Learning Commonalities in RDF

Sara El Hassad  François Goasdoué  Hélène Jaudoin

IRISA, Univ. Rennes 1, Lannion, France

ESWC 2017 28th May - 1st June 2017
Introduction

Least general generalization (lgg)

- Machine Learning in the early 70’s by Gordon Plotkin
- Knowledge representation domain in the early 90’s
- Recently in semantic web
### Least general generalization ($lgg$)

- Machine Learning in the early 70’s by Gordon Plotkin
- Knowledge representation domain in the early 90’s
- Recently in semantic web

### Applications of $lgg$

- Social context : $lgg$ of users descriptions (profiles)
- Research common graph patterns between of datasets
- Linked Data Cloud : links between datasets
Introduction

Least general generalization (1gg)
- Machine Learning in the early 70’s by Gordon Plotkin
- Knowledge representation domain in the early 90’s
- Recently in semantic web

Applications of 1gg
- Social context : 1gg of users descriptions (profiles)
- Research common graph patterns between of datasets
- Linked Data Cloud : links between datasets

Goal
To study the problem in the setting of the entire RDF standard
Outline

Introduction

The Resource Description Framework

Finding commonalities between RDF graphs

Related work

Conclusion
RDF graphs

- Specification of RDF graphs with triples:
  \[(s, p, o) \in (U \cup B) \times U \times (U \cup L \cup B)\]

- Built-in property URIs to state RDF statements

<table>
<thead>
<tr>
<th>RDF statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class assertion</td>
<td>((s, \text{rdf:type}, o))</td>
</tr>
<tr>
<td>Property assertion</td>
<td>((s, p, o)) with (p \neq \text{rdf:type})</td>
</tr>
</tbody>
</table>
RDF graphs

- Specification of RDF graphs with triples:
  \[(s, p, o) \in (U \cup B) \times U \times (U \cup L \cup B)\]
- Built-in property URIs to state RDF statements

<table>
<thead>
<tr>
<th>RDF statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class assertion</td>
<td>((s, \text{rdf:type}, o))</td>
</tr>
<tr>
<td>Property assertion</td>
<td>((s, p, o)) with (p \neq \text{rdf:type})</td>
</tr>
</tbody>
</table>
Adding ontological knowledge to RDF graphs

- Built-in property URIs to state RDF Schema statements, i.e., ontological constraints.

<table>
<thead>
<tr>
<th>RDFS statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>$(s, \lessdot_{sc}, o)$</td>
</tr>
<tr>
<td>Subproperty</td>
<td>$(s, \lessdot_{sp}, o)$</td>
</tr>
<tr>
<td>Domain typing</td>
<td>$(s, \leftarrow_d, o)$</td>
</tr>
<tr>
<td>Range typing</td>
<td>$(s, \rightarrow_r, o)$</td>
</tr>
</tbody>
</table>
Adding ontological knowledge to RDF graphs

- Built-in property URIs to state RDF Schema statements, i.e., ontological constraints.

<table>
<thead>
<tr>
<th>RDFS statement</th>
<th>Triple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclass</td>
<td>((s, \preceq_{sc}, o))</td>
</tr>
<tr>
<td>Subproperty</td>
<td>((s, \preceq_{sp}, o))</td>
</tr>
<tr>
<td>Domain typing</td>
<td>((s, \leftarrow_d, o))</td>
</tr>
<tr>
<td>Range typing</td>
<td>((s, \leftarrow_r, o))</td>
</tr>
</tbody>
</table>

\[
\begin{array}{ll}
\text{ConfPaper} & \text{hasContactAuthor} \\
\text{Publication} & \text{hasAuthor} \\
\text{Researcher} &
\end{array}
\]

\[
\begin{array}{l}
\text{b} \quad \text{hasTitle} \quad "\text{LGG in RDF}" \\
\tau \quad \text{hasContactAuthor} \\
\end{array}
\]
Deriving the implicit triples

