SHACL Shape Extraction for Evolving Knowledge Graphs using QSE

Diploma thesis - Proposal

Eva Pürmayr, 11807199

Jan 2024

Supervision: Univ.Prof. Dr.-Ing. Dipl.-Inf. Katja Hose

1 Problem Statement

Storing complex and interconnected data is quite a challenge for data engineers. While relational systems are commonly used for structured data, tasks which involve a lot of relationships and lack a fixed schema may find a solution in graph databases. These serve as a foundation for constructing knowledge graphs, which have a broad range of applications, in industry and in academia and can also be used as a basis for machine learning. As in many applications, data quality is vital and is the basis for good results later on. As a validation language SHACL can be used for RDF [1]. Previously, an approach called QSE (quality shape extraction) which automatically extracts SHACL shapes from large datasets, has been released [2]. The user can define two important parameters, which omit less useful shapes: support and confidence. Based on these generated node and property shapes, existing data can be validated. An extension to this program is called Shactor, a web-based tool which visualizes the extraction process and provides useful statistics [3].
Since knowledge graphs are not static, there exist different versions of a graph, maybe with minimal changes. QSE and Shactor are specialized on shapes extraction, but not comparing them between different versions of a graph. This work aims at shapes extraction, specifically with evolving knowledge graphs. After this thesis is finished, users should be able to conveniently and efficiently compare SHACL shapes across various versions of a knowledge graph.

2 Goals and Expected Outcome

The main goal of this thesis is to develop a web-based tool which allows shapes extraction for different versions of a graph. Some features such as shape extraction will be based on Shactor. However, users can define different versions of a graph. Users can choose between the exact or approximate QSE approach, define support and confidence and can specify classes and during shape extraction. They can create multiple shape extractions with different parameters. Later, when a new version of the graph is uploaded, users can compare the shapes with previously generated shapes. The web-app shows the users, which shapes stayed the same, which were added and which were deleted. Also, there will be information provided why shapes were included or removed.
The second part of the thesis deals with the QSE-algorithm and how it can be adapted for multiple versions of a graph. An approach is to use the changeset

between two graph versions as additional information during shape extraction so that the changes between the shapes can be generated without generating the shapes for each version first.
There might also be other algorithms, which extract SHACL shapes from RDF graphs. Another goal of this thesis is to implement an algorithm which explains why certain shapes have been added, removed or changed by using partial SPARQL queries instead of using support and confidence from the QSE algorithm. This can then be applied for any extracted shapes and the corresponding dataset, independently from QSE. The web application will also provide graphical interfaces to use these algorithms.
For testing the algorithms different datasets will be used, where graph versions have different degrees of change. Datasets for this use case could be the ones used in BEAR [4].
The success criteria for this project involves the creation of a website enabling users to easily compare SHACL shapes and correctly developing the two mentioned algorithms. Additionally, success is defined by opinion on usefulness of experts.

3 Research Questions

After defining the goals it becomes evident that there is a need for data engineers to ensure the quality of knowledge graphs across multiple versions. Since there is always a manual part in quality assurance, there is the need for a user-friendly web-based tool to evaluate the differences between extracted shapes of knowledge graphs. Considering the frequent changes in knowledge graphs, it would be a lot of work to run and compare extracted SHACL-shapes for every version. The state-of- the art solution would be to download the extracted shapes from Shactor and compare them with text-comparison tools. Given two versions of a graph (V1 and V2) and the corresponding SHACL shapes generated by QSE (S1 and S2):

RQ1.1: What is an appropriate way to compare S1 and S2 in a user-friendly way?

RQ1.2: What is an appropriate way to explain why certain shapes from S1 have been added, removed or changed by using information from the QSE algorithm?

RQ2.1: Given the changeset between V1 and V2, what is an appropriate way to adapt the QSE algorithm so that it calculates S2 by using V1 and the changeset?

RQ2.2 What is an appropriate way to adapt the QSE algorithm so that it calculates S2 and simultaneously creates the changeset between V1 and V2?

RQ2.3: There might graphs, which are only available via an SPARQL-endpoint. What is an appropriate way to explain why certain shapes of S1 have been added, removed or changed in V2 by using partial SPAQRL queries?

