Incremental structural summarization of RDF graphs
François Goasdoué, Pawel Guzewicz, Ioana Manolescu

To cite this version:
François Goasdoué, Pawel Guzewicz, Ioana Manolescu. Incremental structural summarization of RDF graphs. EDBT 2019 - 22nd International Conference on Extending Database Technology, Mar 2019, Lisbon, Portugal. hal-01978784

HAL Id: hal-01978784
https://inria.hal.science/hal-01978784
Submitted on 11 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Incremental structural summarization of RDF graphs

François Goasdoué  
Univ Rennes, Inria, CNRS, IRISA  
France  
fg@irisa.fr

Paweł Guzewicz  
École polytechnique and Inria  
France  
pawel.guzewicz@inria.fr

Ioana Manolescu  
Inria and École polytechnique  
France  
ioana.manolescu@inria.fr

ABSTRACT

Realizing the full potential of Linked Open Data sharing and reuse is currently limited by the difficulty users have when trying to understand the data modeled within an RDF graph, in order to determine whether or not it may be useful for their need.

We demonstrate our RDFQuotient tool, which builds compact summaries of heterogeneous RDF graphs for the purpose of first-sight visualizations. An RDFQuotient summary provides an overview of the complete structure of an RDF graph, while being typically many orders of magnitude smaller, thus can be easily grasped by new users. Our summarization algorithms are time linear in the size of the input graph and incremental: they incrementally update a summary upon addition of new data.

For the demo, we plan to show the visualizations of our summaries obtained from well-known synthetic and real data sets. Further, attendees will be able to add data to the summarized RDF graphs and visually witness the incurred changes.

1 INTRODUCTION

Semantic Web graphs are nowadays being published and shared at a massive scale, e.g., Linked Open Data (LOD) Cloud (https://lod-cloud.net) lists 1,200 graphs, while the LOD Atlas portal (http://lodatlas.lri.fr) references more than 20,000 graphs. Some licenses CC-by-nc-nd 4.0.

Distribution of this paper is permitted under the terms of the Creative Commons © 2019 Copyright held by the owner/author(s). Published in Proceedings of the 22nd International Conference on Extending Database Technology (EDBT), March 26-29, 2019, ISBN 978-3-89318-081-3 on OpenProceedings.org.

RDFQuotient summaries can be built efficiently, in linear time in the size of the graph. Further, they can be incrementally updated: upon addition of a triple \( t \) to a graph \( G \), the summary of \( G \cup \{ t \} \) can be efficiently computed out of the current summary of \( G \) and \( t \), without re-traversing the \( G \) triples.

Motivating example Figure 1 illustrates a possible visualization of an RDFQuotient summary of a BSBM [1] benchmark graph of \( 10^6 \) triples. This visualization reflects the complete structure of the graph, using only 5 nodes and 11 edges, comparable to a simple small Entity-Relationship diagram. This summary reads as follows: (i) Non-leaf graph nodes belong to one of five disjoint classes, each represented by a summary node (boxes labeled N1 to N5 in Figure 1). The number of graph nodes in each class appears in parenthesis after the label N1 of their representative; (ii) Graph nodes from a class may have types. Each such type appears under the summary node label, together with its number of occurrences among graph nodes of that class, e.g., 5919 nodes represented by N3 are of type Producer, while 3050 are of type Vendor; (iii) Graph nodes from a class may have outgoing properties whose values are leaf nodes in the graph; the set of all such properties appears in the corresponding summary node, one property per line. For each property, e.g. country for N3, the summary node specifies how many graph nodes represented by this summary node have it (8969 in this case), and how many distinct leaf nodes are target of these edges (10 in this case); (iv) Graph nodes from a class may have outgoing properties whose values are non-leaf nodes in the graph. For each graph edge \( n_1 \rightarrow a \rightarrow n_2 \), where \( n_1, n_2 \) are non-leaf graph nodes and \( a \) is the property (edge label), an \( a \)-labeled edge in the summary goes from the representative of \( n_1 \) to that of \( n_2 \). Next to \( a \), this summary edge is also labeled with the number of graph edges to which it corresponds; (v) Properties from a small, fixed vocabulary are considered metadata (as opposed to data) and therefore they are not used to group nodes in classes, e.g., rdf-schema#-comment and rdf-schema#-label in Figure 1. More such visualizations can be found online\(^1\); below, we also work out an example leading from an RDF graph to its summary and then such a visualization.

