Discovery of Ontologies from Implicit User Knowledge

David Haller[1][0000−0001−5287−7187] and Richard Lenz[1][0000−0003−1551−4824]

Chair of Computer Science 6 (Data Management)
University of Erlangen
david.haller@fau.de
richard.lenz@fau.de
https://www.cs6.tf.fau.eu

Abstract. The purpose of the Semantic Web is to enable worldwide access to humanity's knowledge in a machine-processable way. A major obstacle to this has been that knowledge is often either represented in an incoherent way, or not externalized at all and only present in people's minds. Populating a knowledge graph and manually building an ontology by a domain expert is tedious work, requiring great initial effort until the result can be used. As a consequence, knowledge will often never be made available to the Semantic Web. The aim of this project is to develop a new approach for building ontologies from implicit user knowledge that is already present, but hidden in various artifacts like SQL query logs or application usage patterns.

Keywords: Semantic Web · Knowledge Graph · Schema Inference · Query-Driven · Data Integration

1 Introduction

In the last decades, the World Wide Web indisputably changed human society and economy. Computers, paradoxically, although essentially operating the Web, cannot make use of it on their own. Knowledge on the Web is represented mostly in a way suitable for humans, as pages containing plain text and graphics. Although web pages can be structured hierarchically and linked with each other, their inherent semantics are only accessible by a human being perceiving the content. Querying the Web is usually restricted to simple keyword-based search engines or web services with proprietary APIs. Apart from these technical obstacles, the Web also does not define coherent sets of terms that shall be used to describe concepts and entities of a particular domain, leaving that tasks to human interpretation.

The Semantic Web [1] offers a framework for machines to make use of this knowledge. Instead of storing linked HTML documents, the Semantic Web links *facts* with each other. This is done using the Resource Description Format (RDF), which operates on a graph-based data model. The graph can then be queried using SPARQL, the default RDF query language, which has a similar expressivity as SQL has for relational databases.

For being able to actually interpret RDF data, an ontology must be defined. This can be done either in RDF Schema or in the Web Ontology Language (OWL). In a nutshell, an ontology is a set of axioms which constrain what statements can or cannot be true and allows to deduce new statements from existing statements. Creating these ontologies manually is tedious work and therefore a blocker for Semantic Web adoption.

Knowledge already exists somewhere, either in people's minds or in various kinds of artifacts: semi-structured file formats like CSV or JSON, plain text in natural language, applications source code, log files, or SQL queries. Transferring all this knowledge by hand into a graph is time-consuming and expensive, wherefore this can be applied only for limited use cases. Developing an at least partly automated method to perform this task could drastically lower the costs for deploying Semantic Web techniques.

2 Past Research

This PhD research project will extend the scope of the previous master thesis project Pharos, which results have been published in a followup research paper [6]. The focus of Pharos was to improve the understanding of heterogeneous data sources within a data lake by analyzing SQL query logs accessing these sources and by extracting knowledge fragments from those queries in order to gain insights about the underlying schema. This may seem unintuitive at first glance, as SQL is usually associated with relational databases, where schemata are already known, but SQL has evolved into a general query language for heterogeneous data sources. When a data scientist encounters an unknown data source, he needs a great deal of cognitive effort to understand its semantics prior to writing the queries that use the data sources for analytics. Therefore, each query implicitly contains hidden knowledge and assumptions about the data and can be seen as a partial schema definition.

For example, joining two tables over an attribute indicates that the data analyst probably identified a foreign-key-relationship, otherwise he would not have made that join. Renaming columns with speaking names or explicit type casts give hints about their meaning.

```sql
select sum(p.salary), dep.id, dep.name
from person p join department dep
on p.dep_id = dep.id
where dep.location='DE' or dep.location ='FR'
group by dep.id, dep.name
order by dep.name;
```

Listing 1.1. Example query with partial schema information

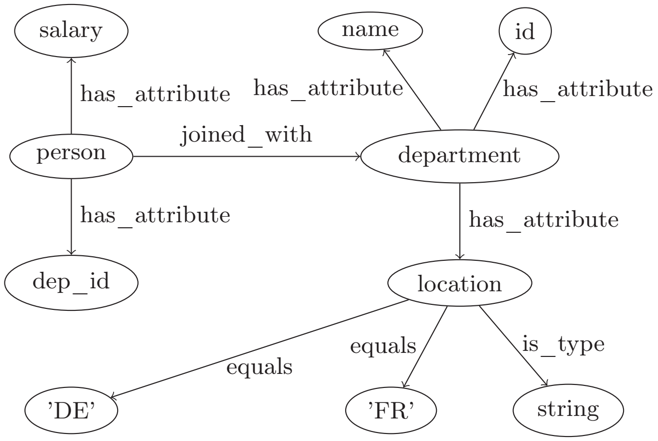

Fig. 1. Partial schema information stored in a knowledge graph

We had decided to build a knowledge graph from SQL query logs that could be used to help understanding the mental model behind data sources. A prototype was written that demonstrates the feasibility of the approach. It was implemented in form of a JDBC proxy driver that can capture and analyze all SQL queries a Java application sends to a SQL query engine like Apache Drill, allowing a minimal-invasive deployment of the prototype into existing workflows, as it is compatible with any software using JDBC drivers. The prototype was evaluated using a test database with a known schema and a set of test queries, based on exercises of our introductory database lecture.

