LOD-a-lot:
A Queryable Dump of the LOD cloud

Javier D. Fernández, WU Vienna, Austria
Wouter Beek, VU University Amsterdam, The Netherlands
Miguel A. Martinez-Prieto, University of Valladolid, Spain
Mario Arias, Mario Arias Software, UK
The greatness of Linked Open Data

The Semantic Web

WE DID IT!!

> 150B triples
1K-6K datasets
>557 SPARQL Endpoints

http://lod-cloud.net/
https://datahub.io
http://stats.lod2.eu/
http://sparqles.ai.wu.ac.at
But what about Web-scale queries

- E.g. retrieve all entities in LOD with the label “Axel Polleres”

```sparql
select distinct ?x {
  ?x rdfs:label "Axel Polleres" .
}
```

- Solutions?
Let’s fish in our Linked Data Eco System
A) Federated Queries!!

1. Get a list of potential SPARQL Endpoints
   - datahub.io, LOV, other catalogs?
2. Query each SPARQL Endpoint

- Problems?
  - Many SPARQL Endpoints have low availability

http://sparqles.ai.wu.ac.at/
A) Federated Queries!!

1. Get a list of potential SPARQL Endpoints
   - datahub.io, LOV, other catalogs?
2. Query each SPARQL Endpoint

- Problems?
  - Many SPARQL Endpoints have low availability
  - SPARQL Endpoints are usually restricted (timeout, #results)
  - Moreover, it can be tricky with complex queries (joins) due to intermediary results, delays, etc
The Web of Data Eco System

B) Follow-your-nose
1. Follow self-descriptive IRIs and links
2. Filter the results you are interested in

- Problems?
  - You need some initial seed
    - DBpedia could be a good start
  - It’s slow (fetching many documents)
  - Where should I start for unbounded queries?
    - ?x rdfs:label “Axel Polleres”
The Web of Data Eco System

C) Use the RDF dumps by yourself

1. Crawl de Web of Data
   ▪ Probably start with datahub.io, LOV, other catalogs?

2. Download datasets
   ▪ You better have some free space in your machine

3. Index the datasets locally
   ▪ You better are patience and survive parsing errors

4. Query all datasets
   ▪ You better are alive by then

   ▪ Problems?
     ▪ Hugh resources!
     ▪ + Messiness of the data
A Linked Data hacker toolkit

1) LOD Laundromat

- Challenges:
  - Still you need to query 650K datasets
  - Of course it does not contain all LOD, but “a good approximation”
2) **HDT**

- Highly compact serialization of RDF
- Allows fast RDF retrieval in compressed space (without prior decompression)
  - Includes internal indexes to solve basic queries with small (3%) memory footprint.
    - Very fast on basic queries (triple patterns), x 1.5 faster than Virtuoso, Jena, RDF3X.
    - Supports FULL SPARQL as the compressed backend store of Jena, with an efficiency on the same scale as current more optimized solutions.

- **Challenges:**
  - Publisher has to pay a bit of overhead to convert the RDF dataset to HDT (but then it is ready to consume efficiently!)
3) Linked Data Fragments

- Challenges:
  - Still room for optimization for complex federated queries (delays, intermediate results, ...)

A Linked Data hacker toolkit
HDT – LDF – LOD Laundromat

consume

consume

hdt

hdt
Oh man I’m hungry and I don’t even know if I will like whatever you are cooking.
HDT – LDF – LOD Laundromat

Oh man I’m hungry and I don’t even know if I will like whatever you are cooking

consume

#LD
Linked Data Fragments

SPARQL endpoint
download
But what about Web-scale queries
But one could be really hungry

https://hwy55burgers.wordpress.com/tag/food-challenge/
LOD-a-lot

Linked Open Data

crawl

clean

index & store

Dataset 1

Dataset 650K

SPARQL endpoint (metadata)

LOD Laundromat

Dataset 1

Dataset 650K

N-Triples (zip)

N-Triples (zip)

28B triples

LOD-a-lot

Linked Data Fragments

hdt

#LD

Linked Data Fragments

hdt

#LD

Linked Data Fragments

hdt

#LD

Linked Data Fragments

hdt
LOD-a-lot (some numbers)

<table>
<thead>
<tr>
<th>#Triples</th>
<th>#Subjects</th>
<th>#Predicates</th>
<th>#Objects</th>
<th>#Literals</th>
</tr>
</thead>
<tbody>
<tr>
<td>28,362,198,927</td>
<td>3,214,347,198</td>
<td>1,168,932</td>
<td>3,178,409,386</td>
<td>1,302,285,394</td>
</tr>
</tbody>
</table>

- Disk size:
  - HDT: 304 GB
  - HDT-FoQ (additional indexes): 133 GB

- Memory footprint (to query):
  - 15.7 GB of RAM (3% of the size)
  - 144 seconds loading time
    - 8 cores (2.6 GHz), RAM 32 GB, SATA HDD on Ubuntu 14.04.5 LTS

- LDF page resolution in milliseconds.

(LOD-a-lot creation took 64 h & 170GB RAM. HDT-FoQ took 8 h & 250GB RAM)
LOD-a-lot

http://purl.org/HDT/lod-a-lot

https://datahub.io/dataset/lod-a-lot
LOD-a-lot (some use cases)

- Query resolution at Web scale
  - Using LDF, Jena

- Evaluation and Benchmarking
  - No excuse 😊

- RDF metrics and analytics

![Graphs showing distribution of subjects, predicates, and objects](image-url)
LOD-a-lot (some use cases)

- Identity closure (→ Wouter Beek)
  - ?x owl:sameAs ?y

- Graph navigations
  - E.g. shortest path, random walk

- Retrieve all entities in LOD with the label “Axel Polleres“

http://hdt.lod.labs.vu.nl/triple?object=%22Axel%20Polleres%22

More use cases:
Roadmap

- Update LOD-a-lot regularly
  - More and newer datasets from the LOD Cloud
- Keep named graphs with the provenance of each triple
  - Currently supported only via LOD Laundromat
- ... implement the use cases and help the community to democratize the access to LOD
  - low-cost access to LOD = high-impact research
ACKs
Thanks!

**LOD-a-lot**: A Queryable Dump of the LOD cloud

- [Javier D. Fernández](mailto:javier.fernandez@wu.ac.at), WU Vienna, Austria
- Wouter Beek, VU University Amsterdam, The Netherlands
- [Miguel A. Martinez-Prieto](mailto:miguel.martinez-prieto@univalle.es), University of Valladolid, Spain
- Mario Arias, Mario Arias Software, UK

[http://purl.org/HDT/lod-a-lot](http://purl.org/HDT/lod-a-lot)