SHER Scalable Highly Expressive Reasoner

Kavitha Srinivas, IBM Research
Outline

- Introduction to SHER
- SHER use cases
- Work in the pipeline
  - RDF in relational (is there hope :-))
  - RDF benchmarks (initial observations)
Brief overview of SHER

- Primarily aimed at Abox relations or instance data
- Built on top of Pellet (a DL reasoner)
- Supports expressive ontologies (OWL-DL minus nominals)
- Sound, complete
- Optimizations for EL+, optimization of query processing using query expansion (SSWS 2009).
Key to SHER scalability – Summarization

Apply DL rules on a smaller compact version of the instance graph

Original Instance Graph  Summary Graph

Great degree of compression on real datasets

- 2M text triples → 20 nodes (small Tbox)
- 60M EMR triples → 11K nodes (450K concepts Tbox)
Key to SHER scalability – Refinement

Based on query, selectively refine parts of the instance graph till you can conclude on the query.

Query: People taking courses taught by Professors

Refinements stops as soon as each all nodes in real subgraph mapped to the summary nodes share the same sets of edges. For complex real queries, 8-10 refinements
SHER optimizations

Incorporate sound, incomplete reasoning in first step

- Person
- Course
- Professor

Isolate answers to query out of the graph

Continue iterative refinement

Reduces refinement significantly
SHER Use cases

Abox reasoning is useful in some cases. E.g., matching patient data to clinical queries such as show me all patients on drugs with active ingredient of ibuprofen (ISWC 2007).

High cost to pay (query times 10-20 mins for complex queries, 60 million records)
SHER Use cases

- AnatomyLens (Semantic search of medical literature) – EL+ reasoning on Tboxes of GO and FMA (JWS, 2009) – scalable and usually what customers go for.


- Information sharing (application for security and access control) SACMAT 2009*

- Combination of probabilistic reasoning and OWL-DL reasoning.

- Personal observation – in many cases (outside of heathcare and defense, customers hesitant to invest resources in modeling complex ontologies.
Current and future work

• Data model alignment (how do we link and align data) including those with weak semantics.

• RDF stores/persistence:
  – RDF in relational databases (how far can we exploit existing DB2 infrastructure)
    • Compressed storage of RDF into a row
    • Initial results promising (comparable to native stores)
  – RDF benchmarks – How appropriate are they?
    • Some benchmarks are quite relational (LUBM approaches TPC)
    • Need benchmarks that better capture use of RDF data (unstructured, semi-structured)