Figure: RDF graph $G$
Deriving the implicit triples

Figure: RDF graph $G$
Deriving the implicit triples

Figure: RDF graph $\mathcal{G}$
Deriving the implicit triples

Figure: RDF graph $G$
Deriving the implicit triples

Figure: RDF graph $G$
Deriving the implicit triples

Figure: RDF graph $G$
Deriving the implicit triples

How to derive implicit triples of an RDF graph?
Sample set of entailment rules

<table>
<thead>
<tr>
<th>Rule [7]</th>
<th>Entailment rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs2</td>
<td>((p, \leftarrow_d, o), (s_1, p, o_1) \rightarrow (s_1, \tau, o))</td>
</tr>
<tr>
<td>rdfs3</td>
<td>((p, \leftarrow_r, o), (s_1, p, o_1) \rightarrow (o_1, \tau, o))</td>
</tr>
<tr>
<td>rdfs5</td>
<td>((p_1, \lessdot_{sp}, p_2), (p_2, \lessdot_{sp}, p_3) \rightarrow (p_1, \lessdot_{sp}, p_3))</td>
</tr>
<tr>
<td>rdfs7</td>
<td>((p_1, \lessdot_{sp}, p_2), (s, p_1, o) \rightarrow (s, p_2, o))</td>
</tr>
<tr>
<td>rdfs9</td>
<td>((s, \lessdot_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o))</td>
</tr>
<tr>
<td>rdfs11</td>
<td>((s, \lessdot_{sc}, o), (o, \lessdot_{sc}, o_1) \rightarrow (s, \lessdot_{sc}, o_1))</td>
</tr>
<tr>
<td>ext1</td>
<td>((p, \leftarrow_d, o), (o, \lessdot_{sc}, o_1) \rightarrow (p, \leftarrow_d, o_1))</td>
</tr>
<tr>
<td>ext2</td>
<td>((p, \leftarrow_r, o), (o, \lessdot_{sc}, o_1) \rightarrow (p, \leftarrow_r, o_1))</td>
</tr>
<tr>
<td>ext3</td>
<td>((p, \lessdot_{sp}, p_1), (p_1, \leftarrow_d, o) \rightarrow (p, \leftarrow_d, o))</td>
</tr>
<tr>
<td>ext4</td>
<td>((p, \lessdot_{sp}, p_1), (p_1, \leftarrow_r, o) \rightarrow (p, \leftarrow_r, o))</td>
</tr>
</tbody>
</table>
Semantics of RDF graphs

Figure: RDF graph $G$
Semantics of RDF graphs

\[ rdfs9 : (s, \leq_{sc}, o), (s_1, \tau, s) \rightarrow (s_1, \tau, o) \]

Figure: RDF graph \( \mathcal{G} \)
Semantics of RDF graphs

\[ rdfs7 : (p_1, \preceq_{sp} p_2), (s, p_1, o) \rightarrow (s, p_2, o) \]

Figure: RDF graph \( G \)
**Semantics of RDF graphs**

\[ rdfs3 : (p, \rightarrow_r, o), (s_1, p, o_1) \rightarrow (o_1, \tau, o) \]
Semantics of RDF graphs

ext4 : (p, ⊑_{sp}, p_1), (p_1, ↪_r, o) → (p, ↪_r, o)

Figure: RDF graph $G$
Semantics of RDF graphs

\[ ext3 : (p, \preceq_{sp}, p_1), (p_1, \leftarrow_{d}, o) \rightarrow (p, \leftarrow_{d}, o) \]

Figure: RDF graph $G$
Semantics of RDF graphs

Figure: Saturated RDF graph $G^\infty$
Let $G$ and $G'$ be two graphs RDF and $R$ a set of RDF entailment rules. There exists relationship to compare $G$ and $G'$ called *entailment between graphs*. $G$ is more specific than $G'$:

