Online Sampling of Summaries from Public SPARQL Endpoints

Thi Hoang Thi Pham, Pascal Molli, Hala Skaf-Molli, Brice Nédelec

Online Sampling of Summaries from Public SPARQL Endpoints

Thi Hoang Thi Pham
thi-hoang-thi.pham@univ-nantes.fr
Nantes Université
Nantes, France

Hala Skaf-Molli
hala.skaf@univ-nantes.fr
Nantes Université
Nantes, France

Pascal Molli
pascal.molli@univ-nantes.fr
Nantes Université
Nantes, France

Brice Nédelec
brice.nedelec@univ-nantes.fr
Nantes Université
Nantes, France

ABSTRACT

Collecting statistics from online public SPARQL endpoints is hampered by their fair usage policies. These restrictions hinder several critical operations, such as aggregate query processing, portal development, and data summarization. Online sampling enables the collection of statistics while respecting fair usage policies. However, sampling has not yet been integrated into the SPARQL standard. Although integrating sampling into the SPARQL standard appears beneficial, its effectiveness must be demonstrated in a practical semantic web context. This paper investigates whether online sampling can generate summaries useful in cutting-edge SPARQL federation engines. Our experimental studies indicate that sampling allows the creation and maintenance of summaries by exploring less than 20% of datasets.

CCS CONCEPTS

• **Information systems → Database query processing**.

KEYWORDS

SPARQL, Sampling, Federation, Summary

ACM Reference Format:
Thi Hoang Thi Pham, Hala Skaf-Molli, Pascal Molli, and Brice Nédelec. 2024. Online Sampling of Summaries from Public SPARQL Endpoints. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651543

1 INTRODUCTION

Public SPARQL endpoints like DBPedia or Wikidata serve thousands of queries daily. However, the fair usage policies of SPARQL endpoints block the collection of simple statistics or basic information online about DBPedia and Wikidata. For example, consider the query QB2 of SPORTAL [7] in Figure 1. QB2 counts the number of available classes. The execution of this simple query is stopped after 60s on DBPedia and Wikidata, delivering no results. Consequently, many downstream tasks such as processing aggregate queries [6], creating portals [7, 8] or computing summaries for federation engines [1, 3, 10, 11, 13, 15–17], cannot be done online.

```
SELECT( COUNT( DISTINCT ?c ) as ?count ) { ?s a ?c }
```

Figure 1: Query QB2 of SPORTAL [7] times out after 60s on DBPedia and Wikidata due to time quotas.

Collecting offline statistics can be done by downloading datasets [12]. This supposes that datasets are available offline and raises the issue of the freshness of statistics. It is also recommended for SPARQL endpoint providers to pre-compute some statistics using the VOID vocabulary[1]. However, such practice is poorly adopted on the Web [8]. It is also possible to rely on new RDF interfaces for online SPARQL query processing, such as TPF [18] and web preemption [9]. If such approaches preserve fair usage by design, they require a new kind of RDF server that is not standard and, consequently, is poorly adopted. The question is: *"What enhancements to the SPARQL standard are necessary for online computation of statistics?"*. This question is challenging as any proposal faces the problem of fair usage policies and possible adoption by the Semantic Web community.

Recently, RAW-JENA [2] proposed sampling for computing statistics online. An extension of Apache Jena has been released to prove the feasibility of sampling in a reference implementation of SPARQL. Sampling avoids the problem of fair usage policies by allowing sampling to continue even if stopped, i.e., sampling is interrupted, partial results are usable, and sampling may continue in another sampling query. Concerning standardization, sampling has been part of SQL standard since 2005, with the `TABLESAMPLE` syntax. Surprisingly, sampling is still not supported in the SPARQL standard in 2024. Although incorporating sampling into the SPARQL standard appears to be beneficial, its actual effectiveness for real-world semantic web applications must be verified. Sampling inherently faces accuracy issues stemming from the processing of incomplete data, which can have substantial implications for subsequent tasks.

In this paper, we evaluate the effectiveness of a sampling-based approach for computing summaries online for federation engines, a mainstream approach for keeping the semantic web decentralized [3]. If these summaries can be computed with SPARQL queries online, they cannot terminate in real settings due to fair usage policies, preventing the deployment of federation engines.