Appropriate for RQ1.1 and RQ1.2 means user-friendly, useful and correct. It can be measured during the evaluation of semi-structured interviews with experts. Appropriate for RQ2.1, RQ2.2 and RQ2.3 means correct and faster in comparison to the method described in the evaluation section.

4 Research Methods

Design science Research is selected as a scientific method. In the relevance cycle as well as in the rigor cycle a **systematic literature review** will be used so that business needs can be derived from the literature and knowledge from the rigor cycle can influence the thesis. Ultimately, this process contributes to the knowledge base and aligns with evolving business needs by the rigor and relevance cycle's conclusion.
However, given the time-consuming nature of a SLR, only a partial review will be done in the beginning of the thesis. During the planning phase, a review protocol will be established and only one online library will be defined, where strict selection criteria will be applied, to limit the number of papers. In the conducting phase, qualitative data extraction will occur, aiming to provide a broad overview of the knowledge base and business needs rather than addressing a specific research question. Reporting will be done in a simplified way, there will not be a dissemination mechanism and the final report will be published in the master thesis.
The review protocol will contain the following items:

- Objectives: Establish a broad understanding of the current knowledge base in the area of evolving knowledge graphs, automatic extraction of data quality constraints from knowledge graphs, comparing of SHACL shapes and differences between knowledge graphs. Identify current research gaps in these areas.

- Search Keywords: “evolving knowledge graphs”, “data quality in knowledge graphs”, “constraint extraction from knowledge graphs”, “comparing shacl shapes”, “differences between knowledge graphs versions”
- Study selection criteria: published after 2000, written in English or German, accessible with TU-Vienna Account for free
- Library: Google Scholar
- Study Exclusion Criteria: Has nothing to do with research questions or objectives
- Procedure: First ten results of will each search keyword will be evaluated in the conduct phase

During the design cycle, different methods will be used to answer the research questions. **RQ1.1 and RQ1.2** will utilize the **prototyping** method within the **construct phase**. The objective is to develop a web-based tool. Initial stages will involve low-fidelity prototypes to determine the most suitable design which satisfies the needs of the end user's requirements. Iteratively, the Minimum Viable Product (MVP) will be developed in close cooperation with the supervisor, incorporating feedback from the evaluation phase.
Contrastingly, **RQ2.1, RQ2.2 and RQ 2.3** will use **algorithmic design** in the construct phase of the design cycle. The approach entails maintaining algorithm simplicity, reusability, and utilizing libraries. Iterative development through small experiments will refine the algorithms. As realistic input, there will be graph snapshots from DBpedia and other data sources. During development, synthesized data and smaller versions of these graphs will be used.

5 Evaluation

For the **evaluation** part in the design cycle for RQ1.1 and RQ1.2 **semi-structured expert interviews** with students, who have already participated in Introduction To Semantic Systems will be conducted. The important part here is to measure the usability of the web-based tool in comparison to finding differences across the shapes without the tool. The interviews will contain a demonstration of the tool's functionality. Open-ended questions will be used to get comprehensive insights about usability. Since this method is time-consuming, 3-5 experts will be interviewed. In the planning phase, the

interview guide will be created with prioritized questions, keeping the length of the interview as short as possible. The format of the interviews, online or offline, will be based the preferences of the experts. It is planned to record the interviews.
The analysis of interview content will employ the inductive approach of qualitative content analysis. Depending on the content of the interviews, categories will be created. Some of the feedback can then be implemented in the web-tool.
For the evaluation phase in the design cycle for RQ2.1, RQ2.2 and RQ 2.3, **technical experiments** will be conducted, aligning with common practices in DSR projects for instantiations. Each research question aims to develop an algorithm. Planning of the experiments will be done carefully, a big part here is the preparation of test data. The experiments will be run on a virtual machine, since the data size will be huge. After it has been ensured, that the algorithm works correctly, speed will become the measurable metric.

