Analyzing SPARQL Query Logs: A Study on Schema Coverage and Query Content

Maryam Mohammadi[1,*], Michel Dumontier[1]

[1]Institute of Data Science, Maastricht University, Paul-Henri Spaaklaan 1, 6229 GT, Maastricht, Netherlands

Abstract

The advent of Knowledge Graphs (KGs) has revolutionized knowledge representation, enabling enhanced understanding, reasoning, and interpretation of complex data for both humans and machines. As a crucial tool for addressing real-world challenges, KGs rely heavily on SPARQL queries for data access and manipulation. Despite their extensive use, the alignment of these queries with the underlying KG schema is not well-understood. This paper introduces 'SPARQL schema coverage' as a novel measure to assess the extent to which SPARQL queries reflect the KGs' content and structure. Utilizing Bio2RDF SPARQL logs as a case study, the paper reveals a SPARQL schema coverage of 98%, demonstrating a strong alignment between user queries and the KG schema, thereby highlighting high user engagement. This finding is significant for KG engineers in reshaping ontology and for triple store administrators in enhancing performance through targeted caching. The study addresses key research questions regarding the nature and extent of KG elements referred to in user queries and their coverage of the available data. This approach not only provides a new perspective in KG utilization but also aids in optimizing KG design and application, offering valuable insights for the future development and optimization of knowledge graphs.

Keywords

RDF Knowledge Graph, SPARQL Query Logs, SPARQL Shema Coverage

1. Introduction

The advent of Knowledge Graphs (KGs) has revolutionized knowledge representation, enhancing the understanding, reasoning, and interpretation of complex data for both humans and machines [1]. As the primary tool for a myriad of real-world challenges, KGs have necessitated sophisticated tools for data access and manipulation, with SPARQL queries playing a pivotal role [2]. However, despite the extensive use of SPARQL queries, our understanding of their alignment with the underlying KG schema remains limited. This paper seeks to bridge this gap by introducing a novel measure of 'SPARQL schema coverage' to assess the extent to which SPARQL queries reflect the content and structure of the KGs.

Previous studies have predominantly focused on the syntactical aspects of SPARQL queries or their general structure [3, 4, 5, 6], neglecting the rich insights that can be gleaned from their content. For instance, Asprino et al. (2023) [7] emphasized categorizing SPARQL queries into templates to identify common usage patterns, while Bielfeldt et al. (2018) [8] differentiated between organic and robotic queries in Wikidata, uncovering distinct patterns in human and automated querying behaviors. These studies, while insightful, stop short of examining how comprehensively these queries cover the schema of the targeted KGs.

This paper aims to fill this research gap. We propose a comprehensive analysis method to determine how well user queries in SPARQL logs align with the KG's schema. This study involves a case study revealing a schema coverage of Bio2RDF SPARQL logs. This insight is crucial for KG engineers and

SWAT4HCLS 2024: The 15th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences, February 26–29, 2024, Leiden, The Netherlands

*Corresponding author.

m.mohammadi@maastrichtuniversity.nl (M. Mohammadi); michel.dumontier@maastrichtuniversity.nl (M. Dumontier)

triple store administrators, guiding them in optimizing KG performance and reshaping ontology models. Our research addresses two fundamental questions:

1) To what extent do user queries in SPARQL logs refer to specific elements of the knowledge graph? 2) How comprehensive is this coverage in terms of the diversity of data available in the knowledge graph?

By answering these questions, this study not only contributes to the technical understanding of KG utilization but also provides practical insights for optimizing KG systems for enhanced user engagement and efficiency.

2. Related work

Previous work by Asprino et al. (2023) [7] introduced a content-centric method on LSQ logs. Their method, termed 'query log summarization', categorizes SPARQL queries into common templates based on their content. In their experiments, they examined the ten most executed templates for each log. This examination led to the identification of common usage patterns and the prevalence of queries originating from a single code source. Moreover, their study explored template relationships, uncovering evidence that related templates often participate in a common process, frequently executed by a similar set of hosts. Additionally, the authors observed relationships between different queries applied to data across various logs, indicating a systematic, automated approach to data querying.

