A Framework for Efficient Representative Summarization of RDF Graphs

Šejla Čebirić, François Goasdoué, Ioana Manolescu

A Framework for Efficient Representative Summarization of RDF Graphs

Šejla Čebirić[1] François Goasdoué[2,1]
Ioana Manolescu[1]
[1]INRIA and Ecole Polytechnique, France [2]Univ. Rennes 1, France

August 28, 2017

Abstract

RDF is the data model of choice for Semantic Web applications. RDF graphs are often large and have heterogeneous, complex structure. Graph summaries are compact structures computed from the input graph; they are typically used to simplify users' experience and to speed up graph processing.

We introduce a formal RDF summarization framework, based on graph quotients and RDF node equivalence; our framework can be instantiated with many such equivalence relations. We show that our summaries represent the structure and semantics of the input graph, and establish a sufficient condition on the RDF equivalence relation which ensures that a graph can be summarized more efficiently, without materializing its implicit triples.

1 Introduction

To facilitate working with very large, complex-structure, heterogeneous graphs, many **graph summaries** have been proposed, including some specifically tailored for Resource Description Framework (RDF) graphs [1, 5, 6]. A summary of an RDF graph `G` is a smaller graph (typically also RDF), based on which questions about `G` may be answered more efficiently than by using `G` directly.

In this work, we define a *formal generic summarization framework* for RDF graphs, based on the classical notion of graph quotients, and on our notion of *RDF node equivalence*. While quotient-style summaries have been studied in the past [1, 3], our first contribution is a formal framework for summarizing RDF graphs including possible *RDF Schema constraints*, which leads us to study *the interplay between summarization and saturation* with such constraints. Specifically, our second contribution is *a sufficient condition* on the RDF node equivalence relation, which guarantees that the summary of the saturation of `G` can be built through a *shortcut* procedure, without actually saturating `G`; this can significantly speed up the summary construction.

Our summaries, representative of the complete (saturated) graphs but often much smaller, can be used in GUIs to help users explore and query RDF graphs, or to

RDF statement	Triple	Shorthand
Class assertion	$(s, \text{rdf:type}, o)$	(s, τ, o)
Property assertion	(s, p, o) with $p \neq \text{rdf:type}$	(s, p, o)

RDFS statement	Triple	Shorthand
Subclass	$(s, \text{rdfs:subClassOf}, o)$	$(s, \prec_{sc}, o)$
Subproperty	$(s, \text{rdfs:subPropertyOf}, o)$	$(s, \prec_{sp}, o)$
Domain typing	$(s, \text{rdfs:domain}, o)$	$(s, \hookleftarrow_{d}, o)$
Range typing	$(s, \text{rdfs:range}, o)$	$(s, \hookrightarrow_{r}, o)$

Name	Entailment rule
`rdfs2`	$(p, \hookleftarrow_{d}, o), (b_{s_1}, p, o_1) \rightarrow (b_{s_1}, \tau, o)$
`rdfs3`	$(p, \hookrightarrow_{r}, o), (b_{s_1}, p, o_1) \rightarrow (o_1, \tau, o)$
`rdfs5`	$(p_1, \prec_{sp}, p_2), (p_2, \prec_{sp}, p_3) \rightarrow (p_1, \prec_{sp}, p_3)$
`rdfs7`	$(p_1, \prec_{sp}, p_2), (b_s, p_1, o) \rightarrow (b_s, p_2, o)$
`rdfs9`	$(b_s, \prec_{sc}, o), (b_{s_1}, \tau, b_s) \rightarrow (b_{s_1}, \tau, o)$
`rdfs11`	$(b_s, \prec_{sc}, o), (o, \prec_{sc}, o_1) \rightarrow (b_s, \prec_{sc}, o_1)$
`ext1`	$(p, \hookleftarrow_{d}, o), (o, \prec_{sc}, o_1) \rightarrow (p, \hookleftarrow_{d}, o_1)$
`ext2`	$(p, \hookrightarrow_{r}, o), (o, \prec_{sc}, o_1) \rightarrow (p, \hookrightarrow_{r}, o_1)$
`ext3`	$(p, \prec_{sp}, p_1), (p_1, \hookleftarrow_{d}, o) \rightarrow (p, \hookleftarrow_{d}, o)$
`ext4`	$(p, \prec_{sp}, p_1), (p_1, \hookrightarrow_{r}, o) \rightarrow (p, \hookrightarrow_{r}, o)$

Table 1: RDF & RDFS statements (left) and sample RDF entailment rules (right).

optimize structured and/or keyword queries etc. as has been done in previous works [3, 4, 5, 6].

