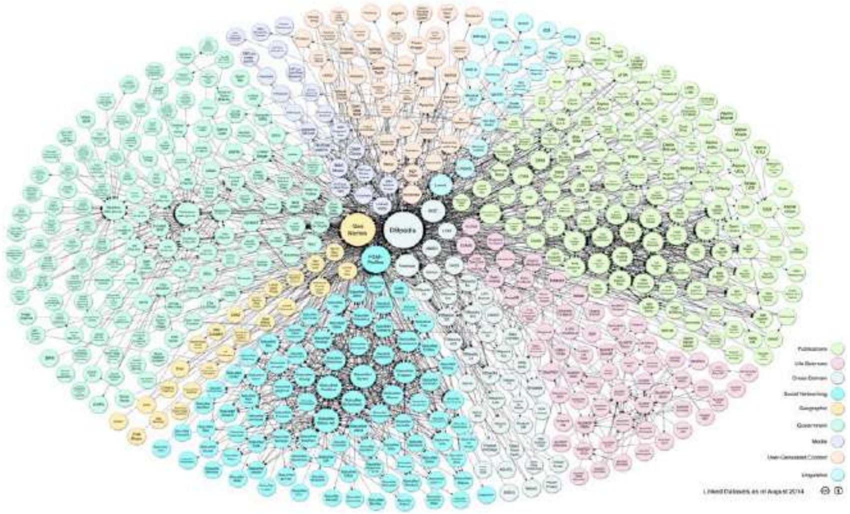
A roadmap to Efficient Query Answering over big RDF data using Spark through summaries

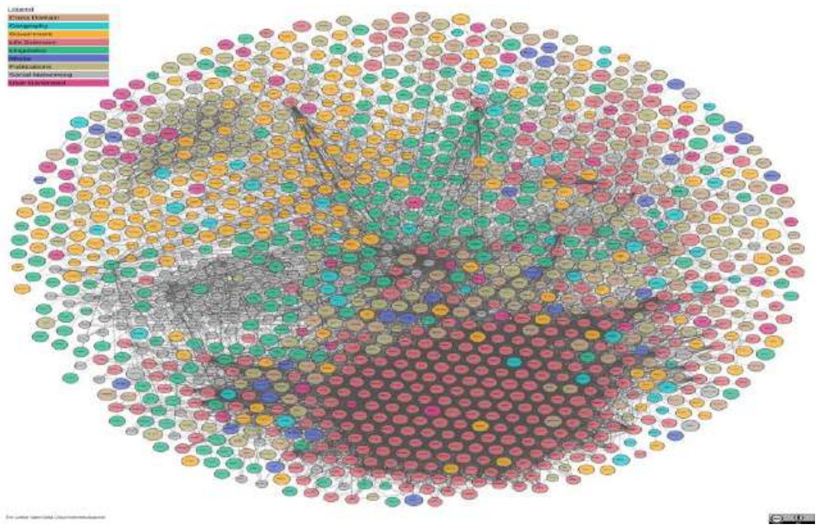
Giannis Agathangelos, Georgia Troullinou,
Haridimos Kondylakis, Dimitris Plexousakis

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.

