RDF Ontology (Re-)Engineering through large-scale Data Mining

Billion Triple Challenge 2011

Johannes Lorey
Ziawasch Abedjan
Felix Naumann
Christoph Böhm

http://tinyurl.com/hpi-btc2011
Motivation

RDF Ontology (Re-)Engineering through Large-scale Data Mining | Christoph Böhm | BTC

Data Publisher

ont:Person

foaf:firstName
foaf:lastName
foaf:geekcode
foaf:myersBriggs

uses

ont:Person

foaf:firstName
foaf:lastName
imdb:acted
imdb:directed

Integration

queries

Data Consumer

ont:Person

foaf:firstName
foaf:lastName
purl:neighborOf
purl:livesWith

uses

ont:Person

foaf:firstName
foaf:lastName

ont:Person

foaf:firstName
foaf:lastName

ont:Person

foaf:firstName
foaf:lastName
We identified two basic misconceptions about class specification:

- Overspecification
- Underspecification
Some properties are specified in the ontology, but rarely used in instance data

- Data publishers are unaware of specified attributes
- Attributes are ambiguous/unfitting/omissible
Some properties occur frequently in instance data, but are not specified in the ontology.

Reasons include:

- Data publishers introduce new attributes (although existing ones might be suitable)
- Ontology definition lacks properties
An instance-data driven approach

- Perform **predicate frequency** analysis
- Perform **rule mining** to identify usage patterns
  - Positive rules: predicates co-occur frequently
    \((\text{foaf:gender} \Rightarrow \text{foaf:weblog})\)
  - Negative rules: predicates occur exclusive
    \((\text{foaf:weblog} \Rightarrow \neg \text{foaf:homepage})\)
- **Match against class definition** to offer re-engineering suggestions such as
  - Adding new properties to class definition
  - Removing properties from class definition
  - Creating subclasses and pushing properties down
Workflow using BTC data

BTC Dump → Type Extractor → Types → Typed Entity Extractor → Typed instances

Class Definition Lookup → Class definitions

Matching Evaluation

Predicate Analysis → Mining Algorithm (FP-Growth) → Association Rules

Predicate Statistics
The data

- around **475GB** of uncompressed data

- **213,382** RDF types

- **441,461,669** typed instances

- type definitions for around **90%** of all instances discovered within BTC dataset

- we ran our approach on the most frequent types
### RDF Type Usage in the BTC dataset

<table>
<thead>
<tr>
<th>Type</th>
<th>#Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>foaf:Person</td>
<td>362,590,928</td>
</tr>
<tr>
<td>foaf:OnlineAccount</td>
<td>2,938,416</td>
</tr>
<tr>
<td>foaf:Document</td>
<td>1,252,681</td>
</tr>
<tr>
<td>rdf:Statement</td>
<td>887,363</td>
</tr>
<tr>
<td>foaf:Image</td>
<td>876,863</td>
</tr>
<tr>
<td>mo:MusicArtist</td>
<td>310,529</td>
</tr>
<tr>
<td>foaf:Agent</td>
<td>204,435</td>
</tr>
</tbody>
</table>

- 3,893 „new properties“ found for foaf:Person

<table>
<thead>
<tr>
<th>Property</th>
<th>#Instances</th>
<th>Status:testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>foaf:nick</td>
<td>361,937,602</td>
<td>Undefined property</td>
</tr>
<tr>
<td>foaf:member_name</td>
<td>19,083,000</td>
<td>Undefined property</td>
</tr>
<tr>
<td>foaf:tagLine</td>
<td>19,062,451</td>
<td>Misusage</td>
</tr>
<tr>
<td>foaf:Image</td>
<td>18,033,515</td>
<td>Most common &quot;name&quot;</td>
</tr>
<tr>
<td>foaf:name</td>
<td>3,597,768</td>
<td>property</td>
</tr>
</tbody>
</table>
The BTC dataset provides a large-scale heterogeneous snapshot of the Web of Data and reveals common (mis-)use patterns.
(Re-)Engineering overspecified class definitions

<table>
<thead>
<tr>
<th>Resource</th>
<th>#Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>foaf:Person</td>
<td>362,590,928</td>
</tr>
<tr>
<td>• foaf:myersBriggs</td>
<td>9</td>
</tr>
<tr>
<td>• foaf:geekcode</td>
<td>7</td>
</tr>
<tr>
<td>• foaf:plan</td>
<td>1</td>
</tr>
</tbody>
</table>

- Data publishers do not provide suitable values for these properties
- Data consumers will have no luck in trying to query for these properties
  - Remove the properties from this class definition (or possibly move them to subclasses)
(Re-)Engineering underspecified class definitions

- Data publishers add more properties (and information) to a predefined class
- Data consumers considering the original class will retrieve unexpected information
- However, blindly adding this property to the class definition might result in fragmentation

➤ Consider mined rules for re-engineering suggestions

<table>
<thead>
<tr>
<th>Resource</th>
<th>#Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>foaf:OnlineAccount</td>
<td>2,938,416</td>
</tr>
<tr>
<td>• foaf:accountName</td>
<td>2,859,090</td>
</tr>
<tr>
<td>• <a href="http://rdfs.org/sioc/ns#account_of">http://rdfs.org/sioc/ns#account_of</a></td>
<td>2,411,233</td>
</tr>
</tbody>
</table>
(Re-)Engineering underspecified class definitions

<table>
<thead>
<tr>
<th>Resource</th>
<th>#Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>foaf:OnlineAccount</td>
<td>2,938,416</td>
</tr>
<tr>
<td>• foaf:accountName</td>
<td>2,859,090</td>
</tr>
<tr>
<td>• <a href="http://rdfs.org/sioc/ns#account_of">http://rdfs.org/sioc/ns#account_of</a></td>
<td>2,411,233</td>
</tr>
</tbody>
</table>

Mining results:

- `foaf:accountName` ⇒ `http://rdfs.org/sioc/ns#account_of`
- Originally specified property co-occurs frequently with new unspecified property
- Add new property to ontology
(Re-)Engineering underspecified class definitions

<table>
<thead>
<tr>
<th>Resource</th>
<th>#Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>mo:MusicArtist</td>
<td>310,529</td>
</tr>
<tr>
<td>• musicbrainz:isInstrumentalArtistOf</td>
<td>4,015</td>
</tr>
<tr>
<td>• musicbrainz:isEngineerOf</td>
<td>1,594</td>
</tr>
<tr>
<td>• musicbrainz:isMixEngineerOf</td>
<td>1,223</td>
</tr>
</tbody>
</table>

Mining results:

- `musicbrainz:isInstrumentalArtistOf` ⇒ ¬ `musicbrainz:isEngineerOf`
- `musicbrainz:isInstrumentalArtistOf` ⇒ ¬ `musicbrainz:isMixEngineerOf`

- Pairwise negative rules for frequent attributes
- Split up class into disjunct subclasses
Summary

- We found certain divergence types between class definitions and usage patterns
- We propose predicate frequencies and association rules as means to suggest possible alterations

Goals:
- Encourage ontology reengineering
- Better (typed) data for consumers