We propose to demonstrate the incremental construction of four related (but different) summaries, i.e., show how summaries quickly adjust when triples are added to the summarized RDF graphs. Our summaries can be built from graphs where none, some or all nodes have one or more types; this is important because in many synthetic and real-life RDF graphs we studied, a large share of nodes is untyped [3]. Two of our summaries give preeminence to types (when available) to build the summary; nodes are first grouped by types and then by the relationships to other nodes. By contrast, the other two (including the one in Figure 1) give preeminence to node relationships; nodes are first grouped according to their relationships with others, then, each group is typed with the types of the graph nodes it represents within the summary (this is how each type has been attached to a summary node in our figure). In the total absence of types, each

\(^1\)https://team.inria.fr/cedar/projects/rdfquotient/
\(o \in n\) whose property is not an RDF graph having (transitive). Given an equivalence relation over the nodes of an RDF graph that is reflexive, symmetric and consists exclusively of type \(G\) majorities of all RDF graphs we encountered. Thus, we consider on summarizing the non-schema triples, which make up the vast between summarization and ontologies [3], below we only focus on summarizing and ontologies. Below, we define our summaries (Section 2) and summarization algorithms (Section 3). Then, we present the demonstration scenario based on summary visualizations (Section 5). Finally, we compare them with related work and we conclude (Section 6).

2 RDFQUOTIENT SUMMARIES

Let \(U\) be a set of URIs, \(L\) be a set of literals and \(B\) be a set of blank nodes as per the RDF specification. An RDF graph \(G\) is a set of triples of the form \((s, p, o)\) where \(s \in U \cup B \cup L\), \(p \in U\) and \(o \in U \cup B \cup L\). The special URI type, part of the RDF standard, is used to attach types to nodes. An RDF graph may contain ontology (schema) triples; while there are interesting interactions between summarization and ontologies [3], below we only focus on summarizing the non-schema triples, which make up the vast majority of all RDF graphs we encountered. Thus, we consider \(G\) consists exclusively of type triples and/or data triples (all those whose property is not type; we call these data properties).

An RDF equivalence relation denoted \(\equiv\) is a binary relation over the nodes of an RDF graph that is reflexive, symmetric and transitive. Given an equivalence relation \(\equiv\), an RDF graph quotient is an RDF graph having \((i)\) one node for each equivalence class of nodes; \((ii)\) for each edge \(n_1 \xrightarrow{a} n_2\), a summary edge \(n_1^2 \xrightarrow{a} n_2^2\), where \(n_i^2, i \in \{1, 2\}\), is the summary node corresponding to the equivalence class of \(n_i\), also called representative of \(n_i\). The literature comprises many quotient graph summaries (see Section 6), which differ by their equivalence relations.

The equivalence relations we use are based on the concept of property cliques, which encodes a transitive relation of edge co-occurrence on graph nodes. Given an RDF graph \(G\), two data properties \(p_1, p_2\) are in the same source clique if: \((i)\) there exists a \(G\) node \(n\) which is the source of \(p_1\) and \(p_2\) (i.e., \((n, p_1, x) \in G\) and \((n, p_2, y) \in G\) for some \(x\) and \(y\)), or \((ii)\) there exists a data property \(p_3\) such that \(p_3\) is in the same source clique as \(p_1\) and \(p_2\) in the same source clique as \(p_3\). Symetrically, \(p_1\) and \(p_2\) are in the same target clique if there exists a \(G\) node which is the target of \(p_1\) and \(p_2\), or a data property \(p_3\) which is in the same target clique as \(p_1\) and \(p_2\). In Figure 2, the properties \(advises\) and teaches are in the same source clique due to \(p_3\). The same holds for \(advises\) and wrote due to \(p_1\); consequently, \(advises\) and wrote are also in the same source clique. Further, the graduate student \(p_2\) teaches a course and takes another, thus teaches, advises, wrote and takes are all part of the same source clique. In this example, \(p_1, p_2, p_3, p_4, p_5\) have the source clique \(SC_1 = \{\text{advises}, \text{taught}, \text{wrote}\}\) and \(a_1, a_2\) have the empty source clique \(SC_3 = \emptyset\). Similarly, the target cliques are, respectively; \(TC_1 = \{\text{advises}\}\) for \(p_2, p_5\), \(TC_2 = \{\text{teaches}, \text{takes}\}\) for \(c_1, c_2, c_3\), \(TC_3 = \{\text{coursedescr}\}\) for \(d_1, d_2\), \(TC_4 = \{\text{wrote}\}\) for \(a_1, a_2\) and \(TC_5 = \emptyset\) for \(p_1, p_3, p_4\).

