Scalable Extraction of Implicit and Explicit Schema Information in Linked Open Data

Ansgar Scherp
Data and Web Science, U Mannheim

Matthias Konrath, Thomas Gottron
Web Science and Technologies, U Koblenz
Linked Data

- We witness a major movement in the Web …
- Publishing and interlinking of data of different quality, purpose and source on the Web
- Technology + Social Phenomenon

<table>
<thead>
<tr>
<th>World Wide Web</th>
<th>Linked Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>Data</td>
</tr>
<tr>
<td>Hyperlinks</td>
<td>Typed Links</td>
</tr>
<tr>
<td>HTML</td>
<td>RDF</td>
</tr>
<tr>
<td>Addresses (URIs)</td>
<td>Addresses (URIs)</td>
</tr>
</tbody>
</table>
Relevance of Linked Data?
Linked Data: May ‘07 ➔ Sept. ‘11

< 31 Billion Triples

Source: http://lod-cloud.net
Linked Data: Based on Four Principles

1. Identification
2. Interlinkage
3. Dereferencing
4. Description
1. Use URIs for Identification

Matt Briggs
http://biglynx.co.uk/people/matt-briggs

Scott Miller
http://biglynx.co.uk/people/scott-miller
2. Interlinking of Resources
3. Dereferencing of URIs

- Pretty easy to look up web documents
- How do we “look up” things of the real world?
4. Description of URIs

- foaf:Person
- dp:Birmingham
- ex:loc
- _1:point
  - wgs84:long
  - "-0.118"
  - wgs84:lat
  - "51.509"

- biglynx:matt-briggs
  - foaf:knows
  - foaf:based_near

- biglynx:scott-miller
  - foaf:based_near
  - dp:London

Ansgar Scherp – ansgar@informatik.uni-mannheim.de
Searching for Data Sources

Persons that are
- Politicians and
- Actors

< 31 Milliarde Triples

Quelle: http://lod-cloud.net
Searching for Data Sources (2)

- No single federated query interface provided
- Schema-based index to find data sources

```sql
SELECT ?x
FROM ...
WHERE {
  ?x rdf:type ex:Actor .
  ?x rdf:type ex:Politician .
}
Idea

- Schema-based index
  - Define families of graph patterns
  - Assign instances to graph patterns
  - Store the source information (context URI)

- Construction
  - Stream-based for scalability
  - Little loss of accuracy
Input Data

- n-Quads

  <subject> <predicate> <object> <context>

- Example:

  
  <http://www.w3.org/People/Connolly/#me>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://xmlns.com/foaf/0.1/Person>
  <http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf>

http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf

w3p: #me

foaf: Person

Linked Open Data
Layer 1: RDF Classes

- All instances of a particular type

```
SELECT ?x
FROM ...
WHERE {
  ?x rdfs:type foaf:Person .
}
```
Layer 2: Type Clusters

- All instances belonging to exactly the same set of types

```
SELECT ?x
FROM ... WHERE {
  ?x rdfs:type foaf:Person .
  ?x rdfs:type pim:Male .
}
```
Layer 3: Equivalence Classes

- Two instances are equivalent iff:
  - They are in the same TC
  - They have the same properties
  - The property targets are in the same TC

- Similar to 1-bisimulation
Layer 3: Equivalence Classes

```
SELECT ?x
WHERE {
  ?x rdfs:type foaf:Person .
  ?x rdfs:type pim:Male .
  ?x foaf:maker ?y .
  ?y rdfs:type
    foaf:PersonalProfileDocument .
}
```
Building the Schema and Index

- RDF classes
- Type clusters
- Equivalence classes
- Data sources
Building the Index from a Stream

- Stream of n-quads (coming from a LD crawler)

• Linear runtime complexity wrt # of input triples
Computing SchemEX: TimBL Data Set

- Analysis of a smaller data set
- 11 M triples, TimBL’s FOAF profile
- LDspider with ~ 2k triples / sec

- Different cache sizes: 100, 1k, 10k, 50k, 100k
- Compared SchemEX with reference schema
- Index queries on all Types, TCs, EQCs
- Good precision/recall ratio at 50k+
Quality of Stream-based Index Construction

- Runtime increases hardly with window size
- Memory consumption scales with window size
- Commodity hardware (4GB RAM, single CPU)
## Computing SchemEX: Full BTC 2011 Data

<table>
<thead>
<tr>
<th></th>
<th>1st billion</th>
<th>2nd billion</th>
<th>full dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>#triples</td>
<td>1 billion</td>
<td>1 billion</td>
<td>2.17 billion</td>
</tr>
<tr>
<td>#instances</td>
<td>187.7M</td>
<td>222.6M</td>
<td>450.0M</td>
</tr>
<tr>
<td>#data sources</td>
<td>13.5M</td>
<td>9.5M</td>
<td>24.1M</td>
</tr>
<tr>
<td>#type clusters</td>
<td>208.5k</td>
<td>248.5k</td>
<td>448.6k</td>
</tr>
<tr>
<td>#equivalence classes</td>
<td>0.97M</td>
<td>1.14M</td>
<td>2.12M</td>
</tr>
<tr>
<td>#triples index</td>
<td>29.1M</td>
<td>24.8M</td>
<td>54.7M</td>
</tr>
<tr>
<td>Compression ratio</td>
<td>2.91%</td>
<td>2.48%</td>
<td>2.52%</td>
</tr>
<tr>
<td>runtime (hh:mm)</td>
<td>6:51</td>
<td>6:05</td>
<td>15:16</td>
</tr>
<tr>
<td>average runtime</td>
<td>247 s</td>
<td>219 s</td>
<td>252 s</td>
</tr>
<tr>
<td>standard deviation</td>
<td>80 s</td>
<td>12 s</td>
<td>57 s</td>
</tr>
<tr>
<td>#triples/sec.</td>
<td>40.5k</td>
<td>45.6k</td>
<td>39.5k</td>
</tr>
</tbody>
</table>

Cache size: 50 k
Billion Triple Challenge 2011

News

Winners of the Semantic Web Challenge

Billion Triples Track

Winner: SchemEX -- Web-Scale Indexed Schema Extraction of Linked Open Data

Mathias Konrath, Thomas Gottron, and Ansgar Scherp

[JWS’12]

Ansgar Scherp – ansgar@informatik.uni-mannheim.de
Did you mean?

Result Set Size

Ranked Retrieval

Result Snippets

Related Queries

Add: ?x foaf:name ?unknown0
Add: ?x foaf:page ?unknown0
Add: ?x dct:term:subject ?unknown0