- $G \models_R G' \iff G^\infty \models G'$

There must exist an embedding of $G'$ in $G^\infty$. 
Entailment between RDF graphs

\[ G \models \mathcal{R} G' \]
Entailment between RDF graphs

\[ G^\infty \models G' \]

\[ b \]

hasTitle \rightarrow "LGG in RDF"

hasAuthor

hasContactAuthor

ConfPaper

Publication

hasContactAuthor

\[ \preceq_{sc} \]

\[ \preceq_d \]

hasAuthor

Researcher

\[ \preceq_{sp} \]

\[ \preceq_r \]

\[ \tau \]

Publication

\[ b_1 \]

\[ b_2 \]

\[ G^\infty \]

\[ G' \]
Entailment between RDF graphs

\[ G^\infty \models G' \]

RDF graph \( G \) is more specific than RDF graph \( G' \)
Outline

Introduction

The Resource Description Framework

Finding commonalities between RDF graphs

Related work

Conclusion
Towards defining lgg in RDF

A least general generalization (1gg) of \( n \) descriptions \( d_1, \ldots, d_n \) is a most specific description \( d \) generalizing every \( d_{1 \leq i \leq n} \) for some generalization/specialization relation between descriptions (G. Plotkin).

### lgg in RDF

- descriptions are RDF graphs
- relation generalization/specialization is entailment between RDF graphs
Defining the lgg of RDF graphs

Definition (lgg of RDF graphs)

Let $G_1, \ldots, G_n$ be RDF graphs and $\mathcal{R}$ a set of RDF entailment rules.

- A generalization of $G_1, \ldots, G_n$ is an RDF graph $G_g$ such that $G_i \models_{\mathcal{R}} G_g$ holds for $1 \leq i \leq n$.

- A least general generalization (lgg) of $G_1, \ldots, G_n$ is a generalization $G_{1\text{gg}}$ of $G_1, \ldots, G_n$ such that for any other generalization $G_g$ of $G_1, \ldots, G_n$, $G_{1\text{gg}} \models_{\mathcal{R}} G_g$ holds.

Result: lgg of $n$ RDF graphs vs lgg of two RDF graphs

\[ \ell_3(G_1, G_2, G_3) \equiv_{\mathcal{R}} \ell_2(\ell_2(G_1, G_2), G_3) \]
\[ \ldots \]
\[ \ell_n(G_1, \ldots, G_n) \equiv_{\mathcal{R}} \ell_2(\ell_{n-1}(G_1, \ldots, G_{n-1}), G_n) \]
\[ \equiv_{\mathcal{R}} \ell_2(\ell_2(\cdots \ell_2(\ell_2(G_1, G_2), G_3) \cdots , G_{n-1}), G_n) \]
Defining the lgg of RDF graphs

**Definition (lgg of RDF graphs)**

Let $G_1, \ldots, G_n$ be RDF graphs and $\mathcal{R}$ a set of RDF entailment rules.

- A **generalization** of $G_1, \ldots, G_n$ is an RDF graph $G_g$ such that $G_i \models_{\mathcal{R}} G_g$ holds for $1 \leq i \leq n$.
- A **least general generalization** (lgg) of $G_1, \ldots, G_n$ is a generalization $G_{1\text{gg}}$ of $G_1, \ldots, G_n$ such that for any other generalization $G_g$ of $G_1, \ldots, G_n$, $G_{1\text{gg}} \models_{\mathcal{R}} G_g$ holds.

**Result : lgg of n RDF graphs vs lgg of two RDF graphs**

\[
\ell_3(G_1, G_2, G_3) \equiv_{\mathcal{R}} \ell_2(\ell_2(G_1, G_2), G_3) \\
\cdots \cdots \\
\ell_n(G_1, \ldots, G_n) \equiv_{\mathcal{R}} \ell_2(\ell_{n-1}(G_1, \ldots, G_{n-1}), G_n) \\
\quad \equiv_{\mathcal{R}} \ell_2(\ell_2(\cdots \ell_2(\ell_2(G_1, G_2), G_3) \cdots, G_{n-1}), G_n)
\]

