Linked Data on the Web (LDOW 2012)

Benchmarking the Performance of Linked Data Translation Systems

Carlos R. Rivero[1], **Andreas Schultz**[2], Christian Bizer[2] and David Ruiz[1]

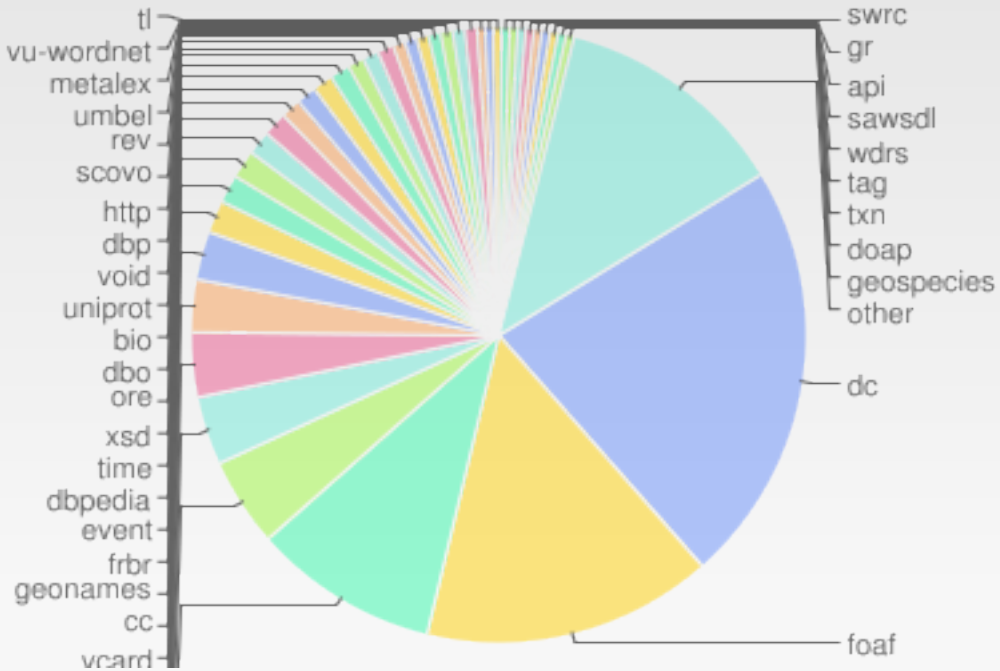
[1] University of Sevilla
[2] Freie Universität Berlin

