Towards Query Generation for PROV-O Data

Jun Zhao[1], HongHan Wu[2], and Jeff Z. Pan[2]

[1] Lancaster University
[2] Aberdeen University

Abstract For a new user of a PROV-O dataset to quickly understand what is in the dataset and what queries can be asked is a challenging task. To address this open issue we present a query generation approach by making use of data summary information about the dataset. We compare preliminary results of two different implementations and discuss open issues.

1 Introduction

The PROV-O Ontology (http://www.w3.org/TR/prov-o/) is aimed to provide the much-needed standard to enable provenance information interexchange between different information systems. Since its release in April 2013, it has been adopted by a number of provenance information systems[3], as well as Linked Data providers, like DBPedia, and etc. Although PROV-O is commonly used in these implementations, we found that using PROV-O to directly query a provenance-related dataset is still a challenge.

We selected three PROV-O datasets published as part of ProvBench2013[4] to evaluate whether we can use PROV-O to effectively retrieve provenance information. Of the three datasets, two of them are from the workflow domain and one from the social computing domain. We systematically executed a number of SPARQL queries expressed using PROV-O terms to benchmark the usage of PROV-O terms in each dataset. The results[5] show that most queries cannot return as complete results as expected. The reasons can be summarised as the following: 1) missing class declarations in the dataset, 2) PROV-O terms can be used differently to express the same provenance information, and 3) the usage of application-specific schemas in the datasets, which although extend the PROV-O schema, making it hard to retrieve complete results without reasoning support.

These issues are not unique to PROV-O data and are also commonly seen in the semantic web. Reasoning can help with issues 1, 3 and partially 2, but domain-specific vocabularies are not always published with data. Although performance of ontology-based query answering systems is improving everyday, often query completeness is compromised given the need for scalability. Query rewriting can be applied to address issue 2, but it still faces a series of open issues, e.g. large computation time or complicated rewriting results [8]. Using

[3] http://www.w3.org/2001/sw/wiki/PROV#Implementations
[4] https://sites.google.com/site/provbench/provbench-at-bigprov-13
[5] https://github.com/junszhao/prov-analytics

PROV-O to access a provenance dataset is commonly driven by two motivations: one is to achieve interoperable provenance information access by using a standard vocabulary, and the other, to obtain an understanding about the content of a PROV-O dataset when little is known about what is there and how to access it. Existing approaches can provide some help to the first need, but the latter is largely unmet.

In this position paper we present a query generation approach to address this open challenge. By query generation we refer to the process of generating valid queries to be executed upon a dataset. Query generation approach has been commonly used in the database as well as semantic web communities for testing purposes[4,6,7,3,2]. The main goal has been to generate valid queries that can cover a specific set of query characteristics, such as query complexity, result size, query structure, etc. Although these approaches can produce helpful queries for testing purposes, they are not appropriate for the data understanding task. Data summarisation is another popular approach for assisting data understanding and query design [1]. However, although the statistical summaries about datasets can be useful to query optimisations and query federations, they provide limited help for users to know the kind of queries that they can ask.

We propose a data profile-led query generation approach, which automatically produces a set of valid SPARQL queries that can be executed against a PROV-O dataset based on a profile summary about the dataset. The resulting queries closely reflect the content of the dataset and provide a helping starting point for understanding the dataset and further query constructions. We compare two different implementations, one taking a customised approach by focusing on generating *PROV-O* queries and the other taking a more generic approach by finding key subgraphs (nodes/edges). We compare preliminary results of these two approaches and identify a set of open issues to be addressed.

2 A Profile-Led Provenance Query Generation Approach

2.1 The General Approach

Fig. 1 provides an outline of a profile-led provenance query generation approach. The key idea is that taking a PROV-O dataset, the framework will generate a set of valid (PROV-O) SPARQL queries for this dataset, so that data consumers can gain understandings about the content of the dataset. This simple framework is targeted at developers and provenance researchers who are proficient in SPARQL and are interested in designing provenance information access and integration services and systems. The two key components are a provenance query generator and a data profile generator: the *provenance query generator* will produce a set of provenance SPARQL queries based on its knowledge about what is inside a PROV-O dataset, that is produced by the *data profile generator*.

Data content summary is expected to provide useful information about the structure and content of a dataset. This information has been used in vocabularies like VoID to facilitate data understanding and query constructions. In these work data summary is expressed as statistical information that describes, e.g, the

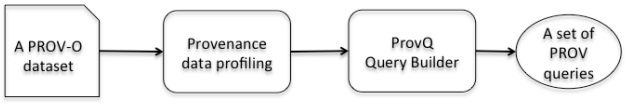

Figure 1. An outline of a provenance query generator framework, which contains two key components: a data profile generator and a provenance query generator.

types of classes/properties used in the data, the number of each class/property used in the data, or something more complex, like the types of properties and the number of unique objects linked to each property[6]. To facilitate query generation, our *data profile generator* component should produce more extensive summary information, which is more like the E-R diagram in relational database systems, to reveal all the relations among individuals in a dataset.

In the following we present two current implementations, one of which takes a generic approach and another taking a more tailored approach for PROV-O dataset. We compare the results from these two implementations and their respective strength.