Due to space constraints, proofs are delegated to our technical report available at [2].

2 Preliminaries

An RDF graph is a set of triples (s, p, o) where s is termed the *subject*, p the *property*, and o the *object*; such a triple states that s is described with the property p that has value o. *Well-formed* triples, as per the RDF specification, belong to $(\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$, where $\mathcal{U}$ is a set of *Universal Resource Identifiers* (URIs in short), $\mathcal{L}$ is a set of *literals* (constants), and $\mathcal{B}$ is a set of *blank nodes*, representing unknown URI or literal values. A triple (s, p, o) states that its subject s has the property p whose value is the object o. RDF allows making *class assertions*, if p is the special built-in RDF property rdf:type (τ in short), and *property assertions* otherwise (Table 1). *RDF Schema* statements (at the bottom left of Table 1, together with the shorthand notations of their properties) allow specifying ontological constraints relating classes and/or properties. The semantics of an RDF graph `G` is its *saturation* (or *closure*) $\mathtt{G}^{\infty}$, defined as the `G` triples together with all the *implicit* triples that can be derived from them and the entailment rules from the RDF standard. Table 1 (right) shows rules that use RDFS constraints to derive implicit facts or implicit constraints. Figure 1 depicts a sample publications graph, where Pub stands for publication ($CPub$ in conferences and $JPub$

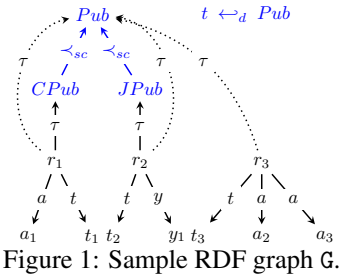

Figure 1: Sample RDF graph `G`.

in journals), a for author, t for title and y for year; class nodes and RDFS triples appear in blue, for instance, the domain of t (title) is Pub. Solid arrows correspond to explicit $\mathtt{G}$ triples, and dotted arrows to *implicit* triples; all together, they depict $\mathtt{G}^{\infty}$.

3 Summarization framework

We start by recalling the classical notion of graph quotient. Let $G = (V, E)$ be a labeled directed graph whose edges E have labels from a set A. Let $\sim$ be an equivalence relation over the graph node set V. The *quotient of G through $\sim$*, denoted $G_{/\sim}$, is a labeled directed graph having (i) a node n_S for each set S of equivalent V nodes, and (ii) an edge $n_{S_1} \xrightarrow{a} n_{S_2}$ for some label $a \in A$ iff there exist two V nodes $n_1 \in S_1$ and $n_2 \in S_2$ such that the edge $n_1 \xrightarrow{a} n_2 \in E$.

When summarizing an RDF graph, class and schema information (e.g., the blue part of Figure 1) should be preserved, as they encode its semantics. Thus, we define:

Definition Let $\equiv$ be a binary relation between the nodes of an RDF graph. We say $\equiv$ is an *RDF equivalence relation* iff (i) $\equiv$ is reflexive, symmetric and transitive, (ii) any class node is $\equiv$ only to itself, and (iii) any property node is $\equiv$ only to itself.

We define an RDF summary as a graph quotient w.r.t. a given RDF node equivalence:

Definition Given an RDF graph $\mathtt{G}$ and an RDF node equivalence relation $\equiv$, the *summary of $\mathtt{G}$ by $\equiv$*, which is an RDF graph denoted $\mathtt{G}_{/\equiv}$, is the quotient of $\mathtt{G}$ by $\equiv$. $\mathtt{G}_{/\equiv}$ data nodes use fresh URIs, one for each set of equivalent $\mathtt{G}$ data nodes.

Different RDF equivalence relations lead to different summaries. Figures 2 illustrates two of them on the saturated $\mathtt{G}^{\infty}$ from Figure 1; circles denote new-URI summary nodes, each of which represents a set of $\mathtt{G}$ nodes. For instance, at left, a single node represents r_1, r_2, r_3; at right, they are separated by their sets of types. Below, we do not discuss any particular summary further; instead, we focus on our summarization framework, and its interplay with saturation.