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

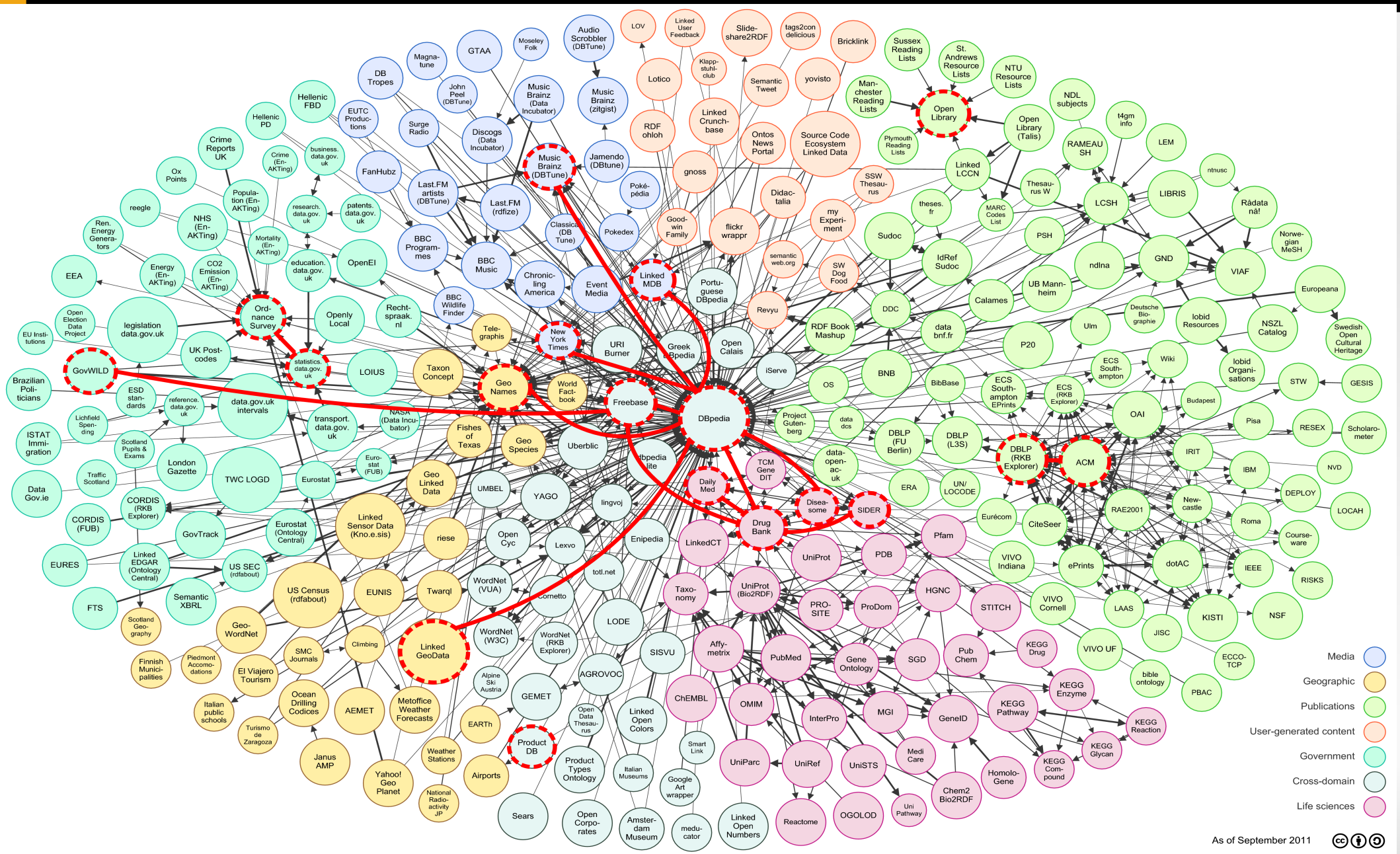
Data is represented...

- 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



Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/

Mapping Patterns

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

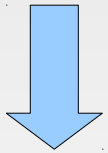
Code - Name	Source triples	Target triples
RC – Rename Class	?x a C_s	?x a C_t
RP – Rename Property	?x P_s ?y	?x P_t ?y
RCP – RC based on Property	?x a C_s . EXISTS {?x P ?y}	?x a C_t
RCV – RC based on Value	?x a C_s . ?x P v	?x a C_t
RvP – Reverse Property	?x P_s ?y	?y P_t ?x
Rsc – Resourcesify	?x P_s ?y	?x Q ?z . ?z P_t ?y
DRsc – Deresourcesify	?x Q ?z . ?z P_s ?y	?x P_t ?y
1:1 – Value Transformation 1:1	?x P_s ?y	?x P_t f(?y)
VtU – Value to URI	?x P_s ?y	?x P_t URI(?y)
UtV – URI to Value	?x P_s ?y	?x P_t LITERAL(?y)
CD – Change Datatype	?x P_s ?y^^DT_s	?x P_t ?y^^DT_t
ALT – Add Language Tag	?x P_s ?y	?x P_t ?y@TAG
RLT – Remove Language Tag	?x P_s ?y@TAG	?x P_t ?y
N:1 – Value Transformation N:1	?x P_1 ?v_1 … ?x P_n ?v_n	?x P_t f(?v_1, ... , ?v_n)
Agg – Aggregate	?x P_s ?y	?x P_t AGG(?y)

Structural Mapping Patterns

RCP - Rename Class based on Property

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 .
```

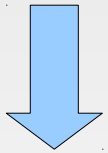


