Data Discovery in RDF Graphs
Ioana Manolescu

To cite this version:
Ioana Manolescu. Data Discovery in RDF Graphs. DEXA 2017 - 28th International Conference on Database and Expert System Applications, Aug 2017, Lyon, France. pp.1-63. hal-01657144

HAL Id: hal-01657144
https://inria.hal.science/hal-01657144
Submitted on 6 Dec 2017

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.
Data Discovery in RDF Graphs

Ioana Manolescu

INRIA and Ecole Polytechnique, France
ioana.manolescu@inria.fr
http://pages.saclay.inria.fr/Ioana.Manolescu

DEXA Conference, Aug 29, 2017
1 Background: **semantic** RDF graphs

2 **Summarizing** semantic-rich RDF graphs
   [ČGM15a, ČGM15b, ČGM17a]
   - Joint work with Šejla Čebirić (Inria) and François Goasdoué (U. Rennes 1 and Inria)

3 Finding **insights** in RDF graphs [DMS17]
   - Joint work with Yanlei Diao and Shu Shang (Ecole Polytechnique and Inria)
Part I

Background: RDF graphs
Big Data needs semantics

AI Magazine, Spring 2015
Do we really need the semantics?

Yes. All the time.

Application knowledge / constraints:
- Every Senator is an ElectedOfficial which is a Person
- (On Wikipedia) being BornInAPlace means being a Person
Do we really need the semantics?

**Yes. All the time.**

**Application knowledge / constraints:**
- Every Senator is an ElectedOfficial which is a Person
- (On Wikipedia) being BornInAPlace means being a Person

**Without the semantics, we may miss query answers**

<table>
<thead>
<tr>
<th>Data</th>
<th>Constraints</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>John is a Senator</td>
<td>Every Senator is a Person</td>
<td>Who is a Person?</td>
</tr>
</tbody>
</table>
Do we really need the semantics?

Yes. All the time.

Application knowledge / constraints:
- Every Senator is an ElectedOfficial which is a Person
- (On Wikipedia) being BornInAPlace means being a Person

Semantic contraints are a compact way of encoding information

“Every ElectedOfficial is a Person” stated only once even if thousands of ElectedOfficials.
Semantics for Web data

Data and metadata on the Web is often structured in graphs, e.g., **RDF** (W3C’s Resource Description Framework)

- Famous application: the Linked Open Data cloud (2017)
RDF graph: set of triples

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Triple</th>
<th>Relational notation</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>s rdf:type o</td>
<td>o(s)</td>
<td>”s is an o”</td>
</tr>
<tr>
<td>Property</td>
<td>s p o</td>
<td>p(s, o)</td>
<td>”The p of s is o”</td>
</tr>
</tbody>
</table>
**RDF graph**: set of **triples**

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Triple</th>
<th>Relational notation</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>s rdf:type o</td>
<td>o(s)</td>
<td>”s is an o”</td>
</tr>
<tr>
<td>Property</td>
<td>s p o</td>
<td>p(s, o)</td>
<td>”The p of s is o”</td>
</tr>
</tbody>
</table>

```
Class
resource (URI)
blank node
“literal (string)”

property →
```

```
Class
property

“1949”
publishedIn

Book

rdf:type

doi1

hasTitle

“El Aleph”

writtenBy

_:b1

hasName

“J. L. Borges”
```
## The Resource Description Framework (RDF)

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Triple</th>
<th>Relational notation</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Property</td>
<td>s rdf:type o</td>
<td>o(s)</td>
<td>&quot;s is an o&quot;</td>
</tr>
<tr>
<td></td>
<td>s p o</td>
<td>p(s, o)</td>
<td>&quot;The p of s is o&quot;</td>
</tr>
</tbody>
</table>

- **RDF**
- **Assertion**
- **Triple**
- **Relational notation**
- **Intuition**
## The Resource Description Framework (RDF)