[1] https://www.w3.org/TR/void/

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05
https://doi.org/10.1145/3589335.3651543

1	http://v16.fr/Offer42	isa	bsbm:Offer	http://v16.fr/
2	http://v16.fr/Offer42	bsbm:price	549.29	http://v16.fr/
3	http://v16.fr/Offer42	bsbm:prod	http://v16.fr/Prod75992	http://v16.fr/
4	http://v16.fr/Prod75992	rdf:label	SamSoung 24	http://v16.fr/
5	http://v16.fr/Prod75992	owl:sameAs	http://ocp/Prod75992	http://v16.fr/
6	http://v16.fr/Offer1337	isa	bsbm:Offer	http://v16.fr/
7	http://v16.fr/Offer1337	bsbm:price	1549.29	http://v16.fr/
8	http://v16.fr/Offer1337	bsbm:prod	http://v16.fr/Prod73	http://v16.fr/
9	http://v16.fr/Prod73	rdf:label	PommePhone 14	http://v16.fr/
10	http://v16.fr/Prod73	owl:sameAs	http://ocp/Prod73	http://v16.fr/

1	http://r0.fr/Review2466	isa	bsbm:Review	http://r0.fr/
2	http://r0.fr/Review2466	bsbm:rating1	5	http://r0.fr/
3	http://r0.fr/Review2466	bsbm:reviewFor	http://r0.fr/Prod75992	http://r0.fr/
4	http://r0.fr/Prod75992	owl:sameAs	http://ocp/Prod75992	http://r0.fr/

(a) Federation $F1$ made of 2 SPARQL endpoints: Vendor16 and RatingSite0.

1	http://v16.fr/	isa	bsbm:Offer	http://v16.fr/
2	http://v16.fr/	bsbm:price	any	http://v16.fr/
3	http://v16.fr/	bsbm:prod	http://v16.fr/	http://v16.fr/
4	http://v16.fr/	rdf:label	any	http://v16.fr/
5	http://v16.fr/	owl:sameAs	http://ocp/	http://v16.fr/

1	http://r0.fr/	isa	bsbm:Review	http://r0.fr/
2	http://r0.fr/	bsbm:rating1	any	http://r0.fr/
3	http://r0.fr/	bsbm:reviewFor	http://r0.fr/	http://r0.fr/
4	http://r0.fr/	owl:sameAs	http://ocp/	http://r0.fr/

(b) FedUP's summary for $F1$.

Figure 2: A summary is a compact representation of a federation made of quads.

```
SELECT * WHERE {
 ?offer  bsbm:price ?price .    #tp1
 ?offer bsbm:product ?lp1 .     #tp2
 ?lp1  owl:sameAs ?product .    #tp3
 ?lp2 owl:sameAs ?product .     #tp4
 ?review bsbm:reviewFor ?lp2 .  #tp5
 ?review bsbm:rating1 ?rating }#tp6
```

(a) Federated Query Q1 that finds Offer with Reviews.

```
SELECT * WHERE {
SERVICE <http://v16.fr> {
 ?offer  bsbm:price ?price .    #tp1
 ?offer bsbm:product ?lp1 .     #tp2
 ?lp1  owl:sameAs ?product }    #tp3
SERVICE <http://r0.fr> {
 ?lp2 owl:sameAs ?product .     #tp4
 ?review bsbm:reviewFor ?lp2 .  #tp5
 ?review bsbm:rating1 ?rating }}#tp6
```

(b) $Q1_s$ Service Query for $Q1$ after decomposition.

Figure 3: Federated Query $Q1$ with its decomposition $Q1_s$.

In this paper, we present the following contributions:

- We computed summary queries using online sampling in two representative federated benchmarks: FedShop [5], a synthetic benchmark and LargeRDFBench [14], a benchmark based on real datasets.
- We evaluated the completeness of summaries per sample size.
- Experimental results show that 90% of summaries can be acquired with a sample size less than 3% of the dataset size.
- Experimental results show that when the federated workload is known, sample efficiency can be greatly increased.

This paper is organized as follows: Section 2 presents the problem of getting summaries for federation engines from online SPARQL endpoints. Section 3 describes how online sampling works with RAW-JENA [2]. Section 4 presents our experimental results conducted on LargeRDFBench and FedShop. Section 5 concludes and outlines future works.

