queries against such graphs are often expected to know (i) how to formulate queries in a language such as SPARQL, and (ii) how entities of interest are described in the graph. In this demo we present a system that allows non-expert users to simultaneously navigate and query knowledge graphs by incrementally exploring the dataset.

1 Introduction

In a previous work [7], we presented a visual language whose operators are based on an interactive graph-based exploration. That exploration allows non-expert users to simultaneously navigate and query knowledge graphs effectively. In that work we discussed various desirable properties that a language for querying SPARQL endpoints should have, such as avoiding interactions that lead to empty results. Here we present a demo of the tool that implements such a language and that allows querying SPARQL endpoints based on an exploration activity.

Most popular user interfaces to query SPARQL endpoints simply propose a text box that allows end users to write a query. However, using such interface for querying graphs is challenging. First, users are required to have technical knowledge of the query language and the semantics of its operators. Second, graph data usually represents a variety of data from different domains, meaning that the users may not be easily able to conceptualize the data that they are querying. Despite these limitations, query services for DBpedia and Wikidata are receiving in the order of millions of queries per day [6,4].

Some approaches adopted by interfaces that allow lay users to visualise, search, browse and query knowledge graphs involve keyword search, faceted browsing, graph-based browsing, query building, graph summarization, visualization techniques, and combinations thereof [3]. Many of these systems however, foster usability by removing SPARQL features such as the use of the OPTIONAL operator or cycles within the query. Furthermore, interfaces that allow cycles within graph patterns assume technical expertise of the query language and/or knowledge of how data are modeled.

Here we present an interface that aims to allow lay users to build and execute graph-pattern queries on knowledge graphs; within our proposal, the user navigates a visual representation of a sub-graph, and in so doing, incrementally builds a potentially complex (cyclical) graph pattern. The interface can be accessed at `https://www.rdfexplorer.org` and the code can be found in its corresponding GitHub repository[3].

2 RDF Explorer

In this section, we present the RDF Explorer system, whose goal is to enable lay users to query and explore RDF graphs. We present the main characteristics of the system as well as a description of how the overall system is implemented.

RDF Explorer Interface The interface is composed of six main components displayed in three panes. Figure 1 provides a screenshot of the interface for querying Wikidata, where we can see three components: a search panel (left pane), a visual query editor (center pane), a node detail view (right pane); in the top right corner are buttons to switch the right pane to display one of the three other components: a node editor (allowing to add restrictions to a highlighted node), a SPARQL query editor (showing the current query), and a help panel.

- The *search panel* allows the user to search for an RDF resource (in Wikidata currently, Figure 1 left). This feature uses a Wikidata index [5] to display the results. The user can drag these resources onto the visual query panel to start either exploring the graph or developing the SPARQL query.
- The *visual query panel* allows the user to visually generate a SPARQL query by adding variables as nodes and edges as properties. To be able to draw an edge there has to be either a variable or an RDF resource previously added from the search panel. If there are two RDF resources the interface automatically adds a variable within the origin node and lists in the node detail view the existing properties that relate these two nodes. The visual query panel allows to draw a graph based on the data stored at the endpoint, displaying information about each node in the node detail view.
- The *node detail view* allows the user to visualize the properties of a node, or a result set in the case that the node is a variable, iterating over the possible results (Figure 1, center). The node view also allows to view the SPARQL code of the query that is being developed and presents to the user the possibility of manually adding specific URIs to the node (Figure 1 right).

[3] `https://github.com/hvarg/RDFExplorer`

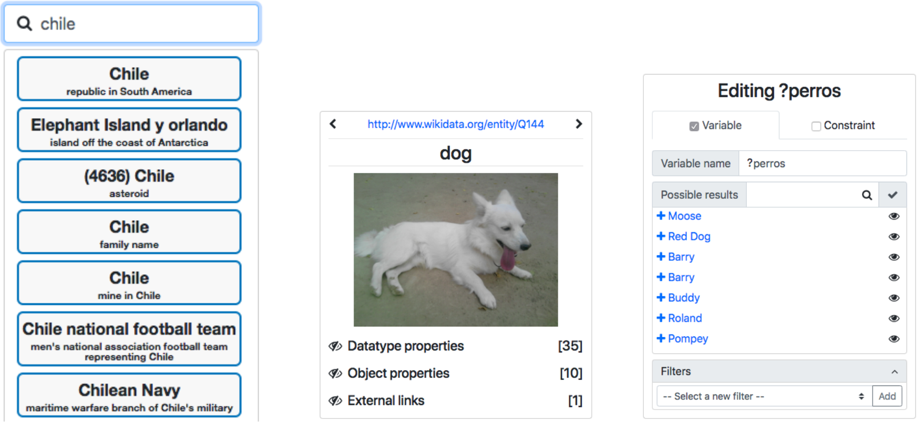

Figure 1: RDFExplorer features: RDF search engine (left), RDF resource descriptor (center) and node view panel (right)