It is easy to see that the set of non-empty source (or target) cliques is a partition over the data properties of an RDF graph \(G\). Further, if a \(G\) node \(n\) is source of some data properties, they are all in the same source clique; similarly, all the properties of which \(n\) is a target are in the same target clique. Based on these cliques, for any nodes \(n_1, n_2\) of \(G\), we define:

- \(n_1\) is weakly equivalent to \(n_2\), denoted \(n_1 \equiv_w n_2\), if \(n_1, n_2\) have the same source clique or the same target clique.
Further, we decide that in any RDF equivalence relation, any
class node, i.e., a URI c appearing in a triple of the form (n, type, c),
is (i) only equivalent to itself and (ii) represented by itself in any
RDFQuotient summary. This ensures that RDF types (classes),
which (when present) denote an important information that data
producers added to help understand their RDF graphs, are pre-
served in the summary.

The equivalence relations $\equiv_\text{w}$ and $\equiv_\text{g}$ lead to the weak, respectiv-
ely strong summaries, defined as quotients of $G$ through $\equiv_\text{w},$
denoted $G/\equiv_\text{w}$, respectively, through $\equiv_\text{g},$ denoted $G/\equiv_\text{g}$. Figure 3
illustrates these on the sample graph in Figure 2. For brevity,
in Figure 3 and from now on, we use $a, w, t, e, t$, $\equiv_\text{w}$ to denote
respectively the properties advises, writes, teaches, takes, and
course_description.

In $G/\equiv_\text{w}$, $N_1$ represents all the people ($p_1$ to $p_5$), $N_2$ represents
the courses, $N_3$ the articles and $N_4$ the course descriptions. Note
the self-loop from $N_1$ to itself; it denotes that some nodes repre-
sented by $N_1$ advise some nodes represented by $N_1$. This summary
has only 4 nodes and 5 edges. It conveys the essential informa-
tion that some nodes advise, write, also they teach and take
something that has course descriptions. The Professor and Grad-
Student types of nodes $p_1$, respectively $p_2$, are attached to their
common representative $N_1$.

$G/\equiv_\text{g}$ differs from $G/\equiv_\text{w}$ by representing the person nodes in
two separate groups: those represented by $N_1$ advise those rep-
resented by $N_2$. This is because the target clique of $p_1$, $p_3$ and $p_4$
are empty, while the target clique of $p_2$ and $p_5$ is \{advises\}. This
example illustrates the fact (visible from the summary defini-
tions) that $G/\equiv_\text{g}$ summarizes at finer granularity than $G/\equiv_\text{w}$ (or,
equivalently, $\equiv_\text{w}$ entails $\equiv_\text{g}$, but the opposite does not hold).

Clique-based structural summarization leads to compact
summaries even in graphs with heterogeneous structure.
This is because of the transitive aspect of the property cliques.
For example, $p_1$ and $p_3$ have the same source clique, even though
their property sets are disjoint: \{a, w\} for $p_1$, \{te\} for $p_3$; they are in
the same source clique e.g. due to $p_4$, which has both $a$ and $te$.
In contrast, previously studied quotient summaries, in particular
those aimed for indexing and query optimization, would not ac-
ccept $p_1$ and $p_3$ as equivalent; in general, such summaries lead to
more equivalence classes (summary nodes), thus also summary
edges, making summaries hard to understand visually.

Type-first summarization The weak and strong summary group
nodes according to their incoming/outgoing data triples and then
just “carry” their types to the summary. A different choice is to
group nodes first by their set of types (if any)\(^2\), and use the
data triples to group the nodes without types. We define:

- $n_1$ is typed-weak equivalent to $n_2$, noted $n_1 \equiv_{\text{tw}} n_2$, iff
  \( i) n_1, n_2 \) have the same non-empty set of types or \( ii) both
  n_1, n_2 \) are untyped, and $n_1 \equiv n_2$;

- $n_1$ is typed-strong equivalent to $n_2$, noted $n_1 \equiv_{\text{ts}} n_2$, iff
  \( i) n_1, n_2 \) have the same non-empty set of types or \( ii) both
  n_1, n_2 \) are untyped, and $n_1 \equiv n_2$.