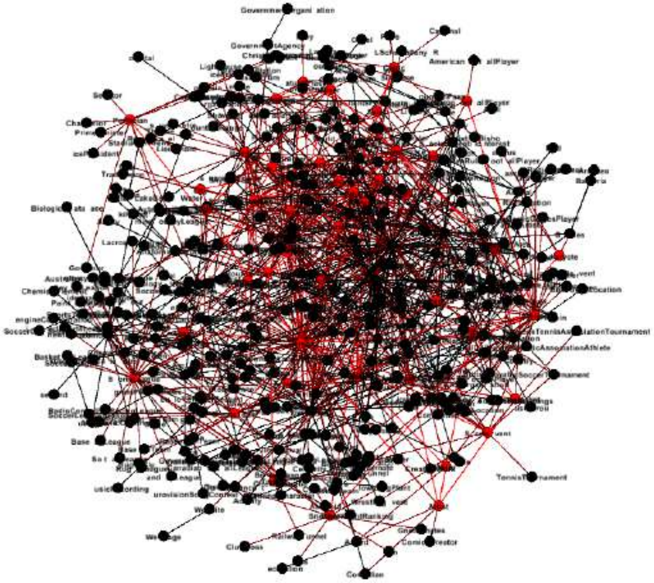




The problems

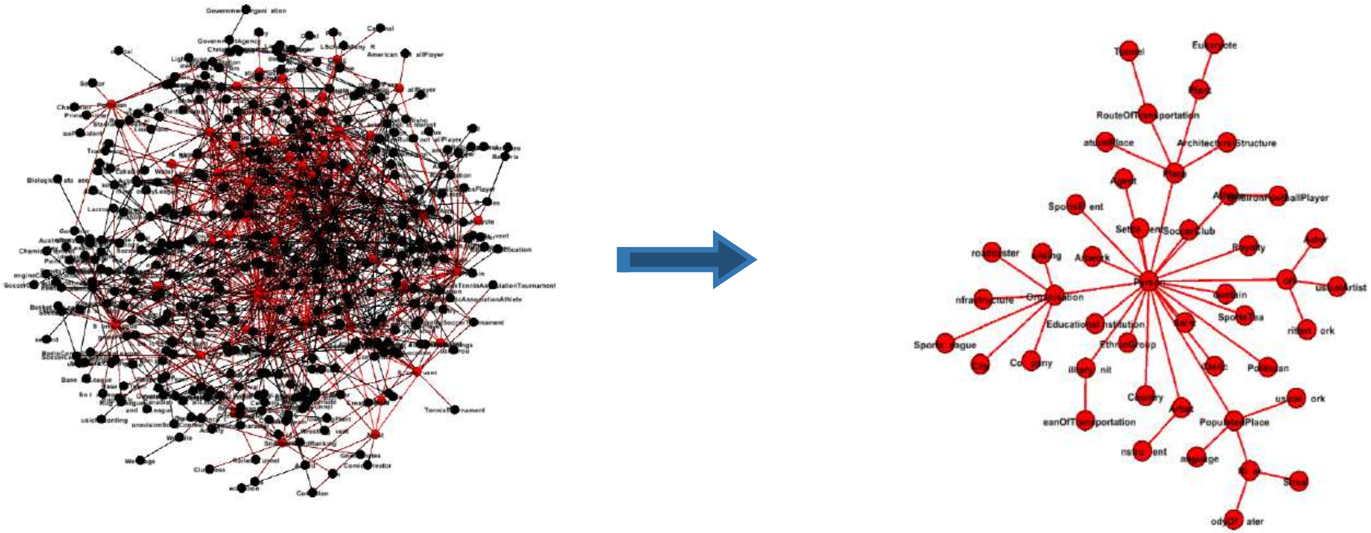
➢ Enormous amount of widely available RDF datasets that are difficult to:

1. Visualize & Comprehend
2. Explore
3. Store, manage, query



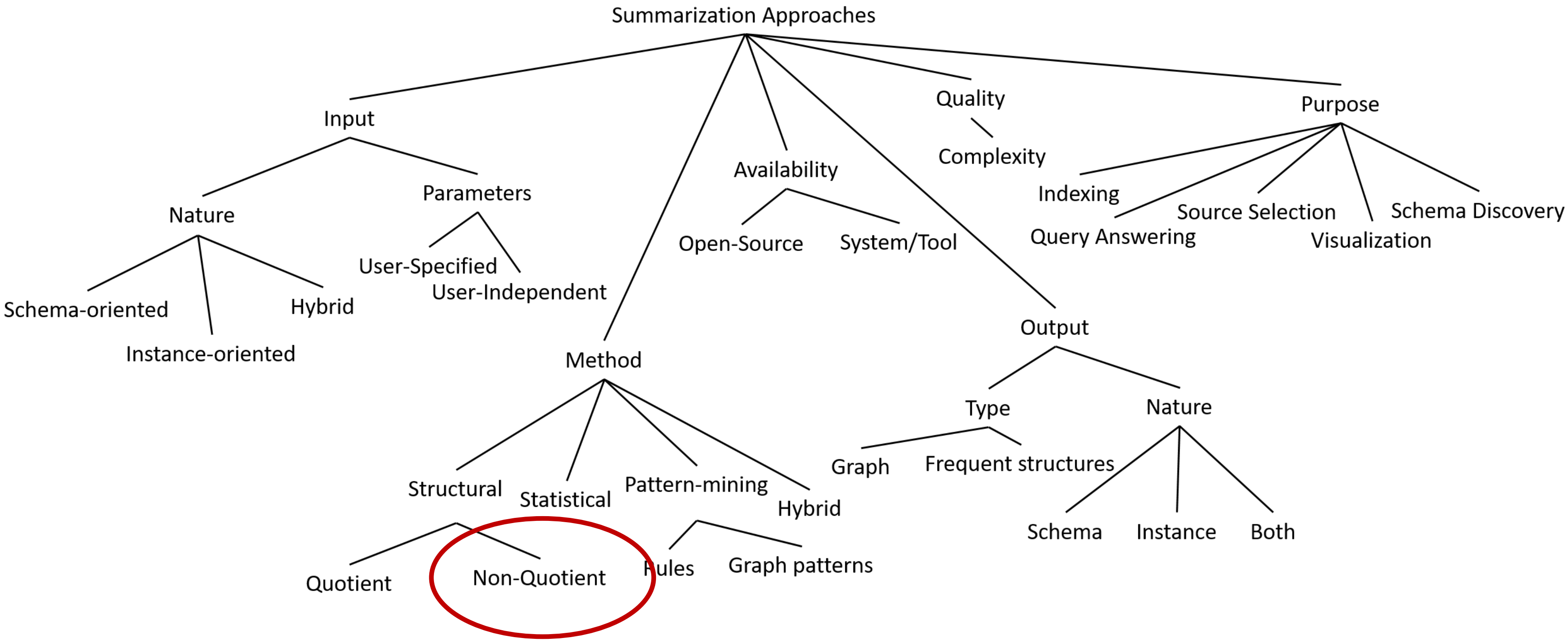
Ontology Summarization

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



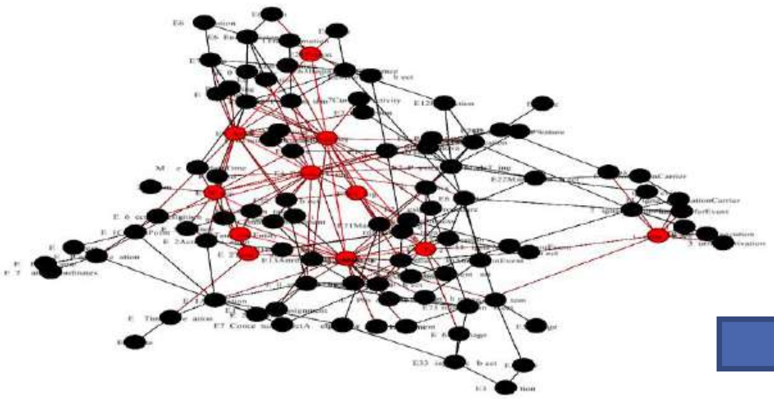
[EDBT, 2019]
[VLDB J, 2019]

