Experience with Visual SPARQL Queries over DBPedia

Kārlis Čerāns, Jūlija Ovčiņņikova, Mikus Grasmanis and Lelde Lāce

Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia

Abstract

We describe a development of tooling for and experience with visual UML-style presentation and visual-based creation of SPARQL queries over *DBPedia*. The tool for visual query creation offers query auto-completion based on the actual *DBPedia* schema; we add to the tool an index-based service to start a query from an individual. We discuss end-user experience in creating visual SPARQL queries over *DBPedia* and evaluate the possibilities of auto-mated visual query generation from their SPARQL form in the context of the public QALD-9 query dataset.

Keywords

SPARQL, DBPedia, Visual queries, Query visualization

1. Introduction

Visual presentation of information artefacts can help their perception. The tools for visual creation of SPARQL queries over RDF data endpoints (cf. e.g., [1-4]) introduce the visual cognitive aspect into the query perception and query composition. Visual method has been successfully used for SPARQL query formulation by do-main experts [1]. *ViziQuer* [4] has shown the possibilities of using a UML-style notation to visually create [5] and visualize [6] complex SPARQL queries, involving e.g., basic graph patterns, aggregation and subqueries, complex data expressions and filters. This paper reports on novel options to use visual queries over *DBPedia* [7,8] - one of the central Linked Data resources of fundamental importance to the entire Linked Data ecosystem. A solution for visual query creation over DBPedia has been out-lined in [9], where the services for query auto-completion, based on class-to-property and property-to-property relations, as well as some client-side enhancements over the generic visual query solution, based on available property ordering and filtering, have been presented. ***The current work reports on expanding the visual DBPedia query environment*** by means for efficient query starting from individual resources, as well as it discusses findings from an ***early user study in formulating the queries*** over *DBPedia* in the context of the expanded visual environment.

The study of creation of visual *DBPedia* queries is accompanied in this work with a ***study of visualization of existing SPARQL queries*** (cf. [6]) over the *DBPedia* data endpoint. We

VOILA! 2022: 7th International Workshop on Visualization and Interaction for Ontologies and Linked Data, October 23, 2022, Virtual Workshop, Hangzhou, China, Co-located with ISWC 2022.

karlis.cerans@lumii.lv (K. Čerāns); julija.ovcinnikova@lumii.lv (J. Ovčiņņikova); mikus.grasmanis@lumii.lv (M. Grasmanis); lelde.lace@lumii.lv (L. Lāce)

CEUR Workshop Proceedings (CEUR-WS.org)

consider the queries from the QALD-9 test dataset[1] used originally in the context of natural language query answering [10] (we look at 150 queries that have textual formulations in English); we provide visual presentations of these queries. For 146 out of 150 queries (over 97%) the visual presentation can be computed automatically, allowing to conclude that there exist contexts (e.g., the QALD-9 queries), where the visual method can be used for the SPARQL query visualization purposes.

The ***gallery of the visually presented queries*** is available in a working visual tool environment, where each visual query can be translated back into SPARQL and executed over the *DBPedia* data endpoint[2]. The user can also use the visual environment to create new queries over *DBPedia*, and to visualize other existing SPARQL queries. The environment can be accessed from the paper's supporting site available at http://viziquer.lumii.lv/examples/dbpedia2022.

The concept of query auto-completion, essential for a satisfactory user experience in the visual query creation, has been implemented over large data sets in various settings, including the Wikidata query service [11], or FAAS [12] solution to support query creation in RDF Explorer [2]. Our solution is the first one that offers the in-stance name lookup for the *DBPedia* data set in the context of a visual query environment.

2. Visual Query Notation Review

A UML-style visual query in *ViziQuer* notation [5,13] consists of nodes describing variables or resources, each node can have a possible class name and attribute specification. One of the nodes is marked as the main query node (orange round rectangle). The edges that connect the nodes correspond to links among the node variables or resources (there can be "same-instance" links, labelled by '==', and "empty" links that do not specify a data connection, labelled by '++', as well). Textual condition/filter fields, along with aggregation and query nesting links are available, as well.