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Triple</th>
<th>Relational notation</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>s rdf:type o</td>
<td>o(s)</td>
<td>&quot;s is an o&quot;</td>
</tr>
<tr>
<td>Property</td>
<td>s p o</td>
<td>p(s, o)</td>
<td>&quot;The p of s is o&quot;</td>
</tr>
</tbody>
</table>

- **Class**
  - resource (URI)
  - blank node
  - "literal (string)"
  - property

- **Property**
  - rdf:property

- **Book**
  - rdf:type
  - writtenBy
  - hasName
  - hasTitle
  - doi

- **"1949"**
  - publishedIn

- **"El Aleph"**

- **"J. L. Borges"**
RDF Schema (RDFS)

Declare **deductive constraints** between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>c₁ rdfs:subClassOf c₂</td>
<td>c₁ ⊆ c₂</td>
</tr>
<tr>
<td>Subproperty</td>
<td>p₁ rdfs:subPropertyOf p₂</td>
<td>p₁ ⊆ p₂</td>
</tr>
<tr>
<td>Domain typing</td>
<td>p rdfs:domain c</td>
<td>(\Pi_{\text{domain}}(p) \subseteq c)</td>
</tr>
<tr>
<td>Range typing</td>
<td>p rdfs:range c</td>
<td>(\Pi_{\text{range}}(p) \subseteq c)</td>
</tr>
</tbody>
</table>

“Any c₁ is also a c₂”
RDF Schema (RDFS)

Declare deductive constraints between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>$c_1 \text{ rdfs:subClassOf } c_2$</td>
<td>$c_1 \subseteq c_2$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$p_1 \text{ rdfs:subPropertyOf } p_2$</td>
<td>$p_1 \subseteq p_2$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$p \text{ rdfs:domain } c$</td>
<td>$\Pi_{\text{domain}}(p) \subseteq c$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$p \text{ rdfs:range } c$</td>
<td>$\Pi_{\text{range}}(p) \subseteq c$</td>
</tr>
</tbody>
</table>

If two resources are related by $p_1$, they are also related by $p_2$
### RDF Schema (RDFS)

Declare deductive constraints between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWA interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>$c_1 \text{ rdfs:subClassOf } c_2$</td>
<td>$c_1 \subseteq c_2$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$p_1 \text{ rdfs:subPropertyOf } p_2$</td>
<td>$p_1 \subseteq p_2$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$p \text{ rdfs:domain } c$</td>
<td>$\Pi_{\text{domain}}(p) \subseteq c$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$p \text{ rdfs:range } c$</td>
<td>$\Pi_{\text{range}}(p) \subseteq c$</td>
</tr>
</tbody>
</table>

“Anyone having $p$ is a $c$”
RDF Schema (RDFS)

Declare deductive constraints between classes and properties

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Triple</th>
<th>OWL interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>$c_1 \text{ rdfs:subClassOf } c_2$</td>
<td>$c_1 \subseteq c_2$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$p_1 \text{ rdfs:subPropertyOf } p_2$</td>
<td>$p_1 \subseteq p_2$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$p \text{ rdfs:domain } c$</td>
<td>$\Pi_{\text{domain}}(p) \subseteq c$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$p \text{ rdfs:range } c$</td>
<td>$\Pi_{\text{range}}(p) \subseteq c$</td>
</tr>
</tbody>
</table>

“Anyone who is a value of $p$ is a $c$”
Open-world assumption and RDF entailment

**RDF data model** based on the **open-world assumption**.

Deductive constraints lead to **implicit triples**: part of the graph even though not explicitly present.
Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

Deductive constraints lead to implicit triples: part of the graph even though not explicitly present.

\[
\text{explicit triples} + \text{entailment rules} \rightarrow \text{implicit triples}
\]
Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

Deductive constraints lead to implicit triples: part of the graph even though not explicitly present.