6 State of the Art

Presently, the QSE algorithm can be run using the command line, allowing the definition of multiple parameters but lacking user-friendliness. The extension, Shactor, addresses this issue by offering a user interface through a web-based tool. While this program allows users to extract shapes for a single version of a graph, it does not support the extraction of shapes from multiple versions. The current solution for users to find similarities and differences between different versions of a graph would be to use Shactor for both versions and download the shapes. Comparison could be done via a text-comparison tool manually.
Regarding evolving knowledge graphs and defining versions of a graph in general, there has already been a lot of research done [5] [6] [7] [8] [9]. GraphDB already offers a feature for data versioning and history [10].
However, in the context of this research project, adopting this form of versioning is currently not possible because not all graphs use the same form of versioning. Standardization across all graphs would be essential, otherwise the application would only be applicable to specific graphs utilizing a particular versioning technique.
There has been some general research on how data quality can be ensured in knowledge graphs [11] [12] [13] [14]. And there exist algorithms, which automatically extract a schema from data but have several shortcomings [15]

[16] [17] [18] [19] [20] [21] [22] [23] as already described in the QSE paper. Briefly explained, the QSE (Qality Shape Extraction) process involves two iterations through all entities. During the initial phase (entity extraction), all instances based on their type declarations are counted. Subsequently, in the second run (entity constraints extraction), the algorithm gathers metadata for property shapes. Following this, support and confidence metrics are computed and finally, in the last step, the algorithm generates SHACL shapes.

7 Relevance to the Curriculum

This subject builds upon the Introduction to Semantic Systems course, which is part of the Information Systems Engineering Core curriculum [24]. Foundational understanding in semantic systems, specifically in RDF-graphs and knowledge graphs is required. It is also interesting to understand the application of knowledge graphs and therefore grasp the significance of their quality. The course Knowledge Graphs provides insights into these aspects [25]. In the context of RDF, familiarity with SPARQL is good to have, yet understanding of SHACL is more important. Given that QSE is written in Java, proficiency in Spring and Web Development becomes essential. Expertise in Spring is covered in Distributed Systems Technologies [26]. Moreover, basic comprehension of usability and software project design (which is taught in Advanced Software Engineering) is valuable [27]. Addressing RQ2.1, RQ2.2 and RQ2.3, knowledge in algorithmic design is essential.

References

[1] *Shapes Constraint Language (SHACL)*, en, Jul. 2017. [Online]. Available: https://www.w3.org/TR/shacl/ (visited on 11/28/2023).

[2] K. Rabbani, M. Lissandrini, and K. Hose, "Extraction of Validating Shapes from Very Large Knowledge Graphs," en, *Proceedings of the VLDB Endowment*, vol. 16, no. 5, pp. 1023–1032, Jan. 2023, ISSN: 2150-8097. DOI: 10.14778/3579075.3579078. [Online]. Available: https://dl.acm.org/doi/10.14778/3579075.3579078 (visited on 11/13/2023).

[3] ——, "SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes," in *Companion of the 2023 International Conference on Management of Data*, ser. SIGMOD '23, event-place:

Seattle, WA, USA, New York, NY, USA: Association for Computing Machinery, 2023, pp. 151–154, ISBN: 978-1-4503-9507-6. DOI: 10.1145/3555041.3589723. [Online]. Available: https://doi.org/10.1145/3555041.3589723.

[4] *BEAR — BEnchmark of RDF ARchives*. [Online]. Available: https://aic.ai.wu.ac.at/qadlod/bear.html (visited on 11/14/2023).

[5] M. Frommhold, R. N. Piris, and N. Arndt, "Towards Versioning of Arbitrary RDF Data," en,

[6] A. Lohfink and D. McPhee, "Resource-Level Versioning in Administrative Geography RDF Data," en,

[7] M. Tasnim, D. Collarana, D. Graux, F. Orlandi, and M.-E. Vidal, "Summarizing Entity Temporal Evolution in Knowledge Graphs," en,

[8] R. Pernischova, D. Dell'Aglio, M. Horridge, M. Baumgartner, and A. Bernstein, "Toward Predicting Impact of Changes in Evolving Knowledge Graphs," Oct. 2019.

[9] M. U. Nuha, "Data Versioning for Graph Databases," en, 2019. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3A3cdbd161-3e6b-463a-957d-00ec0942917a (visited on 11/10/2023).

[10] *Data History and Versioning — GraphDB 10.4 documentation*. [Online]. Available: https://graphdb.ontotext.com/documentation/10.4/data-history-and-versioning.html?highlight=version (visited on 10/20/2023).

[11] *Tsaneva and Sabou - Hybrid Human-Machine Evaluation of Knowledge Graphs.pdf*. [Online]. Available: https://drive.google.com/file/u/0/d/1jzH1KhSn6UxxhayoZV5rCXQF1nsCI5a6/view?pli=1&usp=embed_facebook (visited on 11/10/2023).