These relations lead to the typed weak ($G_{\equiv_{\text{tw}}}$), respectiv-
ely, to the typed strong ($G_{\equiv_{\text{ts}}}$) RDFQuotient summaries. Figure 4
illustrates typed weak summarization on our sample graph; on
this simple example, $G_{\equiv_{\text{tw}}}$ is identical (in general, it may differ).

3 BUILDING RDFQUOTIENT SUMMARIES

We have devised algorithms which build $G_{\equiv_{\text{tw}}}$, $G_{\equiv_{\text{ts}}}$, $G_{\equiv_{\text{tG}}}$ and
$G_{\equiv_{\text{tG}}}^\prime$ through a single traversal of an RDF graph G, in two flav-
s of (i) global: traverse G, compute all the cliques, then traverse
it again and represent nodes according to their cliques and/or
types; (ii) incremental: in a single traversal of G, gradually build
each source and target clique based on the triples traversed up to
that point and simultaneously represent G nodes in a summary
that is continuously updated; after traversing the last triple of
G, each incremental summarization algorithm ends up with the
respective summary of the full G. The algorithms are detailed
and their correctness is proved in [3]; below, we illustrate their
interesting points on minimal examples.

First, let us see on an example how $\equiv_\text{w}$ can grow during incre-
mental weak summarization. Suppose the graph $G$ in Figure 2
is traversed and summarized starting with: ($p_1$ advises $p_2$), then
($p_1$ wrote $a_1$), then ($p_1$ teaches $c_2$) (see the figure above). When
we summarize this third triple, we do not know yet that $p_1$
is equivalent to $p_4$, because no common source of teaches and
advises (e.g., $p_3$ or $p_4$) has been seen so far. Thus, $p_4$ is so far not
equivalent to any other node, and represented separately from
$p_1$. Now, assume the fourth triple traversed is ($p_3$ advises $p_5$):
at this point, we know that advises, wrote and teaches are in
the same source clique, thus $p_1 \equiv p_4$, and their representatives
(highlighted in yellow) must be fused in the summary. More
generally, it can be shown that $\equiv_\text{w}$ only grows as more triples are
visited (i.e., is monotonic), in other words: if in a subset $G'$ of
$G$'s triples, two nodes $n_1, n_2$ are weakly equivalent, then this
holds in any $G''$ with $G' \subseteq G'' \subseteq G$.

Incremental strong summarization is even more complex be-
cause unlike $\equiv_\text{w}$, $\equiv_\text{g}$ may grow and shrink during summariza-
tion (i.e., is non-monotonic). For instance, assume the summa-
ration of the graph in Figure 2 starts with ($p_1$ wrote $a_1$), ($p_2$
writes $a_2$), ($p_2$ takes $c_2$) (see the figure below). After these, we
know $p_1 \equiv p_2$; their source clique is \{wrote, takes\} and their tar-
target clique is $\emptyset$. Assume the next triple traversed is ($p_3$
advises $p_2$): at this point, $p_3$ is not $\equiv_\text{g}$ to $p_2$ any more, because $p_2$'s target clique
is now \{advises\} instead of $\emptyset$. Thus, $p_3$ splits from $p_1$, that is,
it needs to be represented by a new summary node (shown in
yellow below), distinct from the representative of $p_1$.
The core of our work is on defining and efficiently building summaries from Figure 3 (right).

4 FROM SUMMARIES TO VISUALIZATIONS

The literature comprises many RDF summarization techniques, more than a hundred of which we covered in a recent co-authored survey [4]. RDFQuotient summarizes the structure of the data triples, which form a vast majority of RDF graphs; complementary proposals summarize the ontology, the values, find the most frequent property groups etc. Closest to us are quotient summaries which group nodes by the set of their outgoing data properties ("characteristic sets" [6]) and possibly also by by the set of their incoming data properties (forward and backward simulation [5]). Our clique-based summarization differs from these by the transitive aspect of the cliques which leads to heterogeneity-tolerant summarization. Indeed, as we plan to show during our demonstrations, more strict summaries such as [5, 6] support query optimization and indexing well, but have too many nodes and edges, even after inlining, for a comfortable visualization.

In exchange, our summaries are not generally appropriate for indexing, as they do not give e.g. access to "all the resources having properties a and b": the graph nodes whose source clique is \(\{a, b\}\) may have one or another or both.

REFERENCES