# SHACL Shape Extraction for Evolving Knowledge Graphs with QSE

## Introduction

Knowledge graphs 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, the data quality is vital and is the basis for good results later on. As a validation language SHACL can be used for RDF. Previously, a program called QSE (quality shape extraction) has been released, which extracts SHACL-shapes automatically from large datasets. The user can define two important parameters, which omit seldomly used 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.

Since knowledge graphs are not static, there exist different versions of a graph, maybe with minimal changes. QSE and Shactor are specialised on the shapes extraction, but not comparing them between different versions of a graph. This work aims at shapes extraction, specifically with evolving knowledge graphs.

## Goals

The main goal of this thesis is to develop a web-based tool which allows shapes extraction for different versions of a graph. This web-app will be heavily based on Shactor, since it already implements useful features, such as graph upload. Users can define different versions of a graph. For the first version, the user has the same options as in Shactor. He/She can choose between and exact or approximate QSE approach, define support and confidence and can specify classes and extract shapes. The user can create multiple shape extractions with different parameters. Later, when he/she uploads a new version of the graph, he/she can compare the shapes with previously generated shapes. The web-app shows the user, which shapes stayed the same, which are new and which were deleted. Also, there will be information why shapes were added or deleted.

The second goal of this thesis is to adapt the QSE-algorithm, so that it is faster for another version of the graph. An approach is to calculate the difference between two versions of a graph and create shapes for the later version of the graph based on this difference. Another idea is to execute QSE for the first version and then use the shapes and interim results from the first run in the second run.   
The performance of these versions of QSE will be compared to find the best solution. This comparison will be made with the same parameters and will be based on different datasets, where the versions have different degrees of change.

## Current state of research

## Research Questions

RQ1. How can extracted shapes from different versions of a graph be compared in a user-friendly way?

RQ2. Can the QSE-algorithm be more efficient if the difference of two graphs are calculated first instead of running the algorithm twice for both versions?

RQ3. Which interim results or fully extracted shapes from previous versions can be reused to make the execution faster?

## Methods

## Literature

Notizen und Fragen:

* Andere formen von evolving knowledge graphs eh weglassen?
* Kann ich adaption von qse algorithmus nur für exakt machen? Würde mir einiges ersparen
* Web based tool nur existing datasets und upload graph, In shactor geht file upload nicht, kann ich sparql endpoint weglassen?
* In shactor ist approx. nicht implementiert, das würde ich schon machen, mit fixen parametern
* Alles relational lokal speichern
* Mobile version?
* Comparisions nicht speichern

9.11

* Shapes bearbeiten können? -> nein
* Vaadin kann kein Rich Text -> geht mit custom html

10.11

* Node-Shape Auswahl weg, sonst kommt es zu Inkonsistenzen
  + Vll in Shape extractions noch anzeigen
  + Warning anzeigen, wenn ungleiche paramter verglichen werden
* Übersicht, so wie in Shactor?
  + Weglassen?
* Comparision von NodeShapes weg, dient eh nur zur Übersicht
* Alogrithmus Fragen:
  + Keine unit tests? Kann ich mich auf ergebnisse verlassen?
  + Integration in Web-Tool?
  + Big-Data? Oder reichen kleine Datensets? Hab keinen großen Rechner