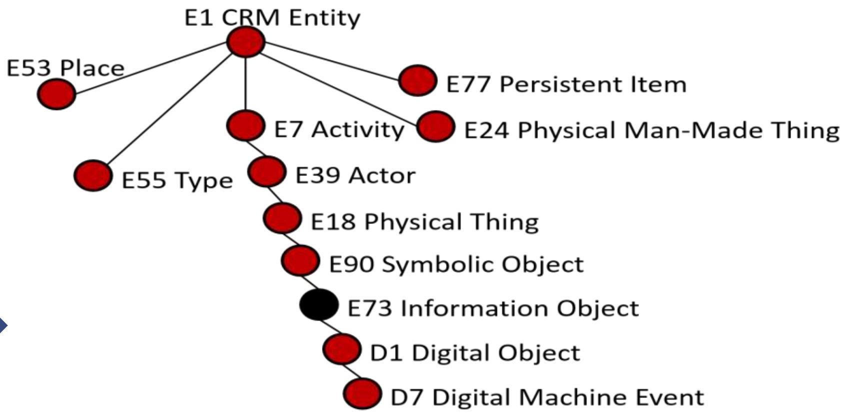
Summarization Approaches



[ESWC 2017]
[Sem. Web J. 2017]
[EDBT 2016]
[ESWC 2015]
[ISWC 2015]

1. Visualize & Comprehend

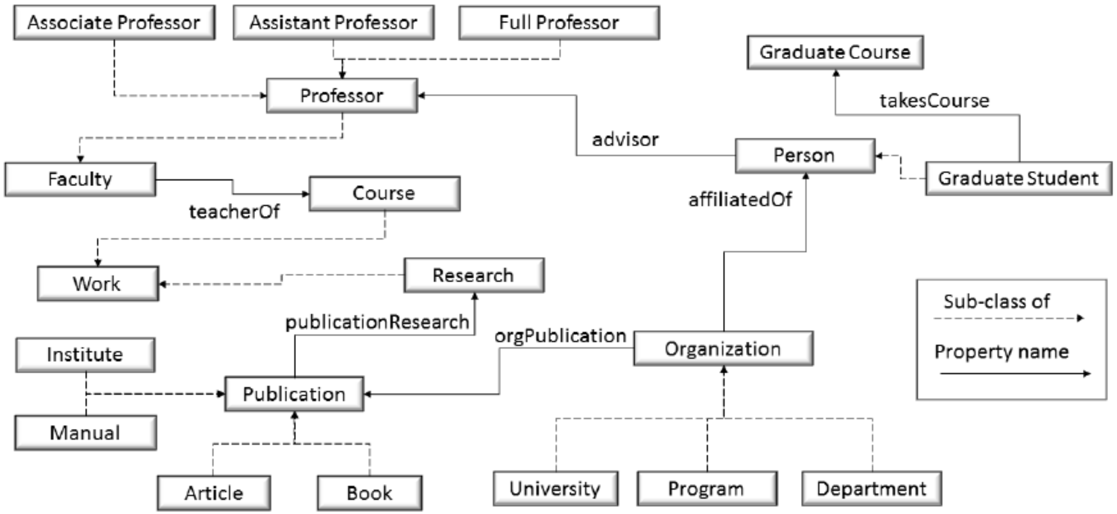




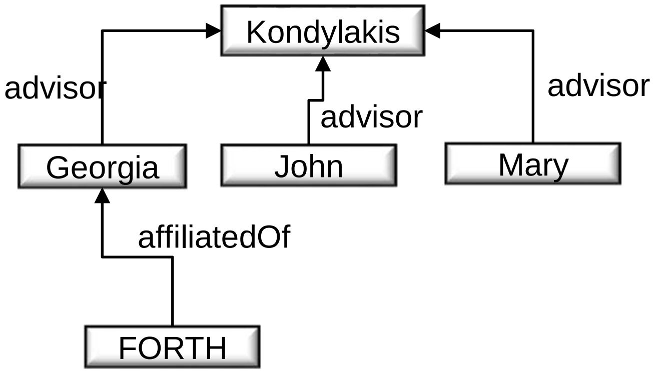
RDF Dataset

- We view an RDF dataset as two distinct and interconnected graphs: the **schema** (G_S) and the **instance**(G_I) graph.