2.2 The K-Drive Generator

The first implementation is an adaptation of the K-Drive Generator, which was previously proposed by Pan et al [5]. K-Drive Generator is designed as a generic query generation framework for semantic data. It can produce insightful queries for an input dataset by identifying typical graph patterns in the datasets. In order to reduce the graph pattern mining space, the graph pattern mining process is guided by a data summary graph. Different mining techniques have been used to produce the summary graph, including inductive logic programming (ILP) and association rule mining. When K-Drive Generator was applied to the Taverna PROV-O dataset, which was published as part of ProvBench 2013, we focused on the key properties used in the datasets and other properties often used together with these properties. This is because classes are often not declared in the Taverna PROV-O dataset, which makes it more challenging to generate queries based on information about class usage. All the queries [7] returned by K-Drive Generator are meaningful and provide a useful starting point for querying this dataset.

2.3 The ProvQ Implementation

Profile summary can be a performance challenging task, particularly for large datasets. Since we are particularly interested in provenance queries, we could reduce the profile summary space further more by focusing on summarising only

[6] https://github.com/joejimbo/HCLSDatasetDescriptions
[7] http://homepages.abdn.ac.uk/honghan.wu/pages/prov2/index.html

the usage of PROV-O terms and terms used together with them for describing the same resources. This is a classical technique that is used in association rule mining for identification of schema information. Using this PROV-O centric data summary information, we can produce a set of provenance queries that closely reflect the usage of provenance-related terms in the dataset by expanding a set of pre-defined atomic 'seed' provenance queries expressed using PROV-O. This approach is currently being implemented in the ProvQ framework that is partially based on an existing data profiling tool, ProLOD++[8].

3 Discussions and Future Work

We compare the queries generated by K-Drive Generator and ProvQ[9]. Both approaches returned 7 queries for the Taverna PROV-O dataset: 3 queries were largely the same, 3 queries were only returned by K-Drive, and the rest had different degrees of overlap. The three largely similar queries apply the same graph pattern but request different variables to be bound and returned.The 3 queries returned by K-Drive used no provenance-related properties in their query patterns. Generally speaking K-Drive can produce more complex candidate queries that can be a combination of several queries produced by ProvQ. A concept is often used in the queries returned by K-Drive. Certainly this makes it easier to interpret the queries, but the queries can be more restricted and return less results. In conclusion although the queries are different, both approaches produced queries that can provide useful starting points to query the PROV-O dataset. We would like to perform more assessment on the usefulness of the results given a set of defined query tasks. So far we have not come across any performance issues. A trade-off between performance and query generation completeness probably needs further investigation. Both approaches did not make use of reasoning, which could impact on the complexity of the resulting query patterns and completeness of query results. This is part of our future work.

4 Conclusions

In this position paper we propose a profile-led approach for generating queries for PROV-O datasets, in order to provide a useful starting point for designing queries for these datasets. We compare queries produced by two different implementations, one more generic for semantic data and one more tailored for PROV-O datasets. Both implementations returned meaningful and useful provenance queries with reasonable performance, but with different degrees of query pattern complexity. This shows that a query generation approach is an effective way to present a set of queries that can be applied to a PROV-O dataset and used as a starting point to construct further queries.

Although our initial results are encouraging, there are many open research questions to be addressed, such as whether reasoning should be incorporated in

[8] http://www.hpi.uni-potsdam.de/naumann/projekte/prolod.html
[9] https://github.com/junszhao/ProvQ

the query generation process and what type of reasoning, whether this approach can produce meaningful queries for querying across multiple PROV-O datasets, or whether more complex query patterns (such as `FILTER`, different types of `JOIN`s or variable binding) should also be generated.

References

1. Alexander, K., Hausenblas, M., Cyganiak, R., Zhao, J.: Describing linked datasets-on the design and usage of void. In: Proc. of Linked Data on the Web Workshop (LDOW 09), in conjunction with WWW 09 (2009)
2. Görlitz, O., Thimm, M., Staab, S.: Splodge: systematic generation of sparql benchmark queries for linked open data. In: Proc. of the 11th International Semantic Web Conference. pp. 116–132 (2012)
3. Harth, A., Hose, K., Karnstedt, M., Polleres, A., Sattler, K.U., Umbrich, J.: Data summaries for on-demand queries over linked data. In: Proc. of the 19th WWW. pp. 411–420 (2010)
4. Mishra, C., Koudas, N., Zuzarte, C.: Generating targeted queries for database testing. In: Proc. of the 2008 ACM SIGMOD. pp. 499–510 (2008)
5. Pan, J.Z., Ren, Y., Wu, H., Zhu, M.: Query generation for semantic datasets. In: Proc. of the 7th K-Cap. pp. 113–116 (2013)
6. Schmidt, A., Waas, F., Kersten, M., Carey, M.J., Manolescu, I., Busse, R.: Xmark: A benchmark for xml data management. In: Proc. of the 28th VLDB. pp. 974–985 (2002)
7. Slutz, D.R.: Massive stochastic testing of sql. In: Proc. of the 24th VLDB. vol. 98, pp. 618–622 (1998)
8. Tsalapati, E., Stoilos, G., Stamou, G., Koletsos, G.: Query rewriting under ontology contraction. In: Proc. of the 26th International Workshop on Description Logics. pp. 172–187 (2013)