Figure 1 shows six simple visual queries over the *DBPedia* SPARQL endpoint, they are chosen to match the visualizations of SPARQL queries from the QALD-9 dataset, as well as to illustrate different query building constructs appearing in the examples. Just four of the nodes in the six queries have their class names specified (the default *dbo:* namespace classes *Volcano*, *Book*, and *Film*, as well as the *yago:* namespace class *WikicatJapaneseMusicalInstruments*). The instances, matching the query nodes are marked by resources (e.g., *dbr:Constitutional_monarchy*, *dbc:Countries_in_Africa* and *dbr:Taiko*), or by variables (e.g., *uri*, *area*, *x*). Each of the presented queries includes a comment stating the QALD-9 ID and the textual purpose of the query.

Since *DBPedia* typically provides English-based human-readable URIs, the results of the queries are often obtained in the form of URIs matching the variables used in the query; the selection of the variables corresponding to the query nodes are marked by the (*select this*) notation within the query node attribute list. The attribute expressions can be specified for selection, as well; *rdfs:label@en* (choose the label in English) and *dbp:birthName* are some examples.

[1] https://github.com/ag-sc/QALD/blob/master/9/data/qald-9-test-multilingual.json
[2] https://dbpedia.org/sparql/

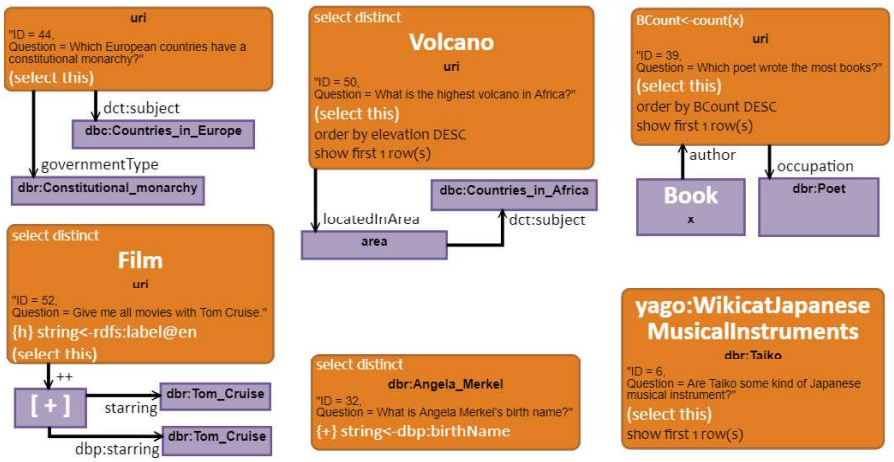

Figure 1: Six QALD-9 queries over *DBPedia* SPARQL endpoint.

An attribute, marked by {+}, is required to have value for the corresponding row to be included in the result set (in *ViziQuer* the attributes are optional by default); the mark h specifies finding the attribute but not including it into the selection list. The query (QALD-9 ID=52) finding all movies with Tom Cruise features a union construct (marked by [+]).

A description of other *ViziQuer* notation features (e.g., the notation for subqueries) can be found in [13] and on the tool website.

3. Visual Query Creation

The visual query environment assists the user in query creation by offering searchable lists of entities (classes, properties and individuals) for starting (seeding) the query, as well as similar lists for extending (growing) a partial query with entities relevant in its context. Since the actual schema of the *DBPedia* endpoint involves over 480 thousand classes and over 50 thousand properties, a relational database support for query completion over *DBPedia* has been introduced in [9] and involves (partial) storing of both the class-to-property and property-to-property relations.

