Linked Data on the Web (LDOW 2012)

Benchmarking the Performance of Linked Data Translation Systems

Carlos R. Rivero¹, Andreas Schultz², Christian Bizer² and David Ruiz¹

¹ University of Sevilla
² Freie Universität Berlin
Outline

- Motivation
- Mapping Patterns
- LODIB Benchmark
- Benchmark Results
Motivation

- Web of Data is heterogeneous
- Many different and overlapping ways to represent information

Distribution of the most widely used vocabularies
Data is represented...

- Using terms from a wide range of vocabularies
- Using diverging structures
- Values are represented in differently
  - Different data types
  - Different measuring units
  - Fine grained vs. aggregated
Outline

- Motivation
- Mapping Patterns
- LODIB Benchmark
- Benchmark Results
Data Sets from the LOD Cloud
We extracted 15 mapping patterns

- Each is defining an atomic data translation operation.
- These patterns covered all the necessary operations we needed to translate instances for the LOD cloud sample.
## Mapping Patterns

<table>
<thead>
<tr>
<th>Code - Name</th>
<th>Source triples</th>
<th>Target triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC – Rename Class</td>
<td>?x a C_s</td>
<td>?x a C_t</td>
</tr>
<tr>
<td>RP – Rename Property</td>
<td>?x P_s ?y</td>
<td>?x P_t ?y</td>
</tr>
<tr>
<td>RCP – RC based on Property</td>
<td>?x a C_s . EXISTS {?x P ?y}</td>
<td>?x a C_t</td>
</tr>
<tr>
<td>RCV – RC based on Value</td>
<td>?x a C_s . ?x P v</td>
<td>?x a C_t</td>
</tr>
<tr>
<td>RvP – Reverse Property</td>
<td>?x P_s ?y</td>
<td>?y P_t ?x</td>
</tr>
<tr>
<td>1:1 – Value Transformation 1:1</td>
<td>?x P_s ?y</td>
<td>?x P_t f(?y)</td>
</tr>
<tr>
<td>VtU – Value to URI</td>
<td>?x P_s ?y</td>
<td>?x P_t URI(?y)</td>
</tr>
<tr>
<td>UtV – URI to Value</td>
<td>?x P_s ?y</td>
<td>?x P_t LITERAL(?y)</td>
</tr>
<tr>
<td>CD – Change Datatype</td>
<td>?x P_s ?yˆ^DT_s</td>
<td>?x P_t ?yˆ^DT_t</td>
</tr>
<tr>
<td>ALT – Add Language Tag</td>
<td>?x P_s ?y</td>
<td>?x P_t ?y@TAG</td>
</tr>
<tr>
<td>RLT – Remove Language Tag</td>
<td>?x P_s ?yˆ@TAG</td>
<td>?x P_t ?y</td>
</tr>
<tr>
<td>N:1 – Value Transformation N:1</td>
<td>?x P_1 ?v_1 ... ?x P_n ?v_n</td>
<td>?x P_t f(?v_1, ..., ?v_n)</td>
</tr>
<tr>
<td>Agg – Aggregate</td>
<td>?x P_s ?y</td>
<td>?x P_t AGG(?y)</td>
</tr>
</tbody>
</table>
Structural Mapping Patterns
Rename class based on the existence of a property relation.

```
dbpedia:William_Shakespeare a dbpedia-owl:Person ;
dbpedia-owl:deathDate "1616-04-23"^^xsd:date .
```

```
dbpedia:William_Shakespeare a fb:people.deceased_person .
```
Instances of the source class become instances of the target class if they have a specific property value.

```
@gw-p:Kurt_Joachim_Lauk_euParliament_1840_P a gw:Person ;
gw:profession "politician"^^xsd:string .
```

```
@gw-p:Kurt_Joachim_Lauk_euParliament_1840_P ;
a fb:government.politician .
```
Represent an attribute by a newly created resource that then carries the attribute value.

dbpedia:The_Usual_Suspects
dbpedia-owl:runtime 6360.0 .

dbpedia:The_Usual_Suspects po:version _:new .
_:new po:duration 6360.0 .
Value Transformation based Mapping Patterns
Transform the value of a data type property.

dbpedia:The_Shining_(film)
  dbpedia-owl:runtime  8520 .