We focus on computing lgg of two RDF graphs
Defining the $\lgg$ of RDF graphs

$G_1$

$G_2$
Defining the \( \text{lgg} \) of RDF graphs

\[ \mathcal{G}_1 \]

\[ \mathcal{G}_2 \]

\[ \mathcal{G}_{1\text{gg}} \]
Defining the lgg of RDF graphs

$\mathcal{G}_1$

$\mathcal{G}_2$

$\mathcal{G}_{1\text{gg}}$

How to compute this graph?
The cover graph of RDF graphs

Definition (Cover graph)

The **cover graph** $G$ of two RDF graph $G_1$ and $G_2$ is the RDF graph such that for every property $p$ in both $G_1$ and $G_2$:

$$(t_1, p, t_2) \in G_1 \text{ and } (t_3, p, t_4) \in G_2 \text{ iff } (t_5, p, t_6) \in G$$

with $t_5 = t_1$ if $t_1 = t_3$ and $t_1 \in \mathcal{U} \cup \mathcal{L}$, else $t_5$ is the blank node $b_{t_1 t_3}$, and, similarly $t_6 = t_2$ if $t_2 = t_4$ and $t_2 \in \mathcal{U} \cup \mathcal{L}$, else $t_6$ is the blank node $b_{t_2 t_4}$. 
The cover graph of RDF graphs

\[ G_1 \]

\[ G_2 \]
The cover graph of RDF graphs

\[ G_1 \]

\[ b_{i_1} \quad \text{hasAuthor} \quad b_{i_2} \]

\[ b_{i_2} \quad \text{hasAuthor} \quad b_{SAVV} \]

\[ G_2 \]
The cover graph of RDF graphs

\[ G_1 \]

\[ G_2 \]

\[ b_{n_1} \quad b_{n_2} \quad b_{AVV} \]
The cover graph of RDF graphs

\[ G_1 \]

\[ G_2 \]

\[ \text{SA} \quad \rightarrow \quad \text{Researcher} \]

\[ i_1 \quad \rightarrow \quad \text{Publication} \quad \leq_{sc} \quad \text{ConfPaper} \quad \text{"DiD"} \]

\[ \text{VV} \quad \rightarrow \quad \text{Researcher} \]

\[ i_2 \quad \rightarrow \quad \text{Publication} \quad \leq_{sc} \quad \text{JourPaper} \quad \text{"CwFOL"} \]

\[ \text{SA} \quad \rightarrow \quad \text{Researcher} \]

\[ \text{hasAuthor} \]

\[ \text{title} \]

\[ \text{"DiD"} \]

\[ \text{hasAuthor} \]

\[ \text{title} \]

\[ \text{"CwFOL"} \]
The cover graph of RDF graphs

$G_1$

$G_2$

Finding commonalities between RDF graphs
The cover graph of RDF graphs

\[ G_1 \]

\[ G_2 \]
Cover graph vs lgg

**Theorem \((\mathcal{R} = \emptyset)\)**

The *cover graph* \(\mathcal{G}\) of the RDF graphs \(\mathcal{G}_1\) and \(\mathcal{G}_2\) is an lgg of them for the empty set \(\mathcal{R}\) of RDF entailment rules (i.e., \(\mathcal{R} = \emptyset\)).
Cover graph vs lgg

**Theorem ($\mathcal{R} = \emptyset$)**

The cover graph $\mathcal{G}$ of the RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is an lgg of them for the empty set $\mathcal{R}$ of RDF entailment rules (i.e., $\mathcal{R} = \emptyset$).

**Proposition ($\mathcal{R} = \emptyset$)**

The cover graph of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be computed in $O(|\mathcal{G}_1| \times |\mathcal{G}_2|)$; its size is bounded by $|\mathcal{G}_1| \times |\mathcal{G}_2|$.
Cover graph vs lgg

**Theorem ($\mathcal{R} = \emptyset$)**

The cover graph $\mathcal{G}$ of the RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is an lgg of them for the empty set $\mathcal{R}$ of RDF entailment rules (i.e., $\mathcal{R} = \emptyset$).