\[
\begin{align*}
\text{explicit triples} & \quad + \quad \text{entailment rules} \\
\rightarrow & \quad \text{implicit triples}
\end{align*}
\]

Exhaustive application of entailment leads to saturation (closure).
The semantics of an RDF graph $G$ is its saturation $G^{\infty}$

**Sample instance entailment rules from schema and instance triples**

\[
\begin{align*}
c_1 & \text{ rdfs:subClassOf } c_2 \land s \text{ rdf:type } c_1 \models_{\text{RDF}} s \text{ rdf:type } c_2 \\
p_1 & \text{ rdfs:subPropertyOf } p_2 \land s \text{ p}_1 \circ \models_{\text{RDF}} s \text{ p}_2 \circ \\
p & \text{ rdfs:domain } c \land s \text{ p } o \models_{\text{RDF}} s \text{ rdf:type } c \\
p & \text{ rdfs:range } c \land s \text{ p } o \models_{\text{RDF}} o \text{ rdf:type } c
\end{align*}
\]
RDF graph discovery

An RDF graph can be large and complex, lack a fixed schema, include many heterogeneous values...
RDF graph discovery

An RDF graph can be large and complex, lack a fixed schema, include many heterogeneous values...
RDF graph discovery

Two approaches:

1. RDF summarization: compactly representing the explicit and implicit structure of a graph
2. Insight discovery in RDF graphs: automatically identify aggregation queries with interesting results
Part II

RDF summarization
RDF summaries

Problem

RDF graph $G$ is large, heterogeneous, partially implicit. How to compactly represent all its structure?

Existing solutions

Partial representation (frequent patterns, statistics etc.) e.g., [NM11, LYL13]

Potentially not compact e.g., [GW97, CFKP15]

Only for explicit data, e.g., [CDT13, ZDYZ14]
A summary of DBLP data

150M triples
A summary of geographic data

French territory division in regions, departments, urban areas, cities, districts etc.

368K triples
RDF summaries

We define

1. **RDF node equivalence relation** $\equiv$: equivalence relation such that class and property nodes are only equivalent to themselves

2. **RDF summary** $G/\equiv$ of an RDF graph $G$: the quotient of $G$ through $\equiv$

Recall: quotient of a directed graph $G$ by $\equiv$

$G = (V, E)$, $\equiv$ equivalence relation on $V$

- $G/\equiv$ nodes: one for $\equiv$ equivalence class of $V$
- $G/\equiv$ edges: $n_1^1 \xrightarrow{a} n_2^2$ iff $\exists n_1 \xrightarrow{a} n_2 \in G$ such that $n_1$ represented by $n_1^1/\equiv$, $n_2$ represented by $n_2^2/\equiv$
Why do we need a special RDF equivalence?

Why not use any node equivalence? E.g., forward and backward bisimilarity $\sim_{f,b}$ [HHK95]
Why do we need a special RDF equivalence?

Why not use any node equivalence? E.g., forward and backward bisimilarity $\sim_{fb}$ [HHK95]

Sample graph $G$ and its quotient through $\sim_{fb}$

$p_1 \sim_{sp} p_2$

$p_3 \sim_{sp} p_4$

$G$ and its quotient through $\sim_{fb}$
Why do we need a special RDF equivalence?

Why not use any node equivalence? E.g., forward and backward bisimilarity $\sim_{fb}$ [HHK95]

Sample graph $G$ and its quotient through $\sim_{fb}$

$p_1 \sim_{sp} p_2$

$p_3 \sim_{sp} p_4$

Loss of class and (some) property names
Why do we need a special RDF equivalence?