Since a variety of natural SPARQL queries over *DBPedia* would involve some individual, whose properties are to be retrieved or analyzed (this applies also to the considered QALD-9 queries), an efficient service for starting a query from an indi-vidual is essential. We have introduced such a service, based on an index of all *DBPedia* resources and categories (all individuals, except the documents), and have integrated it into the query environment. Starting a query from an individual within the *ViziQuer* tool is performed by typing a part of the individual's URI (full or pre-fixed form) in a search-box to the right of the visual diagram pane.

The data for the index are currently collected from the *DBPedia* data dump, and are stored into the same Postgres database, as the data schema, holding the *DBPedia* class and property information. It should in principle be possible to also invoke an external service as e.g., *DBPedia Lookup* [14] to fill the instance auto-completion list; this is seen as a task for a future work.

The extended auto-completion functionality of the visual query environment over *DBPedia* has allowed to carry through a preliminary user study on the suitability of the visual query environment for creation of queries over *DBPedia*. The study has been performed with technically literate end users (IT master's degree students) with limited previous experience of *ViziQuer* (about 15 queries composed over a simpler data endpoint). The study indicates that the environment can be used by the considered end user group, ***provided the encoding of the information in the data endpoint is clear***.

The students were given textual formulations of nine queries similar in spirit to the QALD-9 queries, with some advanced constructs involved, as shown in Table 1.

1. How many films are there?
2. Find (list) all films starring Tom Cruise.
3. Find the 10 youngest tennis players (list the player resource URI and the birth date).
4. Find all soccer players that are born on or after January 1, 2007. Note: use `"yyyy-MM-dd"^^xsd:date` as the format for date literals (replace yyyy, MM and dd by the year, the month and the date, respectively, e.g., as in `"2007-01-01"^^xsd:date`).
5. How many grandchildren did Thomas Jefferson have?
6. Find the person with the largest grandchildren count (list the person resource URI and the grandchildren count).
7. Find the three top countries with largest volcano counts in the country (list the country resource URI and the volcano count).
8. List all persons that each have 100 or more films starring them (list the person resource URI and the respective film count).
9. Find all politicians, born on the same date as Tom Cruise (list the politician's resource URI and the birth date).

Table 1
Queries for the user study.

Out of the 10 study participants all 10 were successful on queries #1-#4 and 9 were successful on the "more advanced" queries #9 and #10[3]. The queries #5 and #6, related to grandchildren count, had 6 successful completions (using a chain of *dbo:child* relations), while the rest attempted to use *dbp:grandchildren* property that does not provide the necessary data. The query #7 however, had only 3 of 10 successful submissions based on the *dbo:locatedInArea* property (the others have used the *dbo:country property*, which provided much less information). Further information on the user study is available on the paper's support page.

The study results show that the size of the *DBPedia* data set schema, as presented within the visual query environment, does not impede the query creation by the con-sidered user group, as the queries have generally been successfully composed. The present query composition failures in queries #5, #6 and #7 are clearly related to the lack of the knowledge of the study participants on the ways, how the information has been en-coded in the *DBPedia* resource. Clearly, the visual method for the query creation alone would not be sufficient to overcome this difficulty. Some possible approaches to the solution might involve encouraging the participants to try different candidate approaches for the information extraction and then choose the one that apparently gives the most relevant results. Some help on the existing variations of the information encoding within the data set could perhaps be offered by the query environment, however, this task would already go into the realm of natural language question answering (cf. e.g., [10]) and is beyond the scope of this short paper.

[3]The criterion for success of a query formulation work is producing a query that returns the results that are equivalent to the expected ones.

4. SPARQL Query Visualization

The visual presentation of a textual SPARQL query provides an additional perspective on a query presentation by invoking the visual cognitive aspect in query perception, so potentially easing the task of understanding a query (note that the visual query form can be used in addition to the textual form, not necessarily replacing it). The visual SPARQL query presentation using ***UML-style*** notation (cf. [6]) can be said to have benefits of (i) presenting the ***classification and attribute selection triples*** in a compact notation (that involves just a class or property name within a query node), and (ii) splitting the query over ***multiple visual elements*** to reduce the local complexity of any visually separated query element.

