1. Motivation

- The ultimate objective of Linked Data is linking and integration for enabling discovery and integrated query answering, and a big number of RDF datasets has already been published and this number keeps increasing.
- However, it is not currently evident how connected the LOD cloud is, only measurements between pairs of datasets are available. It is not possible to find the number of common URIs between 3 or more datasets.
- Measurements and indexes involving more than 2 datasets are important for:

2. Problem - Running Example

We focus on how to compute efficiently

- The datasets containing a particular (or equivalent) URI
- the number of common or equivalent URIs (i.e., same real world objects) in any subset

3. The Proposed Indexes

1. Prefix Index: An index which lists all namespaces and for each one what datasets contain them, e.g., see step 1 of Running Example.

2. SameAs Catalog: A catalog that computes the symmetric & transitive closure of owl:sameAs relationships. All the URIs that belong to the same class of equivalence (i.e., referring to the same entity) are getting the same signature, e.g., see step 2 of Running Example.

3. Element Index: For each real world object (i.e., URI or signature) appearing in two or more datasets, this index stores the datasets where it occurs (e.g., see step 3 of Running Example), by exploiting

   - SameAs Catalog for replacing a URI with its signature
   - Prefix Index for identifying the possible datasets where a URI occurs
   - ASK queries for checking if a URI exists at least in two datasets

4. The Lattice of Measurements

- A lattice is a partially ordered set which can be represented as a Directed Acyclic Graph (DAG) where the edges points towards the direct supersets.
- We compute the intersection of any set of datasets by making the measurements of the lattice incrementally: 
  - directCount(B): the frequency of subset B in the element index. (e.g., see steps 4 & 5 of Running Example)
  - Up(B): the supersets of B that can be found in directCount List (e.g., see step 6 of Running Example)
- The sum of the directCount of Up(B) gives the number of common real world objects in B.
- We propose two incremental algorithms that require only one index scan for computing the lattice (or a part of it) and exploit lattice and set theory properties.

- Top-Down approach using Breadth-First Search (BFS) starts from the maximum level (i.e., quad in our example).
- Then, it continues with the computation of the intersection of triads and finally of the pairs.
- Bottom-Up approach using Depth-First Search (DFS) starts by computing the intersection of a pair and continues upwards following a “Height First Search”.

5. Experimental Evaluation

- New connections thanks to the closure:
  - 19 millions of newly discovered owl:sameAs pairs!
  - 2,393 of newly discovered connected pairs of datasets!

- Measuring the current status of LOD:
  - DBpedia, Freebase and Yago share 2.7 millions of real world objects
  - Only 2.3% of real world objects exist in 3 or more datasets

- Time Efficiency
  - Signature-based algorithm (SBA) needs 45 seconds to compute the closure of 13 millions of owl:sameAs pairs and is faster than a common Connected Components algorithm (CC)
  - Index approach is faster than a straightforward (SF) method that performs binary search.
  - 1.5 billion of subsets intersections computed in 35 minutes with the bottom-up algorithm.
  - Incremental approaches are faster than methods (e.g., DCS) which do not exploit lattice & set theory properties. They need less than 10 seconds for over one million subsets.

6. Publishing and Exchanging Measurements

TRY LODsyndesis: www.ics.forth.gr/isl/LODsnydesis/
& FIND links to: datahub, a 3D visualization page, an active SPARQL Endpoint & a list of answerable queries.

Contact

Michalis Mountantonakis
Yannis Tzitzikas (coordinator)