[12] A. Schmickl, *5 steps to detect inconsistencies in evolving knowledge graphs*, en, Feb. 2021. [Online]. Available: https://alenaschmickl.medium.com/5-steps-to-find-inconsistencies-in-evolving-knowledge-graphs-6f3f88c0ab7b (visited on 11/10/2023).

[13] J. H. Brenas and A. Shaban-Nejad, "Proving the Correctness of Knowledge Graph Update: A Scenario From Surveillance of Adverse Childhood Experiences," *Frontiers in Big Data*, vol. 4, p. 660 101, May 2021, ISSN: 2624-909X. DOI: 10.3389/fdata.2021.660101. [Online]. Available:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8126660/ (visited on 11/10/2023).

[14] K. Kellou-Menouer, N. Kardoulakis, G. Troullinou, Z. Kedad, D. Plexousakis, and H. Kondylakis, "A survey on semantic schema discovery," *The VLDB Journal*, vol. 31, no. 4, pp. 675–710, Jul. 2022, ISSN: 0949-877X. DOI: 10.1007/s00778-021-00717-x. [Online]. Available: https://doi.org/10.1007/s00778-021-00717-x.

[15] N. Mihindukulasooriya, M. R. A. Rashid, G. Rizzo, R. García-Castro, O. Corcho, and M. Torchiano, "RDF shape induction using knowledge base profiling," en, in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau France: ACM, Apr. 2018, pp. 1952–1959, ISBN: 978-1-4503-5191-1. DOI: 10.1145/3167132.3167341. [Online]. Available: https://dl.acm.org/doi/10.1145/3167132.3167341 (visited on 11/29/2023).

[16] D. Fernandez-Álvarez, J. E. Labra-Gayo, and D. Gayo-Avello, "Automatic extraction of shapes using sheXer," *Knowledge-Based Systems*, vol. 238, p. 107 975, 2022, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.107975. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121010972.

[17] B. Spahiu, A. Maurino, and M. Palmonari, "Towards Improving the Quality of Knowledge Graphs with Data-driven Ontology Patterns and SHACL," in *WOP@ISWC*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52193280.

[18] I. Boneva, J. Dusart, D. Fernández Alvarez, and J. E. L. Gayo, *Shape Designer for ShEx and SHACL Constraints*, Published: ISWC 2019 - 18th International Semantic Web Conference, Oct. 2019. [Online]. Available: https://hal.science/hal-02268667 (visited on 11/29/2023).

[19] A. Keely, *Shaclgen: Shacl graph generator*. [Online]. Available: https://github.com/uwlib-cams/shaclgen (visited on 11/29/2023).

[20] *TopQuadrant — Enterprise Models for Data Governance*, en. [Online]. Available: https://www.topquadrant.com/ (visited on 11/29/2023).

[21] H. J. Pandit, D. O'Sullivan, and D. Lewis, "Using Ontology Design Patterns To Define SHACL Shapes," en,

[22] A. Cimmino, A. Fernández-Izquierdo, and R. García-Castro, "Astrea: Automatic Generation of SHACL Shapes from Ontologies," en, in *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, *et al.*, Eds., vol. 12123, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 497–513, ISBN: 978-3-030-49460-5 978-3-030-49461-2. DOI: 10.1007/978-3-030-49461-2_29. [Online]. Available: http://link.springer.com/10.1007/978-3-030-49461-2_29 (visited on 11/29/2023).

[23] P. G. Omran, K. Taylor, S. R. Mendez, and A. Haller, "Towards SHACL Learning from Knowledge Graphs," en,

[24] *188.399 Introduction to Semantic Systems — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/educationDetails.xhtml?dswid=9706&dsrid=233&semester=2023W&courseNr=188399 (visited on 11/28/2023).

[25] *192.116 Knowledge Graphs — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=9706&dsrid=155&courseNr=192116&semester=2023S (visited on 11/28/2023).

[26] *184.260 Distributed Systems Technologies — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=5007&dsrid=808&courseNr=184260&semester=2023S (visited on 11/28/2023).

[27] *180.456 Advanced Software Engineering — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=9706&dsrid=456&courseNr=180456&semester=2022W (visited on 11/28/2023).