2 FEDERATION ENGINES AND SUMMARIES

Consider the federation F_1 in Figure 2a composed of 2 SPARQL endpoints: Vendor16 and RatingSite0. Following the FedShop use-case [5], vendors sell products through offers while rating sites provide reviews for products. Consider the federated query $Q1$ of Figure 3a defined on FedShop schema [5], $Q1$ finds offers with reviews. A federation engine can evaluate $Q1$ over the federation of Vendor16 and RatingSite0. During query processing, $Q1$ is decomposed into subqueries executable on SPARQL endpoints of $F1$. A

```
CONSTRUCT { ?ps ?p ?o } WHERE {
 ?s ?p ?o . FILTER (isIRI( ?s ))
 BIND( URI( REPLACE( STR( ?s ),"^(http ?://?.*?/.* ","$1")) AS ?ps )
 BIND( IF( isIRI( ?o ), URI( REPLACE( STR( ?o ), "^(http ?://?.*?/.* ","$1" )),
                    "any") AS ?po )}
```

Figure 4: FedUP's summary query Q_f.

```
CONSTRUCT { ?p <SA> ?ps . ?p <OA> ?po } WHERE {
 ?s ?p ?o . FILTER (isIRI( ?s ))
 BIND( URI( REPLACE( STR( ?s ),"^(http ?://?.*?/.* ","$1")) AS ?ps )
 BIND( IF( isIRI( ?o ), URI( REPLACE( STR( ?o ), "^(http ?://?.*?/.* ","$1" )),
                    "any") AS ?po )}
```

Figure 5: HiBISCuS' summary query Q_h.

correct decomposition for evaluating $Q1$ on $F1$ is to evaluate $tp1$, $tp2$, and $tp3$ on the SPARQL endpoint of Vendor16 and join with $tp4$, $tp5$ and $tp6$ on RatingSite0 as shown by the SPARQL service query $Q1_s$ in Figure 3b.

Federation engines [3, 4, 16] rely on summaries obtained from endpoints to make this decomposition. Figure 2b presents the summary of the recent federation engine FedUP [3]. Such a summary only keeps the subject authority and object authority of URLs and projects all literals to a single literal "any" (except for `isa` predicates). This summary can be computed online using the SPARQL summary query Q_f in Figure 4. However, Q_f will not terminate on public SPARQL endpoints such as Wikidata or DBPedia. As Q_f scans all triples in the SPARQL endpoints; there is no chance to scan more than 1B triples in less than 60s.

We have the same issue for HiBISCuS-like summaries used in HiBISCuS itself [15] or CostFed [16]. A suitable summary query Q_h for HiBISCuS is presented in Figure 5. In contrast to FedUP summaries, an HiBISCuS summary does not keep the link between subjects and objects. Again, Q_h scans all triples of the SPARQL endpoints and cannot terminate in less than 60s on large graphs.

To overcome this problem, we study if Q_f or Q_h can be sampled, and then, the sample size required for high accuracy.

3 SAMPLING WITH RAW-JENA

RAW-JENA [2] is an online sampling open-source extension of Apache Jena. Given a SPARQL conjunctive query and a number of draws, RAW-JENA returns a sample of query results with its cardinality estimation. The number of sampled results and the accuracy of the cardinality estimation increase over draws.

Thanks to RAW-JENA, we can now sample Q_f and Q_h and get random results with cardinalities. To compute the summary of Figure 2b, we sample Q_f on the SPARQL endpoints of Vendor16 and RatingSite0 of the federation $F1$. As Q_f has only one single triple pattern `?s ?p ?o`, RAW-JENA returns random triples from endpoints, with the cardinality estimation of `?s ?p ?o`.

Suppose we evaluate Q_f with 2 draws on the endpoint of Vendor16 powered by RAW-JENA, and we randomly select the *price of Offer42* (Triple 2 of Figure 2a) and the *sameAs predicate of Product73* (Triple 10 of Figure 2a). RAW-JENA returns the following two triples with a cardinality estimation of 10:

(1) `http://v16.fr/ bsbm:price any`
(2) `http://v16.fr/ owl:sameAs http://ocp/`

Consequently, we know that we explored 20% of Vendor16 and obtained $\frac{2}{5}$ of its summary. When the number of draws increases, we eventually converge to the complete summary of Vendor16. However, it remains impossible to know when completeness is reached. Therefore, we empirically evaluate the number of draws required in real situations to approximate complete summaries.