G_S :schema graph



G_I :instance graph



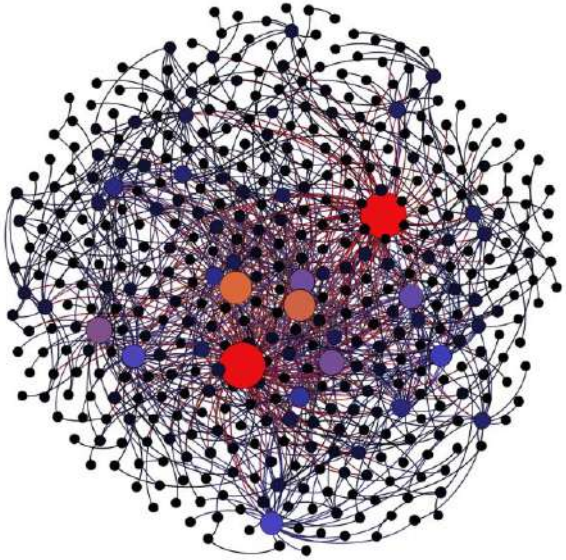
Central questions to the process of summarization

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
- Betweeness
- Bridging Centrality
- Harmonic Centrality
- Radiality
- HITS
- PageRank



$$\text{Adapted } Important\ Measure\text{: } AIM(u) = normal\big(IM(u)\big) + normal(\#instances(u))$$

Linking top-k important nodes

Algorithm	Weighted graph	Un-weighted graph
MST	$O(E \cdot logV)$	$O(\|V + E\|)$
SDISTG	$O(Q \cdot \|V\, logV\|)$	$O(Q \cdot \|V + E\|)$
CHINS	$O(Q \cdot \|V\, logV\|)$	$O(Q \cdot \|V + E\|)$
HEUM	$O(V \cdot \|V\, logV\|)$	$O(V \cdot \|V + E\|)$

- Relevance maximization
- Coverage maximization

- Maximum Cost Spanning Tree

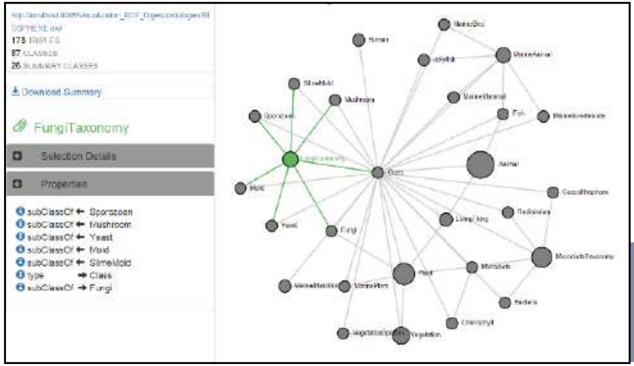
Given an undirected graph $G = (V, E)$, with edge weights $w: E \rightarrow R^{+}$
find a spanning tree $T \in G$ of maximum total edge cost, where $E_t \subseteq E$.

- Graph Steiner Tree
    - SDISTG
    - CHINS
    - HEUM

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

[ISWC 2018]
[ISWC+ 2018]

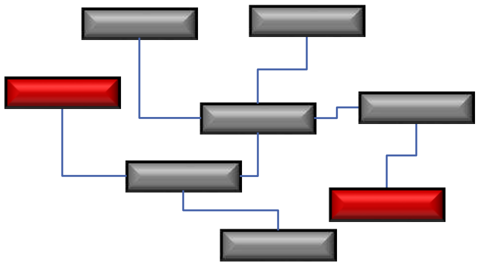
2. Explore



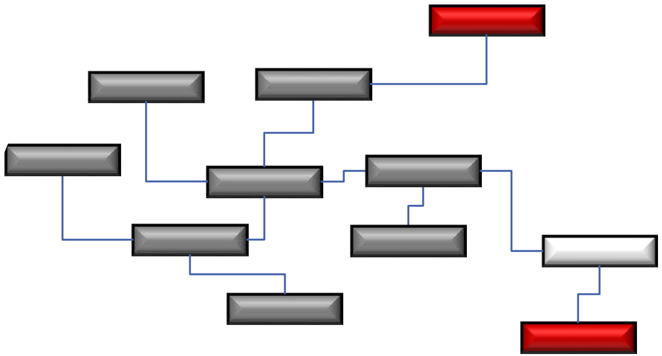
RDFDigest+
http://rdfdigest.ics.forth.gr/

