Problem Statement: ♣ What is the problem that does not exist anymore once the thesis is (successfully) finished? Try to describe it in one or two sentences. ♣ Does the proposal specify any measurable characteristics of the current situation that need to be solved? Are metrics (KPIs) for these characteristics defined? Are there even values for these characteristics given? If so, outline them.

Goals and Expected Outcome: ♣ What is the proposed solution (to overcome the problem)? Try to describe it in one or two sentences. ♣ Are the characteristics of the solution well defined? Are metrics (KPIs) for these characteristics specified? If so. Outline them. ♣ When is the solution considered a success?

Research Questions: ♣ Is it possible to provide scientific evidence for all answers to the research questions? ♣ Are there any research questions that are not precise (Yes/No questions, “How to develop sth”, etc.)? ♣ If the goal is to develop something appropriate/suitable/etc., does it become clear what appropriate/suitable/etc. means?

Research Methods: ♣ Are the envisioned research method(s) appropriate to reach the expected research results? ♣ Are the methodological steps to be followed clear? If so, outline them. ♣ Is each methodological step based on a research method known from the literature? If so, mention the corresponding paper from the literature describing the underlying research method for each methodological step. ♣ [Only for a thesis that follows Design Science: Design Science requires by definition the application of rigorous research methods. Is it clear which research methods are used within the Design Science framework?]

Evaluation: ♣ Which research method is used for the evaluation? ♣ Are the metrics to be used in the evaluation clear? Which are they? ♣ Is the evaluation appropriate? Does the evaluation provide scientific evidence by coming up with results based on KPIs that demonstrate that the solution solves the identified problem? Provide arguments. [Note, an approach could also fail - so after the thesis, the instantiation of the proposed evaluation may show that the problem is still unsolved.]

State of the Art: ♣ Does the elaboration on state of the art show that the student knows the relevant literature in the field? Provide arguments. ♣ Is there any literature missing? Relevance to the

Curriculum: ♣ Do you consider the thesis relevant for the given curriculum? Provide arguments. ♣ Is the relevance argued by a somewhat arbitrary list of courses (without arguing why each course is relevant), or does the section elaborate on the knowledge necessary to conduct the thesis and, then, show in which courses this know-how is taught?

Overall review ♣ Do you consider the proposal concise? ♣ Is the proposal of excessive length and, thus, should be shortened? First, note a proposal should be limited to about five pages (9000 characters including whitespaces) without the literature list. ♣ Is the style of writing appropriate? Is there a clear line of arguments? ♣ What are the strengths of the proposal? ♣ What are the limitations of the proposal? ♣ What should be revised?

Kürzen, drüberlesen, rechtschreibprüfung, unterlagen durchgehen, ob noch was fehlt

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.

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 \cite{noauthor\_shapes\_2017}. Previously, an approach called QSE (quality shape extraction) which automatically extracts SHACL shapes from large datasets, has been released \cite{rabbani\_extraction\_2023}. 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 \cite{rabbani\_shactor\_2023}. \newline

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. As there is currently no tool available for comparing shapes, the characteristics of usability and speed remain unmeasurable. Using manual methods is the only option, but this is tedious and lacks practicality. After this thesis is finished, users should be able to conveniently and efficiently compare SHACL shapes across various versions of a knowledge graph.

\section{Goals and Expected Outcome}

The main goal of this thesis is to develop a usable, 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, can choose between the exact or approximate QSE approach, define support and confidence and can specify classes 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, which shapes stayed the same, which were added and which were deleted. Also, there will be information provided why shapes were included or removed.\newline

The second part of the thesis deals with adaptions of QSE-algorithm. Using the terms V1 and V2 for two versions of a graph and the corresponding SHACL shapes generated by QSE (S1 and S2), an approach is to use the changeset between V1 and V2 during shape extraction so that S2 can be generated without using V2. \newline

An improvement in this direction is to create S2 and simultaneously create the changeset between V1 and V2.

The main goal of this thesis is to develop a usable, web-based tool which allows shapes extraction and comparison for different versions of a graph. Some features such as shape extraction will be based on Shactor. ForHowever, users can define different versions of a graph, can choose between the exact or approximate QSE approach, define support and confidence and can specify classes 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, which shapes stayed the same, which were added and which were deleted. Also, there will be information provided why shapes were included or removed.\newline

The second part of the thesis deals with adaptions of QSE-algorithm. Using the terms V1 and V2 for two versions of a graph and the corresponding SHACL shapes generated by QSE (S1 and S2), an approach is to use the changeset between V1 and V2 during shape extraction so that S2 can be generated without using V2. \newline

An improvement in this direction is to create S2 and simultaneously create the changeset between V1 and V2.

\newline

There might be graphs which are only available via a SPARQL-endpoint. Another goal of this thesis is to implement an algorithm which explains why certain shapes have been added, removed or changed from S1 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. The most important characteristics of these algorithms is the execution time compared to the baseline, which is defined in the evaluation section.

\newline 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 \cite{noauthor\_bear\_nodate}. \newline

The success criteria for this project involves the creation of a website enabling users to easily compare SHACL shapes and correctly developing the three mentioned algorithms.

Design science Research is selected as a scientific method [5]. 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 [6]. Ultimately, this  
process contributes to the knowledge base and aligns with evolving business  
needs by the rigor and relevance cycle’s conclusion [7].  
However, given the time-consuming nature of a SLR, only a partial review  
will be done in the beginning of the thesis. Only one online library (Google  
Scholar) will be used. The review protocol will contain the following items:  
objectives (e.g. to establish a broad understanding of the current knowledge  
base in the area of evolving knowledge graphs), search keywords (e.g. “evolving  
knowledge graphs”), study selection criteria (e.g. published after 2000), study  
exclusion criteria and procedure.  
During the design cycle, RQ1.1 and RQ1.2 will be answered using the  
prototyping method within the construct phase [8]. 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 [9]. 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 de-  
velopment, synthesized data and smaller versions of these graphs will be  
used