Query creation process The process starts with a blank visual query editor. The user must then start by adding a new node, be it a variable node or a constant node; for selecting x, the user can type a keyword phrase into the search pane on the left, which will generate autosuggestions, where any of the results shown can be dragged into the central query editor pane. The user may then proceed to add a second node by the same means. With two or more nodes available, the user can now click and drag between two nodes to generate an edge with a variable edge-label (shown as a box nested inside the source node); a list of potential IRIs will be suggested for replacing the variable, where only IRIs that generate non-empty results for the underlying query will be offered.

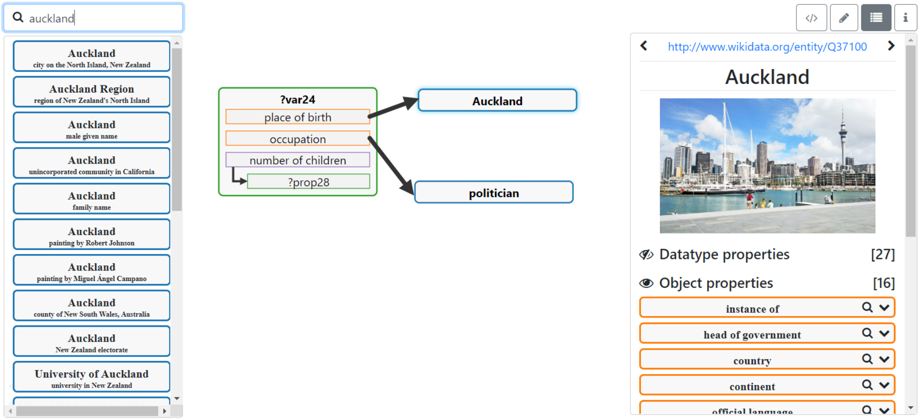

Figure 2: Example visual query finding politicians born in Auckland

Figure 2 illustrates some further features of the interface. Following conventions used in the case of property graphs, we display datatype properties within a given node to avoid clutter; this can be seen for the number of children property in Figure 2. At any point, the user may click on a node to view further details: if the node is variable (see Figure 2), they will be shown a sample of current results for that variable (generated by mapping the current visual query to SPARQL and projecting that variable as described in [7]); if the node is constant (see Figure 2), they will be shown the data available for that node, organized by datatype properties (which take a literal value) and object properties (which take an IRI value). In this way, per the discussion of Bhowmick et al. [1], the user can explore the graph and receive feedback on the results generated thus far, guiding the next steps. Constant nodes can also be converted to variables nodes, enabling the user to start with a specific example and then generalize the graph into a query for similar nodes [2].

3 Conclusions

In this demo we presented the companion user interface to the paper in [7]. This user interface allows lay users to explore and query knowledge graphs, currently configured to access the Wikidata endpoint. The interface does this by allowing users to understand how data is organized and presenting such data in a comprehensible manner through different elements of the interface.

References

1. S. S. Bhowmick, B. Choi, and C. Li. Graph querying meets HCI: State of the art and future directions. In *ACM International Conference on Management of Data*, pages 1731–1736. ACM, 2017.
2. A. Clemmer and S. Davies. Smeagol: a "specific-to-general" semantic web query interface paradigm for novices. In *Database and Expert Systems Applications (DEXA)*, pages 288–302. Springer, 2011.
3. A. Dadzie and M. Rowe. Approaches to visualising linked data: A survey. *Semantic Web 2*, pages 89–124, 2011.
4. S. Malyshev, M. Krötzsch, L. González, J. Gonsior, and A. Bielefeldt. Getting the Most Out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph. In *International Semantic Web Conference (ISWC)*, pages 376–394. Springer, 2018.
5. J. Moreno-Vega and A. Hogan. Grafa: Scalable faceted browsing for RDF graphs. In *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, pages 301–317, 2018.
6. M. Saleem, M. I. Ali, A. Hogan, Q. Mehmood, and A. N. Ngomo. LSQ: The Linked SPARQL Queries Dataset. In *International Semantic Web Conference (ISWC)*, pages 261–269. Springer, 2015.
7. H. Vargas, C. Buil-Aranda, A. Hogan, and C. López. RDF Explorer: A Visual Query Builder for Semantic Web Knowledge Graphs. In *International Semantic Web Conference (ISWC)*. To appear, 2019.