Bielfeldt et al. (2018) [8] introduced the concept of organic and robotic SPARQL queries as a fundamental principle for query log analysis. They proposed a method to partition a dataset of over 200 million SPARQL queries of Wikidata into organic and robotic queries. They defined robotic queries as those generated by software tools or bot-like agents, while organic queries originated from browser-like agents. By distinguishing between these two types of queries, the authors identified clear patterns in human usage, while observing that the robotic component displayed high volatility and unpredictability, even over extended time periods. In their experiments, they showed that organic and robotic traffic are significantly different in many respects. For instance, from examining the most frequent Wikidata properties used in queries as annotations on statements and the most frequent Wikidata properties of statements whose complex form occurs in queries, they conclude that complex statements are a larger fraction in organic queries. Another finding from their experiments suggests that few users from Asia are accessing Wikidata via SPARQL-based applications. Although these studies focus on content-centric analysis of queries, to the best of our knowledge to date, there are no existing studies that show whether SPARQL query logs provide comprehensive coverage of the schema of the target KG, or if certain sections of the target KG remain largely unexplored.

3. Method

This section outlines the workflow of our method, illustrated in Figure 1. The workflow comprises four main steps. The first step involves gathering all schema elements from the KG. Next, we clean the SPARQL logs in preparation for the third step, which entails extracting the schema elements used within these logs. The final step involves calculating the SPARQL schema coverage, as detailed in Equation 1, to assess how well the queries in the logs represent the KG's schema. Detail of each step are described below:

In Step 1, we retrieve the Bio2RDF schema elements by executing a series of queries against the KG, as depicted in Figure 2. We initiate this process with Query 1, which isolates Bio2RDF-specific schema elements, deliberately excluding other domain datasets linked to Bio2RDF, such as "http://www.openlinksw.com/schemas/virtrdf#". Following this, Query 2 is employed to extract unique classes from the Bio2RDF graphs. Notably, our approach ensures that only those classes with at least

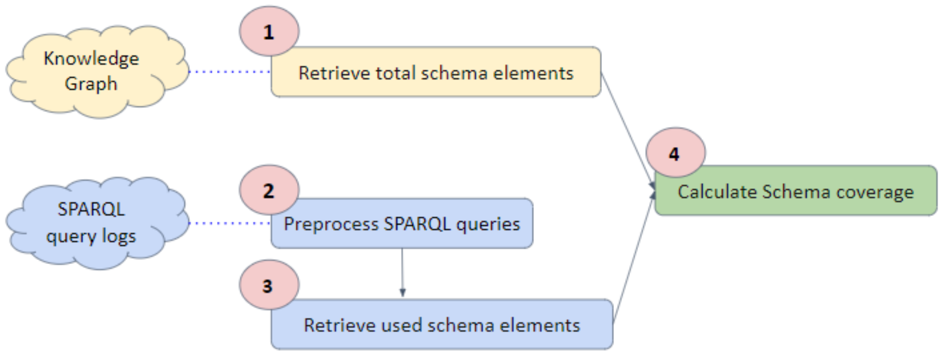
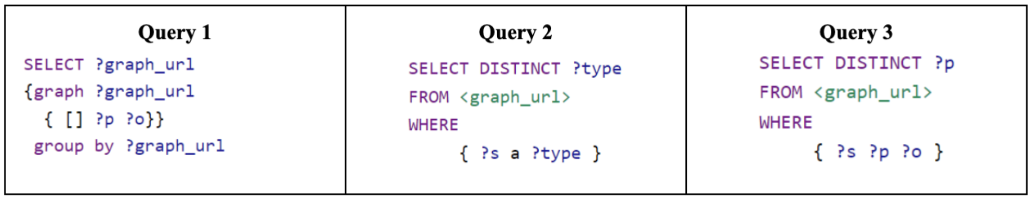

Knowledge Graph
SPARQL query logs
1
Retrieve total schema elements
2
Preprocess SPARQL queries
3
Retrieve used schema elements
4
Calculate Schema coverage

Figure 1: Workflow of proposed method

one instance are returned, omitting unused classes like “http://bio2rdf.org/hgnc_vocabulary:Status”. Finally, Query 3 is executed to retrieve unique schema predicates.