```
dbpedia:William_Shakespeare
     a  fb:people.deceased_person .
```

RCV - Rename Class based on Value

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 .
```

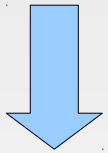


```
gw-p:Kurt_Joachim_Lauk_euParliament_1840_P ;
                a fb:government.politician .
```

Rsc - Resourcesify

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 .
```

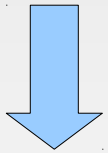
Value Transformation based Mapping Patterns

1:1 - Transform Value 1:1

Transform the value of a data type property.

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

Runtime in seconds



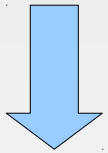
Runtime in minutes

```
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 .
```

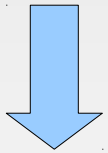


```
dbpedia:Von_Willebrand_disease
  diseasome:omim  <http://bio2rdf.org/omim:193400> .
```

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

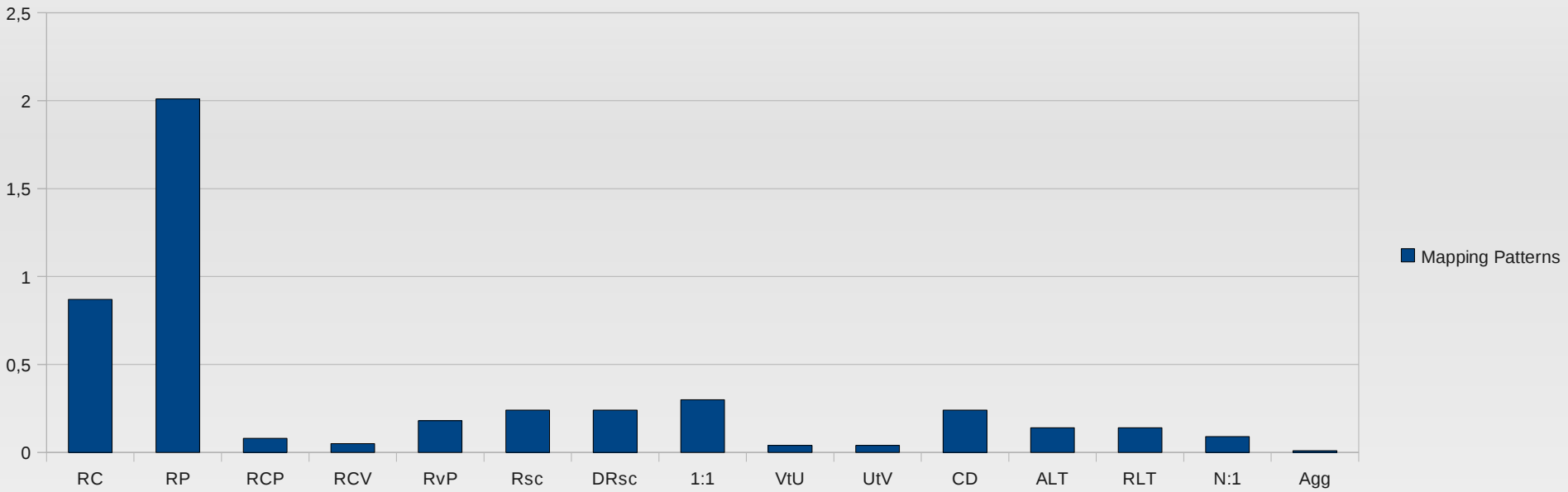
LODIB Grounding

- 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)

How We Counted the Mapping Patterns

- For all examples $(i_1$ 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

Average Pattern Occurrences



- 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

Measured Dimensions

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

	RC	RP	RCP	RCV	RvP	Rsc	DRsc	1:1	VtU	UtV	CD	ALT	RLT	N:1	Agg
Mosto queries	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
SPARQL 1.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
R2R	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

RCP: Rename Class based on Property
RCV: Rename Class based on Value
Agg: Aggregate

Results: Runtime Performance

Runtime results in seconds:

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

* *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*

Conclusion

- 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