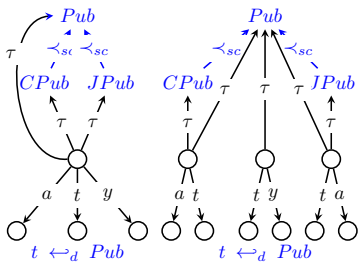

Figure 2: Sample summaries.

For a summary to reflect (*represent*) a graph $\mathtt{G}$, queries having answers on $\mathtt{G}$ should also have answers on the summary. Given an *RDF query language* $\mathcal{Q}$, we define:

Definition Let $\mathtt{G}$ be any RDF graph. $\mathtt{G}_{/\equiv}$ is *$\mathcal{Q}$-representative of* $\mathtt{G}$ if and only if for any query $q \in \mathcal{Q}$ such that $q(\mathtt{G}^{\infty}) \neq \emptyset$, we have $q((\mathtt{G}_{/\equiv})^{\infty}) \neq \emptyset$.

We target summaries representative of any query over the *graph structure* of $\mathtt{G}$, including imprecise queries using variables in some property positions. Thus, we instantiate $\mathcal{Q}$ into Extended Relational Basic Graph Pattern Queries (*RBGP**, in short), a core fragment of SPARQL, defined as follows. A *query triple pattern* belongs to $\mathcal{V} \times (\mathcal{U} \cup \mathcal{V}) \times \mathcal{V}$ or $\mathcal{V} \times \{\tau\} \times \mathcal{U}$, where $\mathcal{V}$ is a set of variables. An RBGP* query q is of the

form: $q(\bar{x}) \leftarrow t_1, \ldots, t_n$ where each t_i is a query triple pattern, $\{t_1, \ldots, t_n\}$ is noted $body(q)$, and $\bar{x}$, called the *answer variables*, is a subset of the variables in $body(q)$. A sample RBGP* query is: $q^*(x_1, x_3)$:- $(x_1, \tau, \texttt{Book}),\ (x_1, \texttt{author}, x_2),\ x_2\ y\ x_3$.

We show (the proof is in [2]) that for any RDF equivalence relation $\equiv$:

Proposition 3.1 *An RDF summary $\texttt{G}_{/\equiv}$ is RBGP*-representative.*

RBGP* representativeness ensures that any query specifying a certain graph pattern in G and/or querying the structure itself (by means of variables in property positions, such as y in the sample query above) which has answers on G, also has answers on $\texttt{G}_{/\equiv}$.

In the presence of an RDF Schema, the semantics of G is its saturation $\texttt{G}^\infty$. Thus, a representative summary must reflect both the explicit and the implicit triples of G. For instance, the summaries in Figure 2 show that some $\texttt{G}^\infty$ resources (e.g., r_1, r_2, r_3) are of type Pub, but the same summaries computed from G alone do not, as the corresponding τ triples are implicit in G. A simple way to obtain $(\texttt{G}^\infty)_{/\equiv}$ is to compute $\texttt{G}^\infty$ and then summarize it. We define a novel alternative *shortcut* method, which avoids saturating G, yet it constructs an RDF graph *strongly isomorphic* to $(\texttt{G}^\infty)_{/\equiv}$, as follows:

Definition A *strong isomorphism between two RDF graphs* $\texttt{G}_1, \texttt{G}_2$, noted $\texttt{G}_1 \simeq \texttt{G}_2$, is an isomorphism which is the identity for the class and property nodes.

Definition Summarization through the RDF node equivalence relation $\equiv$ *admits a shortcut* iff for any RDF graph G, $(\texttt{G}^\infty)_{/\equiv} \simeq ((\texttt{G}_{/\equiv})^\infty)_{/\equiv}$ holds.

The shortcut summarizes G, saturates the result, then summarize it again (the three green edges in Figure 3). Its result is *essentially* $(\texttt{G}^\infty)_{/\equiv}$, as the two have identical graph structures (guaranteed by the strong isomorphism), on which RBGP* representativeness is defined. They only differ in the new URIs of their nodes (circles in Figure 2).

What is the interest of the shortcut? If $\texttt{G}_{/\equiv}$ is much smaller than G, it is much faster to saturate $\texttt{G}_{/\equiv}$ (on the shortcut) than to saturate G; $(\texttt{G}_{/\equiv})^\infty$ is also likely to be small, thus fast to summarize. Further, summarizing $\texttt{G}^\infty$ is faster than summarizing G, given that $\texttt{G}^\infty$ is at least as large as G. Summing up these inequalities, *the time spent on the shortcut may be (much) shorter than the time spent to build $(\texttt{G}^\infty)_{/\equiv}$ directly*.