Why not use any graph node equivalence? E.g., forward and backward bisimilarity \( \sim_{fb} \)

Sample graph \( G \) and its quotient through \( \sim_{fb} \)

\[
\begin{align*}
  p_1 \sim_{sp} & \rightarrow p_2 \\
  p_3 \sim_{sp} & \rightarrow p_4 \\
  u_1 \sim p_1 & \rightarrow u_2 \\
  u_2 \sim p_2 & \\
  u_3 \sim p_1 & \rightarrow u_4 \\
  u_4 \sim p_2 & \\
  u_5 \sim p_3 & \rightarrow u_6 \\
  u_6 \sim p_4 &
\end{align*}
\]

Loss of schema triples
Why do we need a special RDF equivalence?

Why not use any graph node equivalence? E.g., forward and backward bisimilarity $\sim_{fb}$

Sample graph $G$ and its quotient through $\sim_{fb}$

$p_1 \sim_{sp} p_2$

$p_3 \sim_{sp} p_4$

$\hspace{1cm} A \uparrow \tau \hspace{1cm} B \uparrow \tau$

$u_1 \sim_{p_1} u_2$ \hspace{1cm} $u_3 \sim_{p_1} u_4$

$p_2$ \hspace{1cm} $p_2$

$u_5 \sim_{p_3} u_6$ \hspace{1cm} $u_5 \sim_{p_3} u_6$

$p_4$ \hspace{1cm} $p_4$

$\hspace{1cm}$

$\hspace{1cm}$

$\hspace{1cm} \sim_{sp} \rightarrow \bullet$ \hspace{1cm} $\sim_{sp} \rightarrow \bullet$

$\hspace{1cm}$

$\hspace{1cm}$

$\hspace{1cm}$
Why do we need a special RDF equivalence?

Why not use any graph node equivalence? E.g., forward and backward bisimilarity $\sim_{fb}$

Sample graph $G$ and its quotient through $\sim_{fb}$

\[
\begin{align*}
p_1 & \sim_{sp} p_2 \\
p_3 & \sim_{sp} p_4
\end{align*}
\]

Loss of implicit triples
Why do we need a special RDF equivalence?

**Sample graph $G$ and its quotient through $\sim_{fb}$**

$p_1 - \prec_{sp} \rightarrow p_2$

$A$

$\uparrow$

$\tau$

$u_1 - p_1 \rightarrow u_2$

$p_2$

$u_5 - p_3 \rightarrow u_6$

$p_4$

$B$

$\uparrow$

$\tau$

$u_3 - p_1 \rightarrow u_4$

$p_2$

$A \uparrow_{fb} B$

$\uparrow$

$p_1 \uparrow_{fb} p_2$

$p_3 \uparrow_{fb} p_4$

Quotient of the same graph through the RDF node equivalence $\equiv_{fb}$

$p_1 - \prec_{sp} \rightarrow p_2$

$A$

$\uparrow$

$\tau$

$p_1 \rightarrow p_2$

$p_3 - \prec_{sp} \rightarrow p_4$

$B$

$\uparrow$

$p_3 \rightarrow p_4$
## Formal summary properties

For any RDF equivalence relation $\equiv$:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size limit</strong></td>
<td>The summary is at most as large as the graph.</td>
</tr>
<tr>
<td><strong>Schema preservation</strong></td>
<td>The schema of $G/\equiv$ is the schema of $G$.</td>
</tr>
<tr>
<td><strong>Representativeness</strong></td>
<td>Any conjunctive query $q$ with answers on $G$ also has answers on its summary: $q(G^\infty) \neq \emptyset \Rightarrow q((G/\equiv)^\infty) \neq \emptyset$</td>
</tr>
<tr>
<td></td>
<td>This enables <strong>query pruning (for query answering)</strong> without saturating $G$.</td>
</tr>
</tbody>
</table>
Which equivalence relations to use?