Query 1

```
SELECT ?graph_url
{graph ?graph_url
  { [] ?p ?o}}
 group by ?graph_url
```

Query 2

```
SELECT DISTINCT ?type
FROM <graph_url>
WHERE
     { ?s a ?type }
```

Query 3

```
SELECT DISTINCT ?p
FROM <graph_url>
WHERE
     { ?s ?p ?o }
```

Figure 2: Query 1, 2 and 3 to retrieve Bio2RDF Graphs, schema Classes and schema Predicates

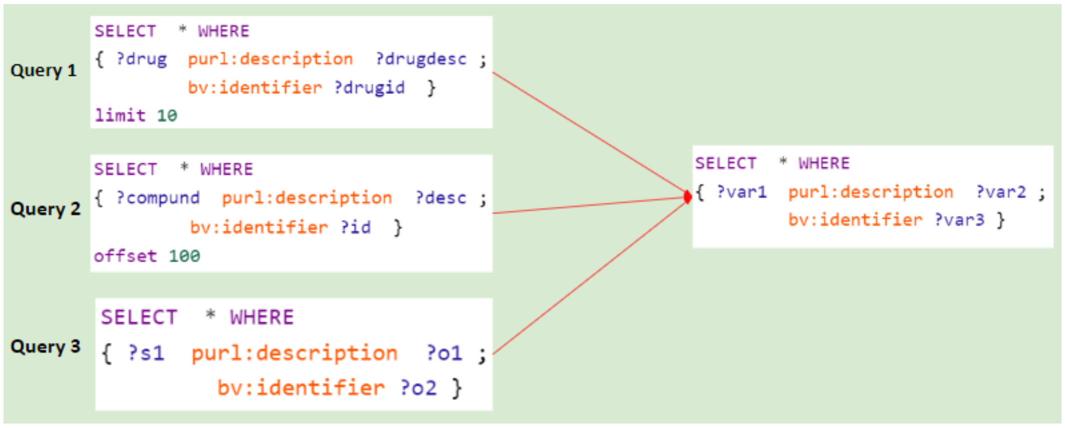
In Step 2, we downloaded and perform data cleaning on Bio2RDF SPARQL query logs from the period 2019-2021, which contain 3.880.939 queries, available at https://download.dumontierlab.com/bio2rdf/logs/. We applied the following processes: the elimination of redundant HTTP Parameters, variable standardization, and prefix addition to remove repetitive and invalid queries. The output of this preprocessing is unique, normalized, valid queries.

For the elimination of redundant HTTP Parameters, queries were stripped of non-essential HTTP parameters to resolve parsing issues. We further standardized the queries to ensure uniformity. This step is crucial because, in the next phase of our method, we aim to distinguish variables from actual entities. By substituting variable names with placeholders like ’varX’, we achieve this distinction. For variable standardization, names were standardized as shown in Figure 3. For example, three different variable names (e.g., ’?drug’, ’?compound’, and ’?s1’) were changed to ’?var1’.

For the Prefix Addition step, essential RDF syntax prefixes (such as rdf, rdfs, owl, etc.) were incorporated to enhance the syntactical validity of the queries. This addition was necessary because some queries lacked the required prefix declarations. In the context of the Virtuoso endpoint, users are not obliged to explicitly include these prefixes in their queries, as it contains pre-registered prefix declarations. However, our parser, ’sparqljs’ [9], a SPARQL 1.1 parser for JavaScript, implemented in Java, did not have this feature. Therefore, we manually added these prefixes to ensure proper parsing of the SPARQL queries.

In Step 3, triple patterns are extracted from the queries, and unique schema elements are then isolated from these triples. To identify schema classes, the type of URLs or literals in subject and object positions is determined using Query 4 and Query 5 as depicted in Figure 4. For schema predicates, Query 5 is executed to check their validity within the Bio2RDF schema. The entities that start with ’var’ (indicating a Variable node resulting from the variable standardization step), or ’g_’ (indicating a blank node as per sparqljs parser) are disregarded.

