Enhancing Knowledge Graphs through Data Validation

Contact persons: Katja Hose (katja.hose@tuwien.ac.at), Kashif Rabbani (kashifrabbani@cs.aau.dk), ?Matteo Lissandrini (matteo@cs.aau.dk)

Knowledge graphs[1] have found extensive applications in both industry and academia as well as in modern Machine Learning approaches[2]. In such application, it is essential that the underlying knowledge graph provides reliable facts. Hence, data quality plays a decisive role and can be improved by validating existing graph data to identify erroneous or missing information. To address this need, the World Wide Web Consortium (W3C) introduced the Shapes Constraint Language (SHACL)[3] to facilitate the validation of knowledge graphs modeled in RDF[4]. One of the first approaches (QSE)[5] able to automatically extract SHACL shapes from very large knowledge graphs uses the data mining principles of support and confidence to obtain a set of meaningful shapes.

Theme 1: Web-Based Knowledge Graph Extraction Interface

There is a broad range of users (experts and non-experts) working with knowledge graphs who are interested in checking, maintaining, and increasing their quality. To help facilitate this process a user-friendly Web application is needed that allows users to extract shapes.

The following extensions can be made to our existing QSE[5] algorithm:

- Extend QSE's scalability and adaptability by transforming it into an anytime algorithm, enabling on-the-fly shape extraction in such a way that users can interrupt the extraction at any time and QSE will still output useful shapes.
- Support for mining additional types of SHACL constraints (e.g., recursive constraints)

The following extensions can be made to our existing SHACTOR tool[6]

- Grouping related shapes to help users understand non-obvious similarities.
- Support of an alternative shape language: SheX[7]

---

[1] Aidan Hogan et al.: Knowledge Graphs. Morgan & Claypool Publishers, 2021, https://kgbook.org/
[2] Katja Hose: Knowledge Engineering in the Era of Artificial Intelligence. ADBIS 2023, https://link.springer.com/chapter/10.1007/978-3-031-42914-9_1
[3] Knublauch, Holger; Kontokostas, Dimitris, eds. (2017-07-20): Shapes Constraint Language (SHACL). W3C. RDF Data Shapes Working Group, https://www.w3.org/TR/shacl/
[4] https://www.w3.org/RDF/
[5] Rabbani, Kashif; Lissandrini, Matteo; and Hose, Katja. Extraction of Validating Shapes from very large Knowledge Graphs. VLDB 2023, https://relweb.cs.aau.dk/qse
[6] Rabbani, Kashif; Lissandrini, Matteo; and Hose, Katja. SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes. SIGMOD 2023, https://relweb.cs.aau.dk/qse/shactor
[7] SheX: https://shex.io/

- Support of additional knowledge graph formats, e.g., Turtle[8] (currently support of N-Triples(NT)[9]-formatted RDF files and GraphDB Triplestore via SPARQL)
- Visualization of the extracted shapes along with the corresponding data constraints
- Interactive identification of low-quality (erroneous/spurious) data, enabling users to comprehend the data quality within the knowledge graph.
- Using the tool to analyze several real-world knowledge graphs, e.g., WikiData

Theme 2: Efficient Parallel SHACL Validation

In addition to extracting shapes, users are naturally also interested in efficient SHAPES validation. Some approaches, such as Trav-SHACL[10] and MagicShapes[11], have proposed been proposed. However, these solutions are not scalable and thus cannot handle validating very large knowledge graphs on commodity machines, especially when accessing the knowledge graph via a SPARQL endpoint. Furthermore, these solutions are not evaluated using real-world datasets, such as DBpedia[12] and WikiData[13]. In this theme, we would therefore first like to evaluate existing approaches on large-scale real-world knowledge graphs and based on the obtained insights develop an efficient SHACL validator that can execute multiple validation queries in parallel and validate the graph progressively, which is particularly challenging for very large knowledge graphs. Since data processing frameworks, such as Apache Spark[14] or Apache Flink[15], have not been utilized to parallelize and partition graph validation, it would be very interesting to explore and apply the strengths of these enignes in this context for graph validation.

[8] Turtle https://www.w3.org/TR/rdf12-turtle/
[9] N-Triples https://www.w3.org/TR/n-triples/
[10] Mónica Figuera, Philipp D. Rohde, and Maria-Esther Vidal. 2021. Trav-SHACL: Efficiently Validating Networks of SHACL Constraints. In Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA 12 Pages. https://doi.org/10.1145/3442381.3449877
[11] Ahmetaj, Shqiponja, et al. "Magic shapes for SHACL validation." Proceedings of the VLDB Endowment. Vol. 15. No. 10. Association for Computing Machinery (ACM), 2022.
[12] https://www.dbpedia.org/
[13] https://www.wikidata.org/
[14] https://spark.apache.org/
[15] https://flink.apache.org/

Theme 3: Creating a Standardized Benchmark for SHACL Extraction and Validation for Large-Scale Knowledge Graphs

While there are already several tools available, both for SHACL extraction and validation, the community is lacking well-defined benchmarks and metrics to objectively compare their performance. Hence, this theme aims to develop a standardized benchmark to objectively evaluate and compare SHACL (Shapes Constraint Language) extraction and validation approaches on large-scale knowledge graphs. The project involves generating diverse synthetic knowledge graphs (with diverse characteristics), defining representative SHACL validation rules (covering SHACL core constraints and complex constraints), building an automated benchmarking framework, and establishing quantitative performance metrics such as precision, recall, F1-score, runtime, and memory consumption.

Theme 4: SHACL Shape Extraction for Evolving Knowledge Graphs

Most approaches that work with knowledge graphs assume the knowledge graph to be static, i.e., never changing. In reality though we see that many knowledge graphs are subject to updates and therefore to knowledge evolution. However, current shapes extraction approaches have to be run for each version of a knowledge graph independently and it is then up to the user to find out if identical or slightly different shapes have been extracted and which ones apply to all versions. The goal of this theme therefore is to help the user in this process and automatically find similarities and differences, and if time allows also to investigate how computations for a previous version of the graph can be reused for new versions.

Theme 5: SHACL Shape Validation for Evolving Knowledge Graphs

Whereas Theme 4 focuses on extracting shapes over evolving knowledge graphs, this theme focuses on the validation process. Current approaches are designed to validate a single static graph against a set of shapes and it is up to the user to perform validation over evolving graphs by manually comparing validation reports against different version. Furthermore, existing shapes validators are not able to handle validation over evolving graphs in a progressive manner; the idea of progressive shapes validation is to validate RDF data against SHACL shapes incrementally. Overall, the goal of this theme is to develop a visual tool that enables a user to efficiently get on overview of which shapes are violated in which versions of the graph and where exactly the validation failed, and if time allows also to investigate incremental shapes validation in this context.