Identification of SHACL Constraints from a Knowledge Graph

Erhard Eibl, Siyabend Sakik, Niels Schneider, Oya Beyan, and Nils Lukas

RWTH Aachen University, Germany

1 Introduction

RDF based Knowledge Graphs can make use of SHACL constraints to validate groups of entities against schemas. This is not just helpful for general data validation, but also for enforcing a common data schema when inserting new data. We describe an algorithm that automatically extracts most probable SHACL constraints from Knowledge Graphs. The goal is to enhance data ingestion in collaborative databases, specifically for the medical domain where data veracity is imperative. Our approach consists of three steps, namely to categorize entities, embed them into a vector space and finally, apply a data mining approach to generate most probable schemas. The hereby presented algorithm makes use of statistical analysis to find constraints and is validated on the DrugBank RDF dataset.

2 Conceptual Approach

The main challenge is to mine embeddings for the categorization of the data. We achieve this by grouping all entities by predicate-object pairs. For p predicates and o objects we create $p \cdot o$ vectors of length p and feed them to our algorithm. These categories are found using dynamically generated SPARQL-queries which can be reused later for SHACL target definition. Compare the sentences "Jonagold is a type of apple", "McIntosh is a type of apple" and "Cavendish is a type of Banana". In terms of triples, this might translate to *(Jonagold, instanceOf, Apple)*, *(McIntosh, instanceOf, Apple)*, *(Cavendish, instanceOf, Banana)* respectively. With our approach, we would generate the categories "InstanceOfApple" and "InstanceOfBanana". Since the property is involved in our categories, our method goes beyond the simple class hierarchies. Consider the sentences "Panadol contains Acetaminophen." and "Sedapap contains Acetaminophen", obviously Panadol and Sedapap are similar in some regards, but a class "ThingsContainingAcetaminophen" is not to be expected in a Knowledge Graph. In order to keep the amount of information in a Knowledge Graph manageable in a Big Data environment, it is desirable to make it publicly accessible like in Wikidata or automate knowledge acquisition wherever possible. In both cases, false or even malicious data might be added to the Knowledge Graph which threatens its consistency. This problem makes it necessary to find a method like ours to detect and remove inconsistencies.

3 Realization

We implemented our concept in Java using the Apache Jena RDF graph database. We used the RDF DrugBank dataset that is publicly available and consists of about 780.000 triples. The taxonomy of the data is relatively shallow with only six top level classes. With our approach, we identify 168 distinct categories containing at least 50 entities. Categories with fewer entities are ignored, assuming they do not contain enough information. In our work we use SHACL-SPARQL[2], which is a SHACL extension allowing us to define the shape's target by using SPARQL-queries. The combination of SHACL and SPARQL proved useful because it solves the problem of SHACL Core being unable to define arbitrary sets of entities as shape targets. For each of the categories we measure the minimal and maximal occurrence of a relationship and generate SHACL constraints based on this information. Moreover, we compute a confidence score for each relationship based on its relative occurrence. The observed numerical values are added to the shape files as upper and lower bounds. Even with a confidence threshold of at least 90%, our algorithm suggests constraints on average for 20 distinct properties for each category. Our current approach executed in less than one minute and generated 168 SHACL files on an Intel Core i7-4790K. We expect to decrease the runtime significantly with further optimizations. An example for our result is the shape for ”drugs” in the DrugBank dataset. Our approach gives us the ”minCount” of the ”creationDate” property of ”drugs” entities as 1 and the ”maxCount” also as 1. For the ”drugType” property of ”drugs” entities the generated shapes file notes ”minCount” as 1 but ”maxCount” as 4. Using this shape file and a data graph including a ”drugs” entity with more than one ”creationDate” property for a SHACL validation would return a violation and not accept the data graph. This error could then be corrected.

4 Conclusion

Our algorithm is able to mine basic SHACL constraints from Knowledge Graphs for our automatically defined categories, like minimal and maximal occurrences of values. Building upon this knowledge, our algorithm can detect outliers and suggest a modification to make them conformant to the graph. Furthermore, the embedding into a vector space allows the usage of more sophisticated machine learning algorithms that detects more abstract patterns between entities of the same categories and is more resilient to outliers. We are currently unable to correctly treat blank nodes. For instance in a scenario where recipes use blank nodes for ingredient lists, our implementation would compare all ingredient lists containing raisins but make no connection to the dish itself. In the future, we plan to not only analyze the amount of occurring predicates but also their targets using categories as well as datatype analysis. This would make it possible to restrict the office of president of the United States to entities of type person or to have categories for cities with roughly one million inhabitants. Furthermore we plan on implementing a functionality to manage and improve consistency of a Knowledge Graph even during runtime.

5 Acknowledgment

This work is conducted as part of the Knowledge Graphs Praktikum offered by RWTH Aachen University Informatik 5 department in collaboration with OSTHUS. We thank OSTHUS for defining research goals and requirements from the industry perspective, and providing student travel grants.

References

1. Shapes Constraint Language (SHACL) W3C Recommendation 20 July 2017, https://www.w3.org/TR/shacl/
2. SHACL Advanced Features W3C Working Group Note 08 June 2017, https://www.w3.org/TR/shacl-af/
3. The DrugBank database https://www.drugbank.ca/