By the summary definition, to every node in G corresponds exactly one node in the summary $\texttt{G}_{\equiv}$. We call **representation function** and denote $f_\equiv$ the function associating a summary node to each G node; we say $f_\equiv(n)$ *represents* n in the summary. An important structural property relates G, $\texttt{G}^\infty$ and the function $f_\equiv$ (see Figure 3):

Lemma 3.2 (Summarization Homomorphism) *Let G be an RDF graph, $\texttt{G}_{/\equiv}$ its summary and $f_\equiv$ the corresponding representation function from G nodes to $\texttt{G}_{/\equiv}$ nodes. $f_\equiv$ defines a homomorphism from $\texttt{G}^\infty$ to $(\texttt{G}_{/\equiv})^\infty$.*

Based on the Lemma, we establish the sufficient condition [2] (see Figure 3):

Theorem 3.3 (Sufficient condition for shortcuts) *Given an RDF node equivalence relation $\equiv$, and an RDF graph G, let $\texttt{G}_{/\equiv}$ be its summary and $f_\equiv$ the corresponding representation function from G nodes to $\texttt{G}_{/\equiv}$ nodes.*

If $\equiv$ satisfies: for any RDF graph G and any pair (n_1, n_2) of G nodes, $n_1 \equiv n_2$ in $\texttt{G}^\infty$ iff $f_\equiv(n_1) \equiv f_\equiv(n_2)$ in $(\texttt{G}_{/\equiv})^\infty$, then $(\texttt{G}^\infty)_{/\equiv} \simeq ((\texttt{G}_{/\equiv})^\infty)_{/\equiv}$ holds.

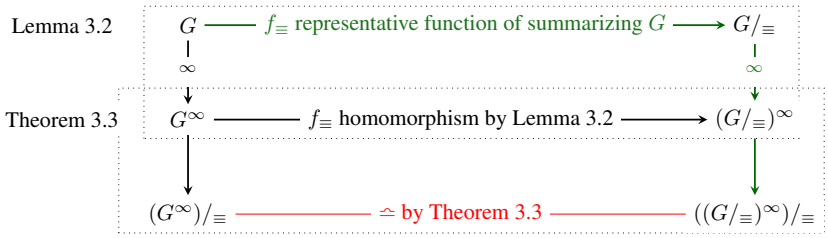

Figure 3: Illustration for Lemma 3.2 and Theorem 3.3.

The summary illustrated at left Figure 2 turns out to admit the shortcut; in our experiments, the shortcut was up to **20x faster** than saturating G and then summarizing $\mathtt{G}^{\infty}$. The summary illustrated at right in Figure 2 does not admit the shortcut.

Conclusion and perspectives Finding a necessary (and sufficient) condition for the shortcut is currently open. We have instantiated our framework into many summaries, and implemented a summarization tool available online (together with many sample summaries) at `https://team.inria.fr/cedar/projects/rdfsummary`. We are currently working on summary-based query pruning, where we decide based on $(\mathtt{G}^{\infty})_{/\equiv}$ whether a query may have answers on $\mathtt{G}^{\infty}$ or not.

References

[1] Stéphane Campinas, Renaud Delbru, and Giovanni Tummarello. Efficiency and precision trade-offs in graph summary algorithms. In *IDEAS*, 2013.

[2] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. A framework for efficient representative summarization of RDF graphs. Inria Research Report no. 9090, available at `https://hal.inria.fr/hal-01577431`, 2017.

[3] Qun Chen, Andrew Lim, and Kian Win Ong. $D(K)$-index: An adaptive structural summary for graph-structured data. In *SIGMOD*, 2003.

[4] Sairam Gurajada, Stephan Seufert, Iris Miliaraki, and Martin Theobald. Using graph summarization for join-ahead pruning in a distributed RDF engine. In *SWIM*, 2014.

[5] Thanh Tran, Günter Ladwig, and Sebastian Rudolph. Managing structured and semistructured RDF data using structure indexes. *IEEE TKDE*, 25(9), 2013.

[6] Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. RDF digest: Efficient summarization of RDF/S KBs. In *ESWC*, 2015.