3 Research Objectives

The resulting knowledge graph describes how data sources have been used by analysts, but does not describe the semantics of the data sources themselves, like value constraints or foreign key relationships. Human interpretation of the results is required to gain insights about the used data sources. Therefore, the next level will be to perform automatic reasoning on top of the knowledge graph. This requires to generate an (incomplete) ontology for describing the semantics of data sources. As knowledge derived from SQL queries may be contradictory - when the query log contains queries that are not conforming to the underlying schema - an approximate approach is needed to deal with this ambiguity.

Queries do not reflect the semantics of a data source, but the mental model a data scientist has made of it. Analyzing these mental models can already give valuable insights. For example, if someone uses a "grade" attribute and compares it with values that are not present in the dataset, the explanation could be that the user originated from a country with a different school grading system. There are multiple concepts out there about what a "grade" should be, a query-driven approach could provide more transparency about these concepts as an intermediate step.

When analyzing SQL query logs, there will always be queries that are based on wrong assumptions about the schema, especially if the origin of the query log is from an interactive session where queries with undesired results may be rewritten. With multiple query logs from different sessions and users, finding the similarities in their behavior and their mental model could lead to the intended semantics of the used data sources.

A self-learning system shall be developed that makes suggestions to data scientists about suitable sources or queries they may find helpful for their task. Based on their given feedback and performed queries, the system shall incrementally approximate the true semantics behind the data sources.

Thus far, only SQL query logs were considered as a source for query-driven schema inference. But there are other types of queries to consider, like query strings from search engines, application usage patterns extracted from graphical analysis tools or even source code from programs accessing a data source. Other query languages like XQuery or languages from various NoSQL database systems could be included. The approach does not depend on a specific language.

4 Related Work

Many approaches for schema inference are *data-driven*, using data profiling methods to reconstruct the underlying schema of a given dataset. A significant example is the Metanome project [9], which provides an extensible framework offering various algorithms, for example to discover functional dependencies [7]. Datatype-based schema inference for JSON datasets is demonstrated in [2] and [3]. In [8] it is shown how to identify the domains the values of a column come from. The Datamaran project [5] aims to discover structure in text files like applications logs and transforming them into normalized relational tables. The ESKAPE platform [11] allows users to assign instances to semantic models [10]. A general overview of dataset search and integration techniques is given in [4].

5 Evaluation Approach

The existing prototype will be extended to use Semantic Web reasoning techniques to deduce the meaning of a data source by the knowledge extracted from query fragments. A framework of rules should be defined to achieve this, possibly with the Rule Interchange Format (RIF). This prototype shall then be tested on real world query logs, so the resulting knowledge graph can be compared with the actual schema the data sources are based on. A supplementary user study will show if the software is able to enhance the workflow of data scientists to understand heterogeneous data sources.

References

1. A Semantic Web Primer. Cooperative Information Systems, MIT Press, Cambridge, Mass, 3rd ed edn. (2012)
2. Baazizi, M.A., Ben Lahmar, H., Colazzo, D., Ghelli, G., Sartiani, C.: Schema inference for massive JSON datasets. In: Proceedings of the 20th International Conference on Extending Database Technology. pp. 222–233. Venice, Italy (2017)
3. Baazizi, M.A., Colazzo, D., Ghelli, G., Sartiani, C.: Parametric schema inference for massive JSON datasets. The VLDB Journal **28**(4), 497–521 (Aug 2019)
4. Chapman, A., Simperl, E., Koesten, L., Konstantinidis, G., Ibáñez, L.D., Kacprzak, E., Groth, P.: Dataset search: A survey. The VLDB Journal **29**(1), 251–272 (Jan 2020)
5. Gao, Y., Huang, S., Parameswaran, A.: Navigating the Data Lake with DATAMARAN: Automatically Extracting Structure from Log Datasets. In: Proceedings of the 2018 International Conference on Management of Data - SIGMOD '18. pp. 943–958. ACM Press, Houston, TX, USA (2018)
6. Haller, D., Lenz, R.: Pharos: Query-Driven Schema Inference for the Semantic Web. In: Machine Learning and Knowledge Discovery in Databases, vol. 1168, pp. 112–124. Springer International Publishing, Cham (2020)
7. Jiang, L., Naumann, F.: Holistic primary key and foreign key detection. J Intell Inf Syst (Jun 2019)
8. Ota, M., Müller, H., Freire, J., Srivastava, D.: Data-driven domain discovery for structured datasets. Proc. VLDB Endow. **13**(7), 953–967 (Mar 2020)
9. Papenbrock, T., Bergmann, T., Finke, M., Zwiener, J., Naumann, F.: Data profiling with metanome. Proc. VLDB Endow. **8**(12), 1860–1863 (Aug 2015)
10. Pomp, A., Kraus, V., Poth, L., Meisen, T.: Semantic Concept Recommendation for Continuously Evolving Knowledge Graphs. In: Enterprise Information Systems, vol. 378, pp. 361–385. Springer International Publishing, Cham (2020)
11. Pomp, A., Paulus, A., Jeschke, S., Meisen, T.: ESKAPE: Information Platform for Enabling Semantic Data Processing:. In: Proceedings of the 19th International Conference on Enterprise Information Systems. pp. 644–655. SCITEPRESS - Science and Technology Publications, Porto, Portugal (2017)