We used the automated visualization of SPARQL queries [6] (a slightly optimized and fine-tuned version of it) to create a visual query library from the QALD-9 test queries over *DBPedia* to see, how the SPARQL query visualization methods work in practice, as well as to demonstrate the capabilities of the visual notation on a set of externally available queries. It can be argued that the visual presentation of these particular queries do provide insights into different ways, how a factual information may happen to be encoded into *DBPedia* (via class names, via linked resources (*dbr:* namespace entities) or categories (*dbc:* namespace), or even via fragments within class names (within *yago:* namespace); the use of the default *dbo:* namespace and the *dbp:* namespace for properties is to be considered, as well).

There are 4 SPARQL queries in the QALD-9 query set that are of the ASK format. Since the *ViziQuer* notation currently supports only SELECT queries, we visualize an ASK query into a SELECT query that returns a single line in the case of a result *true*, and an empty result set in the case of the result *false*.

There are 150 queries in the considered QALD-9 query subset, all of them are presented in the visual notation within the visual QALD-9 query library (Figure 1 depicts 6 of these visualizations). In the case of 146 queries (involving the 4 ASK queries) the visual presentation can be produced automatically (followed by a manual tuning of the query placement)[4] [5]. 4 queries have been drawn or corrected manually: 2 queries involving HAVING construct (expected to be modeled in the current visual notation via a filter in an enclosing query) and 2 queries involving complex UNION constructions (query optimization and implementation fine-tuning would resolve these issues).

The visual *DBPedia* query library is available as a project within the *ViziQuer* environment, accessible from the paper's support page.

5. Conclusions

The presented work shows the possibility to use UML-style visual presentation both in visual creation of SPARQL queries over DBPedia, and in visualization of existing SPARQL queries. The provided technical solutions of the data schema and instance information pre-computation and

[4] The non-standard SPARQL expressions as *SELECT xsd:date(?var)* found in 10 QALD-9 queries need to be re-written into *SELECT (xsd:date(?var) AS ?v)* before the visualization.

[5] A query is successfully visualized, if it produces equivalent results to the original query, when executed. The structural equivalence is considered as a primary visualization success criterion for queries that do not produce any results, when run over the *DBPedia* endpoint.

storage allow to ensure a smooth auto-completion experience both for the initial query seeding and its further context-based growing. An interesting future work would be to fine-tune the auto-completion process, by providing the class-to-individual connections (to suggest efficiently the names of individuals in a class context) based on *DBPedia Lookup* [14] or some FAAS-style component [12].

The presented user study could be seen as preliminary confirming that the size and complexity of the *DBPedia* query environment are not limiting the use of the schema and data elements in formulating a query. It has shown clearly, however, also that the visual method alone does not necessarily indicate which elements of the data schema should be used to find answers to the textually formulated queries. An interesting future work would be to develop and test an eco-system of semi-automated extracting the information from a SPARQL endpoint, like *DBPedia*, based on different knowledge encoding patterns, known for the data set.

The visual presentation of all QALD-9 example SPARQL queries demonstrates the capability of the automated query visualization method to handle queries from an externally existing data set. This result complements an earlier work on authors on *Wikidata* example query visualization [15] to show the maturity level of the visual query notation (as well as to indicate the directions of its further improvements).

The SPARQL query visualization provides a visual structure for any considered SPARQL query, thus adding the visual perception benefits to the query understanding. Considering the set of visualizations for the entire QALD-9 dataset allows also for visual conceptual evaluation of patterns of information encoding into *DBPedia*, up to possible discussion in what ways *DBPedia* can be used as an *ad hoc* information source in different knowledge areas.

Acknowledgements This work has been partially supported by a Latvian Science Council Grant lzp-2021/1-0389 “Visual Queries in Distributed Knowledge Graphs”.