In Step 4, we calculate SPARQL schema coverage using the results from Steps 1 and 3, applying the



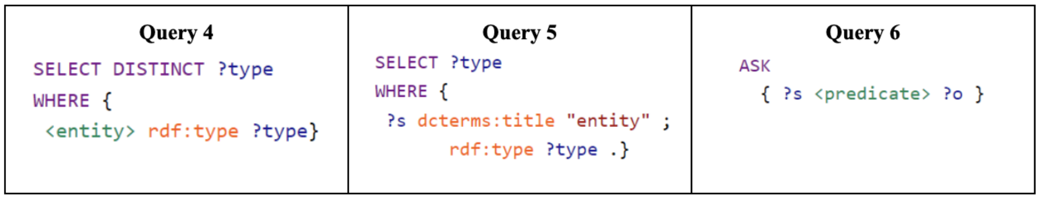
```
SELECT  * WHERE
{ ?drug  purl:description  ?drugdesc ;
         bv:identifier ?drugid  }
limit 10
```

```
SELECT  * WHERE
{ ?compund  purl:description  ?desc ;
         bv:identifier ?id  }
offset 100
```

```
SELECT  * WHERE
{ ?s1  purl:description  ?o1 ;
         bv:identifier ?o2 }
```

```
SELECT  * WHERE
{ ?var1  purl:description  ?var2 ;
         bv:identifier ?var3 }
```

Figure 3: Variable standardization



Query 4

```
SELECT DISTINCT ?type
WHERE {
 <entity> rdf:type ?type}
```

Query 5

```
SELECT ?type
WHERE {
 ?s dcterms:title "entity" ;
      rdf:type ?type .}
```

Query 6

```
ASK
  { ?s <predicate> ?o }
```

Figure 4: Query 4, 5, and 6 to retrieve types for URL entities, literal entities and predicates

formula outlined in Equation 1. This equation involves the count of all distinct classes and predicates in the Bio2RDF dataset (Total Schema Elements, or TSE) and the count of all distinct classes and predicates used in user SPARQL query logs (Used Schema Elements, or USE). The SPARQL schema coverage (SC) is then calculated using Equation 1. The code for this Method is available on our GitHub repository: https://github.com/marmhm/SPARQL_queries.

$$\text{SC (\%)} = \frac{\text{USE}}{\text{TSE}} \times 100 \tag{1}$$

4. Results

In this section, we describe the outcomes from the different steps of our method. Query 1, in Figure 1 generated 44 results, of which only 26 were used, as these are graphs within the Bio2RDF domain. This was determined by filtering out results that do not have “bio2rdf.dataset” in their graph URL. In the preprocessing step, from an initial dataset of 3.880.939 queries, we derived 633.453 unique and valid queries. It is important to note that the addition of prefixes allowed for the parsing of an additional 8.176 queries.

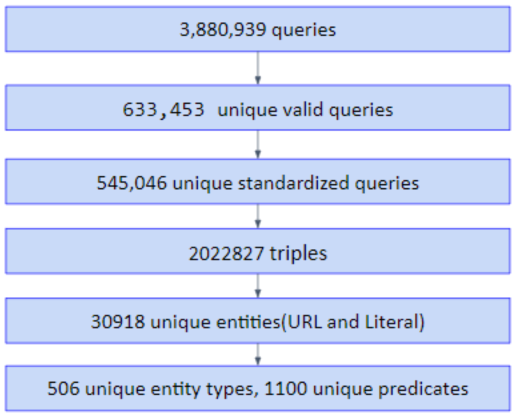

3,880,939 queries
633,453 unique valid queries
545,046 unique standardized queries
2022827 triples
30918 unique entities(URL and Literal)
506 unique entity types, 1100 unique predicates

Figure 5: Results of step 2 and step 3

The variable standardization process further refined the dataset to 545.046 queries. The details of these findings are summarized in Figure 5. Our analysis extracted 2.022.827 triples from these queries, which included 30.918 unique entities (URLs and Literals), 506 unique entity types, and 1.100 unique predicates.

