A roadmap to
Efficient Query Answering over
big RDF data using Spark
through summaries

Giannis Agathangelos, Georgia Troullinou,
Haridimos Kondylakis, Dimitris Plexousakis

ISIP 2019, May 2019
Big Web Of Data

➢ Exponential growth of the web and the extended use of semantic technologies.
  ○ Enormous amount of widely available RDF datasets.
  ○ Explosive growth in data size.
The problems

- Enormous amount of widely available RDF datasets that are difficult to:
  1. Visualize & Comprehend
  2. Explore
  3. Store, manage, query
Ontology Summarization

- **Ontology Summarization** aspires to produce an abridged version of the original data source highlights its most important concepts reducing the size and the complexity.
Summarization Approaches

Input
- Nature
  - Schema-oriented
  - Instance-oriented
- Parameters
  - User-Specified
  - User-Independent

Method
- Quality
  - Complexity
- Availability
  - Open-Source
  - System/Tool
- Purpose
  - Indexing
  - Source Selection
  - Visualization

Output
- Type
  - Frequent structures
  - Graph
- Nature
  - Schema
  - Instance
  - Both

Non-Quotient
- Structural
- Statistical
- Pattern-mining
- Hybrid
- Rules
- Graph patterns
1. Visualize & Comprehend
We view an RDF dataset as two distinct and interconnected graphs: the schema ($G_s$) and the instance ($G_i$) graph.
Central questions to the process of summarization

1. How to identify the top-k nodes

2. How to link those nodes to produce a valid sub-schema graph.
Adapting Importance Measures (IM)

- Relevance
- Degree
- Betweenness
- Bridging Centrality
- Harmonic Centrality
- Radiality
- HITS
- PageRank

Adapted Important Measure: $AIM(u) = normal(IM(u)) + normal(#instances(u))$
Linking top-k important nodes

- Relevance maximization
- Coverage maximization
- Maximum Cost Spanning Tree

Given an undirected graph $G = (V, E)$, with edge weights $w: E \rightarrow \mathbb{R}^+$, find a spanning tree $T \subseteq G$ of maximum total edge cost, where $E_T \subseteq E$.

- Graph Steiner Tree

  - SDISTG
  - CHINS
  - HEUM

Given an undirected graph $G = (V, E)$, with edge weights $w: E \rightarrow \mathbb{R}^+$ and a node set of terminals $S \subseteq V$, find a minimum-weight tree $T \subseteq G$ such that $S \subseteq V_T$ and $E_T \subseteq E$.
2. Explore

RDFDigest+
http://rdfdigest.ics.forth.gr/
Zoom Operator

Zoom-in

Zoom-out
Extend Operator

**Dependence between two classes:**
- Infrequent connections between two classes are more informative than frequent ones
- Adapted Important Measures of the classes (AIM)
- Distance
A glimpse of Evaluation

- Dataset: **DBpedia** version 3.8
  - 359 classes, 1323 properties, 400M triples, more than 2.3M instances
  - 50K user queries from a specific period of time

- Evaluation Measure
  - **Query Coverage**: Assess the percentage of the queries that can be answered solely by using the generated schema summary along with the corresponding instances
Summaries Evaluation

Average Coverage for all Queries

Percentage of Zoom-out per Approximation Algorithm Used

- Random
- PageRank
- HITS
- Betweenness
3. Store, Manage, Query
Locality Aware Partitioning: Centroid Identification
Locality Aware Partitioning: Dependence
Locality Aware Partitioning: Schema Partitioning
Locality Aware Partitioning: Instance Partitioning

RDF Instance Triples

Associate Professor  Assistant Professor  Full Professor

Professor

Faculty  Course

Partition_1

Graduate Course
takesCourse

Person

Partition_2

Graduate Student

Partition_3

Work  Research

Publication

Institute

Manual

Article  Book

University  Program  Department

Organization

publicationResearch

orgPublication

Property name
Bounding Locality Aware Partitioning (BLAP)
LAWA Overview

Preprocessing

(B)LAP

RDF Dataset

Indexes

Class Index

Instance Index

SubGraph 1

SubGraph 2

SubGraph k-1

SubGraph k

HDFS

Query Processor

Input Query

Output
Datasets

➢ Real World Datasets & Workloads generated by FEASIBLE benchmark generator based on real log queries
  ○ DBpedia 3.8
    ■ 110 queries (star, complex) range in 1-5 tps
  ○ Semantic Web Dog Food (SWDF)
    ■ 270 queries (cartesian, 1-tp, star) range in 1-5 tps

➢ Synthetic Benchmark
  ○ Lehigh University Benchmark (LUBM) 1K
    ■ 14 queries (Star, Chain, Complex) range in 1-6 tps

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Num. Triples</th>
<th>Storage Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWDF</td>
<td>304,583</td>
<td>49,2M</td>
</tr>
<tr>
<td>LUBM1K</td>
<td>13,405,454</td>
<td>2,3G</td>
</tr>
<tr>
<td>DBpedia</td>
<td>176,593,742</td>
<td>24,7G</td>
</tr>
</tbody>
</table>
Competitors & Execution Environment

- **SPARQLGX** (Vertical Partitioning)
  - Creates one partition for each predicate.

- **S2RDF** (Extended Vertical Partitioning)
  - Aims on data access reduction

- Cluster of 4 physical machines
  - 400 GB of storage
  - 235 GB of memory
  - 38 cores
  - Apache Spark 2.3.2
Query Execution - DBpedia

Outperforms by 1 order of magnitude SPARQLX
Query Execution – SWDF

Outperforms by
1 order of magnitude
S2RDF

2 orders of magnitude
SPARQLGX
Conclusion

➢ Summaries are ideal for RDF/S KB understanding, exploration, partitioning and querying

➢ Next Steps
  ➢ Machine Learning to identify top-k nodes
  ➢ Diverse summaries (or even fair)
  ➢ Incrementally updating summaries on data update
  ➢ Process streaming RDF data through summaries
  ➢ Directly go to schema-less datasets
  ➢ Extend our indexing techniques with the use of statistics
    ➢ e.g selecting first the partitions that can answer bigger query fragments
  ➢ Implement advanced statistics-based reordering on query execution to improve query performance
<table>
<thead>
<tr>
<th>The Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haridimos Kondylakis, FORTH</td>
</tr>
<tr>
<td>Dimitris Plexousakis, FORTH</td>
</tr>
<tr>
<td>Kostas Stefanidis Tampere Univ, FL</td>
</tr>
<tr>
<td>Georgia Troulinou, FORTH</td>
</tr>
<tr>
<td>Georgia Trouli, Tech. Univ. Crete</td>
</tr>
<tr>
<td>Dimitris Kotzinos Université de Cergy-Pontoise, FR</td>
</tr>
<tr>
<td>Giannis Agathangelos, Amadeus, FR</td>
</tr>
<tr>
<td>Alexandros Pappas, Market Logic, DE</td>
</tr>
<tr>
<td>Ioana Manolescu INRIA, FR</td>
</tr>
<tr>
<td>Evangelia Daskalaki FORTH</td>
</tr>
<tr>
<td>Giannis Roussakis, FORTH</td>
</tr>
<tr>
<td>Giorgos Flouris, FORTH</td>
</tr>
</tbody>
</table>