Equivalence notions previously studied
- Forward / backward / forward and backward simulation
- Forward / backward / forward and backward bisimulation

Adapted to semantic RDF graphs

Novel equivalence notions we introduce (see next)
- Flexible similarity suited to heterogeneous graphs
- Based on property cliques and possibly on RDF types
RDF node equivalence based on property cliques

Intuition: $a_1, a_2$ are similar; $r_1, r_2, r_3, r_4, r_5$ are similar

Diagram:

- Book
  - $r_6$ to $a_1$ for author
  - $r_1$ to $t_1$ for title
- Journal
  - $r_2$ to $t_2$ for title
  - $r_7$ to $e_1$ for editor
  - $r_3$ to $c_1$ for comment

- Spec
  - $r_4$ to $a_2$ for author
  - $r_5$ to $t_3$ for title
  - $r_8$ to $t_4$ for title
RDF node equivalence based on property cliques

Output property cliques: \{a, t, e, c\}; \{r\}; \{p\}; \emptyset
Input property cliques: \{a\}; \{t\}; \{e\}; \{c\}; \{r, p\}; \emptyset
Weak clique-based summaries

Two nodes are weakly equivalent ($\equiv W$) iff they have the same input clique or the same output clique.

Weak summary $G/\equiv W$ of the sample RDF graph $G$:

Property: In $G/\equiv W$, each data property appears exactly once $\Rightarrow$ its nodes are “source of $p$, target of $p$” for each $p$ [ČGM15b].
Weak clique-based summaries

Property: $G/\cong W$ nodes are “source of $p$, target of $p$” for each $p$.

Detecting errors in the data: why do the birthplace and deathplace loop?

Looking in the data, we find:

\[
\langle \text{http://dbpedia.org/resource/Kunitomo_Ikkansai} \rangle \langle \text{http://www.w3.org/1999/02/22-rdf-syntax-ns#type} \rangle \\
\langle \text{http://xmlns.com/foaf/0.1/Person} \rangle.
\]

\[
\langle \text{http://dbpedia.org/resource/Kunitomo_Ikkansai} \rangle \langle \text{http://dbpedia.org/ontology/birthPlace} \rangle \\
\langle \text{http://dbpedia.org/resource/Kunitomo_Ikkansai} \rangle.
\]
Strong clique-based summaries

Two nodes are strongly equivalent \((\equiv_S)\) iff they have the same input clique and the same output clique.

Strong summary \(G/\equiv_S\) of the sample RDF graph \(G\):
Using types for summarization

Group nodes **first by their types**; then group untyped nodes by their property cliques.

Typed weak summary $G/\equiv_{TW}$ of the sample RDF graph $G$:

On this example, this is also the typed strong summary $G/\equiv_{TS}$.
RDF summaries outline

<table>
<thead>
<tr>
<th>Summary</th>
<th>Weak?</th>
<th>Strong?</th>
<th>Types first?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/\equiv W</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/\equiv S</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/\equiv TW</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>G/\equiv TS</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
### RDF summaries outline

<table>
<thead>
<tr>
<th>Summary</th>
<th>Weak?</th>
<th>Strong?</th>
<th>FW bisim?</th>
<th>BW bisim?</th>
<th>Types first?</th>
</tr>
</thead>
<tbody>
<tr>
<td>G/≡ W</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/≡ S</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/≡ TW</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/≡ TS</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/≡ fw</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G/≡ bw</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/≡ fb</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/≡ fw,T</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/≡ bw,T</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>G/≡ fb,T</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Relations between RDF summaries [ČGM17b]
## Summary size comparison (more in [ČGM17b])

| Graph G | |G|  | Summary G/≡ | |G/≡|  | cf≡ |
|---------|-----------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| DBLP    | 150,787,464     | G/W                     | 71                      | 2,123,767               |
| DBLP    | 150,787,464     | G/S                     | 206                     | 731,978                 |
| DBLP    | 150,787,464     | G/fw                    | 262,695                 | 574                     |
| LUBM1M  | 1,227,868       | G/W                     | 161                     | 7,579                   |
| LUBM1M  | 1,227,868       | G/S                     | 207                     | 5,903                   |
| LUBM1M  | 1,227,868       | G/fw                    | 1982                    | 617                     |
| LUBM10M | 11,990,183      | G/W                     | 162                     | 74,013                  |
| LUBM10M | 11,990,183      | G/S                     | 206                     | 58,204                  |
| LUBM10M | 11,990,183      | G/fw                    | 24,958                  | 480                     |
| LUBM10M | 11,990,183      | G/bw                    | 6,162                   | 1,944                   |
| LUBM10M | 11,990,183      | G/fb                    | 11,990,076              | 1                       |
Summarizing $G^\infty$