Table 1
Statistics of schema elements from queries and KG

Source	*Bio2RDF SPARQL Query logs*	*Bio2RDF Knowledge Graph*
Total number of distinct Elements	1606	1635
Number of distinct Classes	506	532
Number of distinct Predicates	1100	1106

Table 1 shows the statistics of schema elements derived from the queries and the Knowledge Graph (KG). The total number of distinct schema elements in the Bio2RDF KG stands at 1.635, comprising 532 distinct classes and 1.106 distinct predicates. In comparison, the query logs of Bio2RDF showed a total of 1.606 distinct schema elements, with 506 distinct classes and 1.100 distinct predicates. By applying Equation 1, we assessed the SPARQL schema coverage (SC) as follows:

$$\text{SC } (\%) = \frac{\text{USE}}{\text{TSE}} \times 100 = \frac{1606}{1635} \times 100 = 98\% \tag{2}$$

This assessment quantifies the extent of Bio2RDF schema utilization, indicating a substantial coverage of approximately 98%.

5. Conclusion and Future Work

This study sheds light on the alignment between SPARQL query logs and the Knowledge Graphs (KGs) schema, enhancing our understanding of schema coverage of queries. This work reveals that the SPARQL queries make references to nearly the entirety of the actual RDF schema. This result is surprising because we expected that only some parts of the RDF graph would be of interest to users. Historically, there are two main reasons for the inclusion of datasets: (1) to support query answering / data mining use cases that spanned a subset of elements across multiple dataset, and (2) the opportunistic inclusion of new or popular datasets as biological linked data [10]. For future work, we plan to undertake two main initiatives: Firstly, we aim to incorporate additional datasets, like the LSQ, to broaden our analysis. Secondly, we intend to segment the dataset used in this paper into shorter periods, such as six or three-month intervals, to investigate how schema coverage fluctuates over these timescales. These steps are expected to deepen our understanding of schema usage patterns and improve the methodology's applicability and effectiveness in diverse real-world scenarios.

6. Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 860801.

References

[1] B. Abu-Salih, Domain-specific knowledge graphs: A survey, Journal of Network and Computer Applications 185 (2021) 103076.

[2] W. Ali, M. Saleem, B. Yao, A. Hogan, A.-C. N. Ngomo, A survey of rdf stores & sparql engines for querying knowledge graphs, The VLDB Journal (2022) 1–26.

[3] A. Bonifati, W. Martens, T. Timm, An analytical study of large sparql query logs, The VLDB Journal 29 (2020) 655–679.

[4] C. Buil-Aranda, M. Ugarte, M. Arenas, M. Dumontier, A preliminary investigation into sparql query complexity and federation in bio2rdf, in: Alberto mendelzon international workshop on foundations of data management, 2015, p. 196.

[5] M. Saleem, M. I. Ali, A. Hogan, Q. Mehmood, A.-C. N. Ngomo, Lsq: the linked sparql queries dataset, in: The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II 14, Springer, 2015, pp. 261–269.

[6] C. Stadler, M. Saleem, Q. Mehmood, C. Buil-Aranda, M. Dumontier, A. Hogan, A.-C. Ngonga Ngomo, Lsq 2.0: A linked dataset of sparql query logs, Semantic Web (2022) 1–23.

[7] L. Asprino, M. Ceriani, How is your knowledge graph used: Content-centric analysis of sparql query logs, in: International Semantic Web Conference, Springer, 2023, pp. 197–215.

[8] A. Bielefeldt, J. Gonsior, M. Krötzsch, Practical linked data access via sparql: The case of wikidata., in: LDOW@ WWW, 2018.

[9] SPARQL.js, Sparql.js-a sparql 1.1 parser for javascript https://www.npmjs.com/package/sparqljs, 2023. URL: https://www.npmjs.com/package/sparqljs.

[10] A. Callahan, J. Cruz-Toledo, P. Ansell, M. Dumontier, Bio2rdf release 2: improved coverage, interoperability and provenance of life science linked data, in: The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10, Springer, 2013, pp. 200–212.