References

1. Soylu, A., Giese, M., Jimenez-Ruiz, E., Vega-Gorgojo, G.., Horrocks, I.: *Experiencing OptiqueVQS: A Multi-paradigm and Ontology-based Visual Query System for End Users.* Universal Access in the Information Society, March 2016, Volume 15, Issue 1, pp 129–152.
2. Haag, F., Lohmann, S., Siek, S., and Ertl, T. *QueryVOWL: Visual Composition of SPARQL Queries.* In: The Semantic Web: ESWC 2015 Satellite Events. Springer LNCS, Vol.9341, pp. 62-66. (2015).
3. Vargas, H., Buil-Aranda, C., Hogan, A. and Lopez, C. *RDF Explorer: A Visual SPARQL Query Builder*, Proc. of ISWC 2019, Springer LNCS, Vol. 11778, pp. 647-663. (2019).
4. Čerāns, K., Šostaks, A., Bojārs, U., Ovčiņņikova, J., Lāce, L., Grasmanis, M. Romāne, A., Sproģis, A., Bārzdiņš, J. *ViziQuer: A Web-Based Tool for Visual Diagrammatic Queries Over RDF Data.* In: ESWC 2018 Satellite Events. Springer LNCS, vol 11155, pp. 158-163. (2018).
5. Čerāns, K., Šostaks, A., Bojārs, U., Bārzdiņš, J., Ovčiņņikova, J., Lāce, L., Grasmanis, M., Sproģis, A.: *ViziQuer: A Visual Notation for RDF Data Analysis Queries.* In Proc. of MTSR’2018, Springer CCIS, Vol.846, pp.50-62 (2019)

6. Čerāns, K., Ovčiņņikova, J., Grasmanis. M., Lāce, L. and Romāne, A. *Visual Presentation of SPARQL Queries in ViziQuer.* In Proc. of VOILA! 2021, CEUR Workshop Pro-ceedings Vol. 3023, pp. 29-40 (2021). http://ceur-ws.org/Vol-3023/paper12.pdf
7. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S. *DBpedia - A crystallization point for the Web of Data.* Web Semantics: Science, Services and Agents on the World Wide Web. 7 (3): pp. 154–165. (2009).
8. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hell-mann, S., Morsey, M., Van Kleef, P., Auer, S. and Bizer, C. *DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia.* Semantic Web 6(2), 167–195. (2015)
9. Čerāns, K., Lāce, L., Grasmanis, M. and Ovčiņņikova, J. *A UML-style Visual Query Environment over DBPedia.* In Proc. of MTSR2021, Springer CCIS, Vol. 1537 (2022).
10. Usbeck, R., Gusmita, R.H., Ngonga Ngomo, A.-C., Saleem, M. *9th challenge on question answering over linked data (QALD-9).* CEUR Workshop Proceedings Vol. 2241, pp. 58-64. (2018). http://ceur-ws.org/Vol-2241/paper-06.pdf
11. Wikidata Query Service. https://query.wikidata.org/
12. de la Parra, G., and Hogan, A. *Fast Approximate Autocompletion for SPARQL Query Builders.* CEUR Workshop Proceedings Vol. 3023, pp.41-55. (2021). http://ceur-ws.org/Vol-3023/paper10.pdf
13. Čerāns, K., Bārzdiņš, J., Šostaks, A., Ovčiņņikova, J., Lāce, L., Grasmanis, M. Sproģis, A.: *Extended UML Class Diagram Constructs for Visual SPARQL Queries in ViziQuer/web.* In Voila!2017, CEUR Workshop Proceedings, Vol.1947, pp.87-98. (2017) http://ceur-ws.org/Vol-1947/paper08.pdf
14. DBPedia Lookup, https://lookup.dbpedia.org/
15. Čerāns, K., Ovčiņņikova, J., Grasmanis, M. and Lāce, L. *Towards UML-style Visual Queries over Wikidata.* In ESWC 2022: The Semantic Web: ESWC 2022 Satellite Events, Springer LNCS volume 13384, pp.11-15. (2022).