**Proposition ($\mathcal{R} = \emptyset$)**

The cover graph of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be computed in $O(|\mathcal{G}_1| \times |\mathcal{G}_2|)$; its size is bounded by $|\mathcal{G}_1| \times |\mathcal{G}_2|$.

**Theorem ($\mathcal{R} \neq \emptyset$)**

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two RDF graphs, and $\mathcal{R}$ a set of RDF entailment rules. The cover graph $\mathcal{G}$ of $\mathcal{G}_1^\infty$ and $\mathcal{G}_2^\infty$ is an lgg of $\mathcal{G}_1$ and $\mathcal{G}_2$. 
Cover graph vs lgg

Theorem ($\mathcal{R} = \emptyset$)

The cover graph $\mathcal{G}$ of the RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is an lgg of them for the empty set $\mathcal{R}$ of RDF entailment rules (i.e., $\mathcal{R} = \emptyset$).

Proposition ($\mathcal{R} = \emptyset$)

The cover graph of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be computed in $O(|\mathcal{G}_1| \times |\mathcal{G}_2|)$; its size is bounded by $|\mathcal{G}_1| \times |\mathcal{G}_2|$.

Theorem ($\mathcal{R} \neq \emptyset$)

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two RDF graphs, and $\mathcal{R}$ a set of RDF entailment rules. The cover graph $\mathcal{G}$ of $\mathcal{G}_1^\infty$ and $\mathcal{G}_2^\infty$ is an lgg of $\mathcal{G}_1$ and $\mathcal{G}_2$.

Corollary ($\mathcal{R} \neq \emptyset$)

An lgg of two RDF graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be computed in $O(|\mathcal{G}_1^\infty| \times |\mathcal{G}_2^\infty|)$ and its size is bounded by $|\mathcal{G}_1^\infty| \times |\mathcal{G}_2^\infty|$.
Cover graph vs lgg

\[
\begin{align*}
&\text{SA} \xrightarrow{\tau} \text{Researcher} \\
&b_{i_1i_2} \xrightarrow{\tau} b_{SAV} \\
&b_{SAi_1} \xrightarrow{\tau} b_{RJP} \\
&b_{DC} \xrightarrow{\tau} b_{CPJP} \leq_{sc} \text{Publication} \\
&b_{PJP} \xrightarrow{\tau} b_{CPP} \xrightarrow{\tau} b_{CPJP} \leq_{sc} \text{Publication} \\
&b_{RP} \xrightarrow{\tau} b_{SAi_2} \xrightarrow{\tau} b_{RJP}
\end{align*}
\]
Outline

Introduction

The Resource Description Framework

Finding commonalities between RDF graphs

Related work

Conclusion
Related work

Structural based approach

- Description Logics $\mathcal{EL}$

- RDF
  - SPARQL : tree queries
  - Rooted graphs, ignore RDF entailment :
    - S. Colucci and al. : *Defining and computing least common subsumers in RDF*. J. Web Semantics, 39(0), 2016.

Independent structure approach

- Conceptual Graphs
Conclusion

- Revisit the problem of computing a least general generalization in the entire setting of RDF.
- Algorithms to compute lgg's of small-to-huge RDF graphs.
  - Memory
  - Data management system
  - MapReduce
- Perspective: Heuristics in order to compute lgg without redundants triples.
Thank you!

Questions?
References I

Computing least common subsumers in description logics with existential restrictions.
In *IJCAI*, 1999.

Computing the least common subsumer w.r.t. a background terminology.

*Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs*.

Defining and computing least common subsumers in RDF.
*J. Web Semantics*, 39(0), 2016.

Common subsumers in RDF.
In *AI*/*IA*, 2013.

Autosparql: Let users query your knowledge base.
In *ESWC*, 2011.

https://www.w3.org/TR/rdf11-mt/.

Most specific generalizations w.r.t. general EL-TBoxes.
In *IJCAI*, 2013.