Zoom Operator

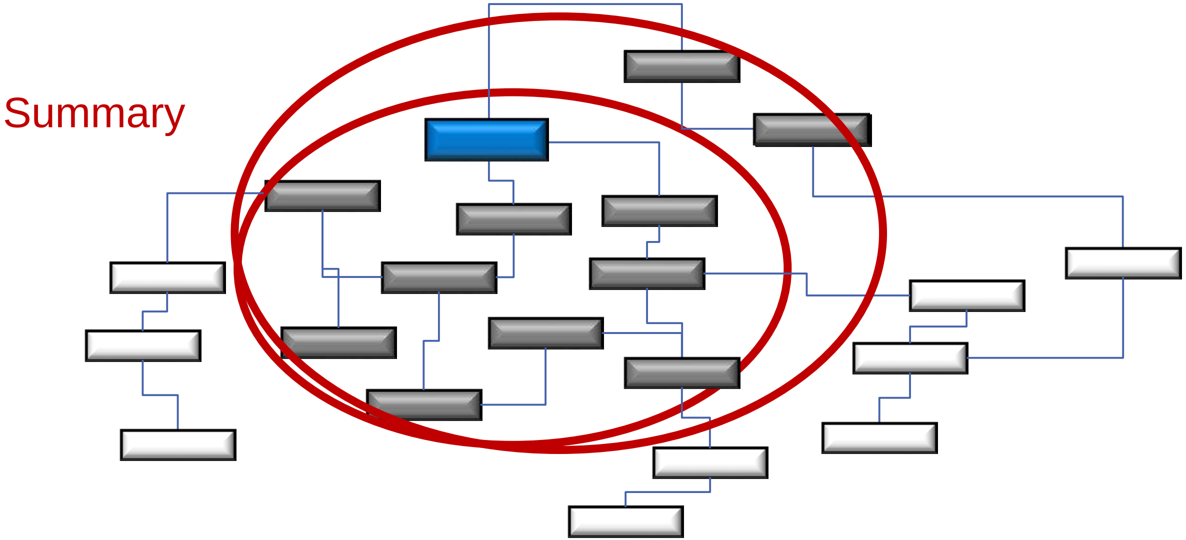
Zoom-in



Zoom-out



Extend Operator



Dependence between two classes:

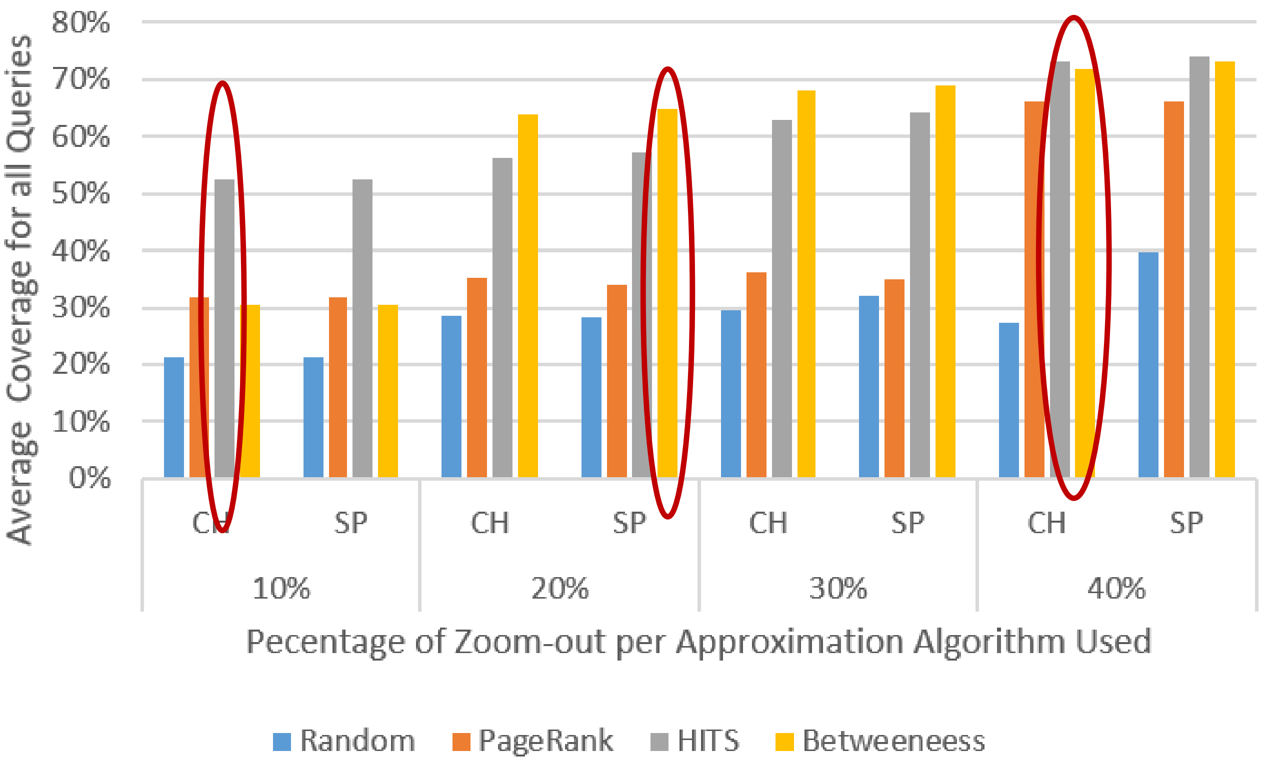
- Infrequent connections between two classes are more informative than frequent ones
- Adapted Important Measures of the classes (AIM)
- Distance