Recall: With an RDF Schema, the semantics of $G$ is $G^\infty$ ⇒ We really need $(G^\infty)/\equiv$!

1. Saturate $G$, then summarize
2. Can we avoid saturating $G$?...
Recall: With an RDF Schema, the semantics of $G$ is $G^\infty \Rightarrow$ We really need $(G^\infty)/\equiv$!

1. Saturate $G$, then summarize
2. Can we avoid saturating $G$?

**Shortcut theorem** [ČGM17a]

For the summaries $G/W$, $G/S$, $G/fw$, $G/bw$, $G/fb$:

$$(G^\infty)/\equiv \text{ is the same as } ((G/\equiv)^\infty)/\equiv$$

Also: **sufficient condition** for any $\equiv$ to admit the shortcut.
Shortcut toward the summary of $G^\infty$

Direct $G \rightarrow \text{sat.} \rightarrow G^\infty \rightarrow \text{summ.} \rightarrow (G^\infty)_{\equiv}$

Shortcut $G \rightarrow \text{summ.} \rightarrow G_{\equiv} \rightarrow \text{sat.} \rightarrow (G_{\equiv})^\infty \rightarrow \text{summ.} \rightarrow ((G_{\equiv})^\infty)_{\equiv}$

If $G_{\equiv}$ is much smaller than $G$, the shortcut may be faster! Up to 20 times in our experiments [ČGM17b]
Shortcut example: $G_W$
Shortcut counter-example: $G_{TW}$
Summary-enabled LOD cloud exploration

ILDA Inria team (E. Pietriga, H. Ozaygen)
Use summary to derive visualisation instead of the original graph (smaller, faster)
Part III

Finding insights in RDF graphs
Insight in an RDF graph

We consider an insight to be the result of an aggregation query over the RDF graph. We focus on one-dimensional aggregates \( \Rightarrow \) 2D layout.

An insight is interesting if a certain measure (e.g., variance) on its set of aggregation values is high.

Problem

**Problem:** given a graph \( G \), find the top-\( k \) insights.
Dagger approach

**Dagger**: Digging for Interesting Aggregates in RDF Graphs [DMS17] (ongoing)

1. **Candidate facts** Resources from G: of a certain type, or having certain property sets
2. **Candidate dimension** Properties of the candidate facts, with strong support and relatively few distinct values. Also: derived properties, e.g., authors count;
3. **Candidate measure** Another property of the candidate facts Also: automatic value typing
4. **Candidate aggregation function** Chosen depending on the measure type
Dagger-selected aggregate in DBLP data

Average number of authors of journal articles, per publication year
Dagger-selected aggregate in DBLP data

Number of book authors, per book publication year
Dagger-selected aggregate in DBLP data

The number of books by each publisher (highest: Springer)
Part IV

Conclusion
The need for RDF graph discovery tools

- RDF graphs can be **large and complex**, they lack a prescriptive schema
- Semantic rules lead to **implicit data**
- Toward helping users to discover RDF graphs:
  1. **Structural quotient summaries** representing the complete graph structure; compact clique-based summaries; available at: https://team.inria.fr/cedar/projects/rdfsfsummary/
  2. **Insight discovery**: interesting aggregate queries; project Web page: https://team.inria.fr/cedar/projects/dagger/
- Many follow-up directions: parallelization, more interestingness measures, extensions to ML.


References (cont.)