Interestingly, sampling is very practical for summary maintenance, i.e., it allows continuous sampling of the federation with time-to-live attached to the summary entry. In this way, the addition or deletion of federation members, or the modification of the content of one member, will eventually be reflected in the summary.

4 EXPERIMENTAL STUDY

The goal of this experimental study is to empirically answer the following questions: (1) How many draws are required to reach a complete summary for two representative federated benchmarks, namely FedShop200 [5] and LargeRDFBench [14]? (2) Does the knowledge of the query workload improve sample efficiency?

The client code of the experiment is available on GitHub at https://github.com/phamthi1812/online-endpoint-summaries.

4.1 Experimental Setup

Dataset and Queries. We used two benchmarks for our study:

LargeRDFBench [14] is a federated benchmark based on 13 real-world datasets with more than 1B of quads. The benchmark includes different sets of queries; we used the 14 simple queries (S) and 10 complex queries (C).

FedShop [5] is a federated synthetic benchmark based on an e-commerce use-case. We reused the largest configuration of 200 vendors and rating sites with more than 43M quads. The query workload is based on 12 template queries with 10 random instantiations totaling 120 queries.

Federation configuration. We install the federations as named graphs into RAW-JENA using the TDB2 backend of Apache-Jena. As hardware, we used a local cloud instance with Ubuntu 20.04.4 LTS, an AMD EPYC 7513-Core processor with 16 vCPUs allocated to the VM, 1TB SSD, and 64GB of RAM.

Table 1: Characteristics of summaries.

	#Quads		Distinct		
Benchmark	Dataset	Summary	subjects	predicates	objects
FedShop-200	43 159 409	5800	200	38	203
LargeRDFBench	1 003 893 350	6070	208	2030	3227

Approaches. We consider the following summaries:

Complete summaries: For each SPARQL endpoint E_i of the federation, we evaluate Q_f over E_i and append results in the named graph $N(E_i)$. The summary of a federation is the dataset composed of all named graphs.

SPO-Sampling For each SPARQL endpoints E_i of the federation, we sample the Q_f with one draw over E_i and append results in the named graph $N(E_i)$. The completeness of the sampled summary is measured by dividing the sample summary's size by the complete summary's size. We repeat sampling until reaching 100% of completeness or when the number of draws reaches 20% of the number of the triples in the federation.

Workload-Aware (WA) Sampling We suppose that we know all triple patterns $tp_1, ..., tp_k$ of the workload. We build $Q_f^{tp_i}$ by rewriting the query Q_f with tp_i, i.e., we replace the `?s ?p ?o` triple pattern of Q_f with tp_i by variable substitutions. For each SPARQL endpoint E_i of the federation, for each tp_i of the workload, we sample $Q_f^{tp_i}$ with one draw over E_i and append the results in the named graph $N(E_i)$. We repeat sampling as with SPO-sampling.

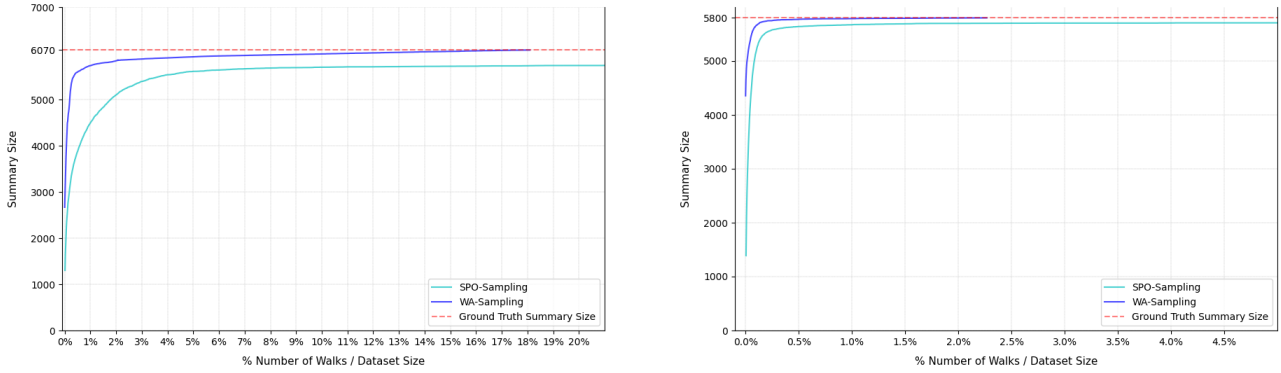
Summaries. Table 1 describes the characteristics of the complete summaries for both federated benchmarks. The size FedUP's summaries is tiny compared to the size of original datasets. For each summary, we measured the number of distinct predicates, distinct subject authorities, and distinct object authorities. LargeRDFBench has much more distinct predicate authorities and object authorities than FedShop.