Runtime in seconds

Runtime in minutes

dbpedia:The_Shining_(film)
  movie:runtime  142 .
VtU - Value to URI

Transform a literal value into a URI.

dbpedia:Von_Willebrand_disease
dbpedia-owl:omim 193400 .

dbpedia:Von_Willebrand_disease
N:1 - Transform Value N:1

Transform multiple values from different properties to a single value.

dbpedia:William_Shakespeare
  foaf:givenName   "William" ;
  foaf:surname     "Shakespeare" .

  dbpedia:William_Shakespeare
  foaf:name       "Shakespeare, William" .
Outline

- Motivation
- Mapping Patterns
- LODIB Benchmark
- Benchmark Results
LODIB Benchmark

- Based on a made up use case about products, reviews and persons
- Goal: Translating data from three different source data sets to the target representation
- Data for the use case is generated by a scalable data generator
- Frequency of mapping patterns corresponds to the statistics that we discuss next
LODIB Grounding

- We analysed 84 examples in the LOD Cloud
- Criteria: more than 25,000 owl:sameAs
- Selected Linked Data sources:
  - ACM (Publications)
  - DBLP (Publications)
  - Dailymed (Life Sciences)
  - Drug Bank (Life Sciences)
  - DataGov Statistics (Government)
  - Ordnance Survey (Government)
  - Dbpedia (Cross-domain)
  - GeoNames (Geographic)
  - Linked GeoData (Geographic)
  - LinkedMDB (Media)
  - New York Times (Media)
  - Music Brainz (Media)
  - Sider (Life Sciences)
  - GovWILD (Government)
  - ProductDB (Cross-domain)
  - OpenLibrary (Publications)
For all examples \((i_1 \text{owl:sameAs } i_2)\) for a pair of data sets \((d_1, d_2)\), where \(i_1\) and \(i_2\) are instances of \(d_1\) respectively \(d_2\):

- Count the occurrences of mapping patterns in the direction from \(i_1\) to \(i_2\)
- Average over all examples for each pair \((d_1, d_2)\)
- Average over the results of the previous step
• 62% simple renaming patterns (RC, RP)
• 17% structural mapping patterns (RCP, RCV, RvP, Rsc, DRsc)
• 12% changing the type of RDF nodes (VtU, UtV, CD, ALT, RLT)
• 9% value transformations (1:1, N:1)
• <1% aggregation
1) Expressivity
   - Number of expressible mapping patterns
   - Results are verified by test driver

2) Run time performance
   - Time needed to translate source data
     - Time span between reading the input and serializing the output files
   - Input: N-Triples files
   - Output: N-Triples file(s)
Outline

- Motivation
- Mapping Patterns
- LODIB Benchmark
- Benchmark Results
**Systems Under Test**

**Mosto**
- Automatically generates SPARQL Construct queries given a set of correspondences and constraints.

**LDIF**
- Extract, Transform, Load (ETL) tool for Linked Open Data
- R2R as mapping language

**Jena TDB**
- RDF store
- SPARQL 1.1 Construct queries as mapping
Results: Expressivity

<table>
<thead>
<tr>
<th></th>
<th>RC</th>
<th>RP</th>
<th>RCP</th>
<th>RCV</th>
<th>RvP</th>
<th>RsC</th>
<th>DRsc</th>
<th>1:1</th>
<th>VtU</th>
<th>UtV</th>
<th>CD</th>
<th>ALT</th>
<th>RLT</th>
<th>N:1</th>
<th>Agg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosto queries</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SPARQL 1.1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>R2R</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

RCP: Rename Class based on Property
RCV: Rename Class based on Value
Agg: Aggregate
## Results: Runtime Performance

Runtime results in seconds:

<table>
<thead>
<tr>
<th></th>
<th>25M</th>
<th>50M</th>
<th>75M</th>
<th>100M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosto SPARQL queries / Jena TDB(^1)</td>
<td>3,121</td>
<td>7,308</td>
<td>10,622</td>
<td>15,763</td>
</tr>
<tr>
<td>R2R / LDIF(^1)</td>
<td>1,506</td>
<td>2,803</td>
<td>4,482</td>
<td>*5,718</td>
</tr>
<tr>
<td>SPARQL 1.1 / Jena TDB(^1)</td>
<td>2,720</td>
<td>6,418</td>
<td>10,481</td>
<td>16,548</td>
</tr>
<tr>
<td>R2R / LDIF(^2)</td>
<td>1,485</td>
<td>2,950</td>
<td>4,715</td>
<td>*5,784</td>
</tr>
<tr>
<td>SPARQL 1.1 / Jena TDB(^2)</td>
<td>2,839</td>
<td>6,508</td>
<td>12,386</td>
<td>19,499</td>
</tr>
<tr>
<td>SPARQL 1.1 / Jena TDB</td>
<td>2,925</td>
<td>6,858</td>
<td>12,774</td>
<td>20,630</td>
</tr>
</tbody>
</table>

* Hadoop version of LDIF as single node cluster. Out of memory for in-memory version.
\(^1\) without RCP, RCV and AGG mappings
\(^2\) without AGG mapping
Simple mapping patterns were predominant

- 62% renaming pattern (RC, RP)
- Also simple structural patterns
- And different kinds of value transformations

SPARQL 1.1 engines are able to express them all

SPARQL 1.0 engines can express only 9 out of 15
Thanks!

http://lodib.wbsg.de