4.2 Experimental results

Figure 6a presents the sample efficiency of SPO and WA sampling to build the complete summary of LargeRDFBench. The x-axis represents the ratio of draws over the dataset size, while the y-axis shows the summary size.

As expected, the overall evolution of the completeness is asymptotic. Finding 90% of the complete summary is very fast, but finding the remaining missing elements requires much more draws. SPO-sampling fails to find complete summaries before reaching the limit of draws. WA-sampling is significantly more efficient, i.e., a complete summary is reached after drawing 18% of the dataset size. SPO-sampling is biased by predicate frequency, i.e., as some predicates are less frequent than others, they are more difficult to find. WA-sampling requires no effort to find all predicates in the workload. However, as the triple pattern `?s ?p ?o` is in the workload, the remaining predicates may be difficult to find.

Figure 6b describes the sample efficiency of SPO and WA sampling on FedShop. We observe the same general asymptotic behavior observed with LargeRDFBench. As for the previous experiment,



(a) Online sampling for summary on LargeRDFBench. **(b) Online sampling for summary on FedShop200.**

Figure 6: Sample efficiency on sampling FedUP summaries with RAW-JENA.

WA-sampling is more efficient than SPO-sampling and reaches completeness after drawing less than 3% of the dataset size.

If we compare the two experiments on LargeRDFBench and FedShop, both sampling methods are much more efficient with FedShop. This is explained by the large difference between distinct predicate and distinct object authority described in Table 1.

5 CONCLUSION

The current state of SPARQL standards hinders the utilization of SPARQL endpoints for essential downstream tasks, including aggregate query processing, portals, and federation engines. For the first time, we have introduced a sampling approach capable of computing and maintaining summaries online across a federation of SPARQL endpoints. This method aligns with the fair usage policies of SPARQL endpoints and has proven effective for federation engines. Similarly to the SQL standard, the integration of sampling into the SPARQL standard warrants serious considerations.

For future work, federation summaries should also include cardinalities, which are essential for query optimization and join ordering. RAW-JENA can already provide cardinalities for sampled queries, enabling the integration of each predicate's cardinality into the summaries. However, accessing the cardinality of distinct results is often useful, as illustrated in Figure 1. Approximate count-distinct queries are challenging and currently not supported by RAW-JENA. Further research is needed to efficiently compute the count-distinct queries using sampling in RAW-JENA.

ACKNOWLEDGMENTS

This work is supported by the French Labex CominLabs project MiKroloG (The Microdata Knowledge Graph), and the French ANR project MeKaNo (ANR-22-CE23-0021).

REFERENCES

[1] Maribel Acosta, Maria-Esther Vidal, Tomas Lampo, Julio Castillo, and Edna Ruckhaus. 2011. ANAPSID: an adaptive query processing engine for SPARQL endpoints. In *10th International Semantic Web Conference (ISWC2011)*. Springer, Bonn, Germany, 18–34.

[2] Julien Aimonier-Davat, Minh-Hoang Dang, Pascal Molli, Brice Nédelec, and Hala Skaf-Molli. 2023. RAW-JENA: Approximate Query Processing for SPARQL Endpoints. In *22nd International Semantic Web Conference (ISWC'23)*. CEUR-WS.org, Athens, Greece, 5.

[3] Julien Aimonier-Davat, Minh-Hoang Dang, Pascal Molli, Brice Nédelec, and Hala Skaf-Molli. 2024. FedUP: Querying Large-Scale Federations of SPARQL Endpoints. In *The ACM Web Conference (WWW'24)*. ACM, Singapore, Singapore, 10.

[4] Angelos Charalambidis, Antonis Troumpoukis, and Stasinos Konstantopoulos. 2015. SemaGrow: Optimizing federated SPARQL queries. In *11th International Conference on Semantic Systems*. ACM, New York, NY, USA, 121–128.

[5] Minh-Hoang Dang, Julien Aimonier-Davat, Pascal Molli, Olaf Hartig, Hala Skaf-Molli, and Yotlan Le Crom. 2023. FedShop: A Benchmark for Testing the Scalability of SPARQL Federation Engines. In *International Semantic Web Conference (ISWC)*. Springer, Springer Nature Switzerland, Athens, Greece, 285–301.

[6] Arnaud Grall, Thomas Minier, Hala Skaf-Molli, and Pascal Molli. 2020. Processing SPARQL Aggregate Queries with Web Preemption. In *17th Extended Semantic Web Conference (ESWC 2020)*. Springer, Heraklion, Greece, 235–251.

[7] Ali Hasnain, Qaiser Mehmood, and Syeda Sana e Zainab ang Aidan Hogan. 2016. SPORTAL: Profiling the Content of Public SPARQL Endpoints. *Int. J. Semantic Web Inf. Syst.* 12, 3 (2016), 134–163.

[8] Pierre Maillot, Olivier Corby, Catherine Faron, Fabien Gandon, and Franck Michel. 2023. IndeGx: A model and a framework for indexing RDF knowledge graphs with SPARQL-based test suits. *J. Web Semant.* 76 (2023), 100775.

[9] Thomas Minier, Hala Skaf-Molli, and Pascal Molli. 2019. SaGe: Web Preemption for Public SPARQL Query Services. In *The World Wide Web Conference 2019 (WWW'19)*. ACM, San Francisco, USA, 1268–1278.

[10] Gabriela Montoya, Hala Skaf-Molli, and Katja Hose. 2017. The Odyssey approach for optimizing federated SPARQL queries. In *International Semantic Web Conference (ISWC)*. Springer-Verlag, Maui, Hawaii, USA, 471–489.

[11] Gabriela Montoya, Hala Skaf-Molli, Pascal Molli, and Maria-Esther Vidal. 2017. Decomposing federated queries in presence of replicated fragments. *Journal of Web Semantics* 42 (2017), 1–18.

[12] Emmanuel Pietriga, Hande Gözükan, Caroline Appert, Marie Destandau, Šejla Čebirić, François Goasdoué, and Ioana Manolescu. 2018. Browsing Linked Data Catalogs with LODAtlas. In *International Semantic Web Conference*. Springer, Springer, Monterey, United States, 137–153.

[13] Bastian Quilitz and Ulf Leser. 2008. Querying Distributed RDF Data Sources with SPARQL. In *Extended Semantic Web Conference (ESWC)*. Springer Berlin Heidelberg, Tenerife, Canary Islands, Spain, 524–538.

[14] Muhammad Saleem, Ali Hasnain, and Axel-Cyrille Ngonga Ngomo. 2018. LargeRDFBench: A billion triples benchmark for SPARQL endpoint federation. *J. Web Semant.* 48 (2018), 85–125.

[15] Muhammad Saleem and Axel-Cyrille Ngonga Ngomo. 2014. HiBISCuS: Hypergraph-based source selection for SPARQL endpoint federation. In *European Semantic Web Conference (ESWC)*. Springer, Cham, 176–191.

[16] Muhammad Saleem, Alexander Potocki, Tommaso Soru, Olaf Hartig, and Axel-Cyrille Ngonga Ngomo. 2018. CostFed: Cost-based query optimization for SPARQL endpoint federation. In *14th International Conference on Semantic Systems (SEMANTICS)*. Elsevier, Amsterdam, The Netherlands, 163–174.

[17] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. 2011. FedX: Optimization techniques for federated query processing on linked data. In *International Semantic Web Conference (ISWC)*. Springer, Bonn, Germany, 601–616.

[18] Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. 2016. Triple Pattern Fragments: A low-cost knowledge graph interface for the Web. *J. Web Sem.* 37-38 (2016), 184–206.