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

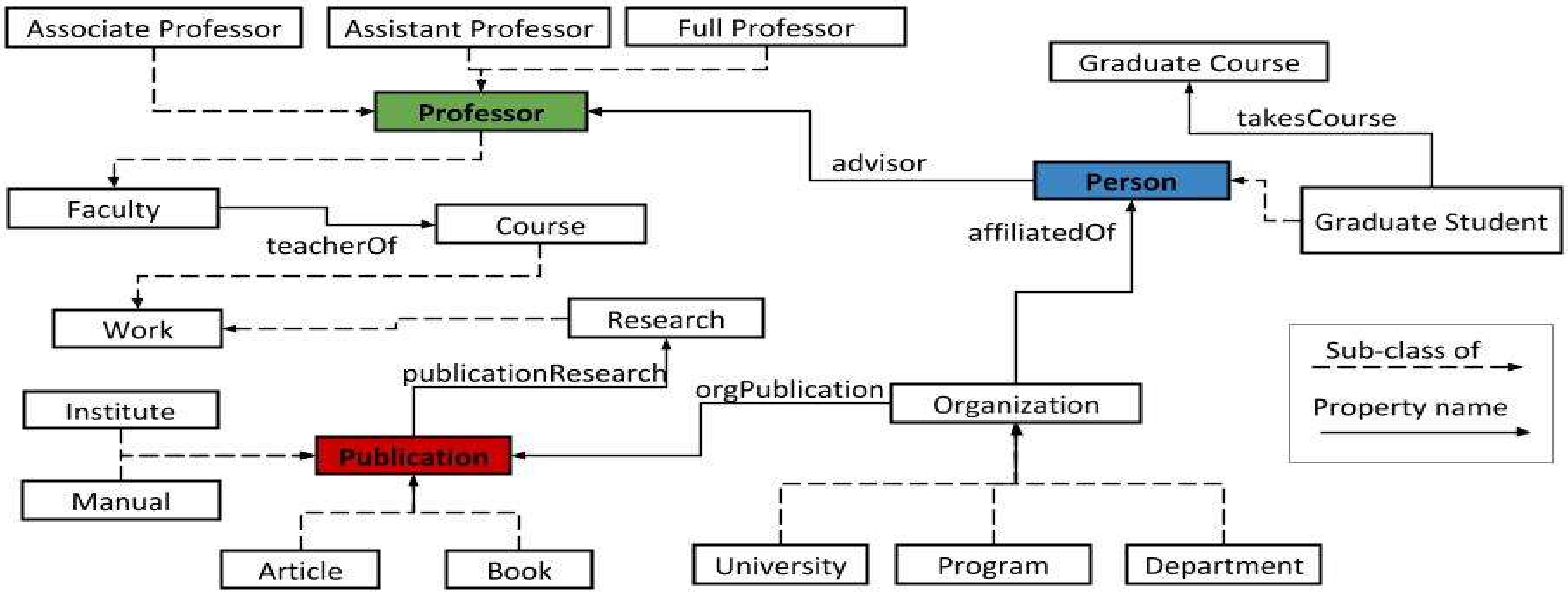


[ESWC 2018]
[ICDE 2018]

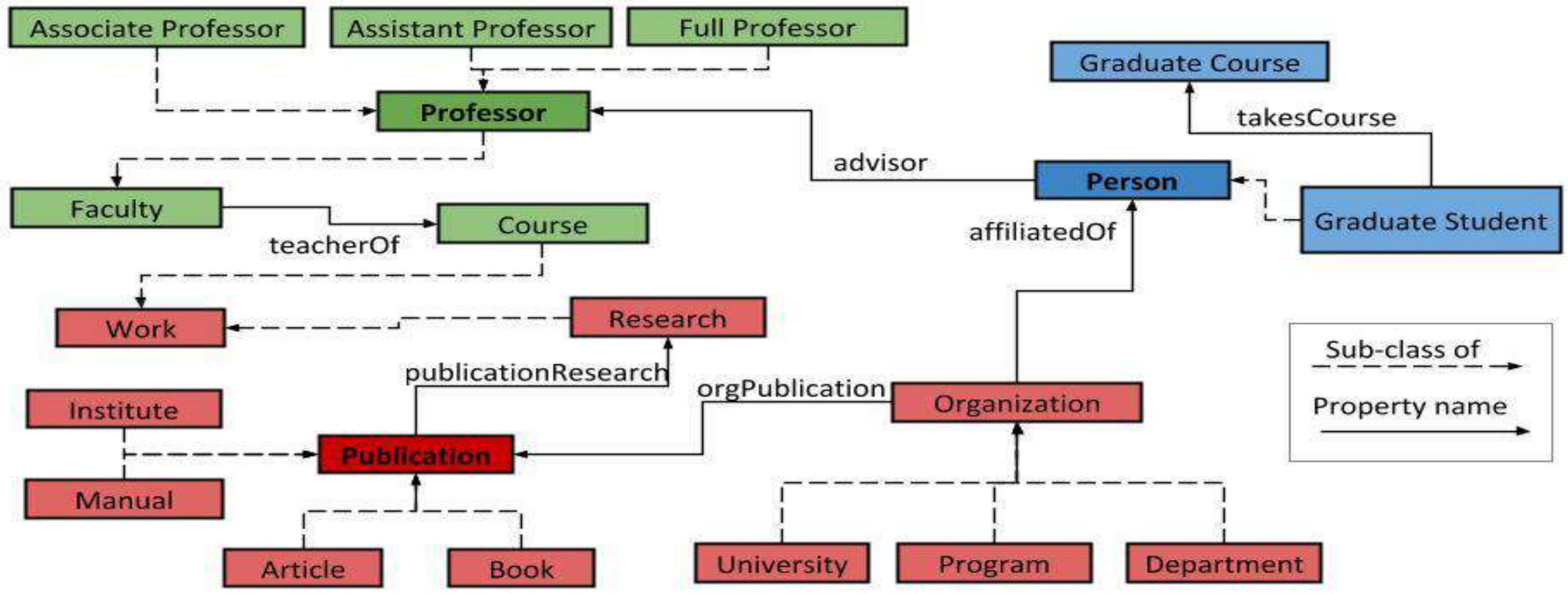
3. Store, Manage, Query

LAWA

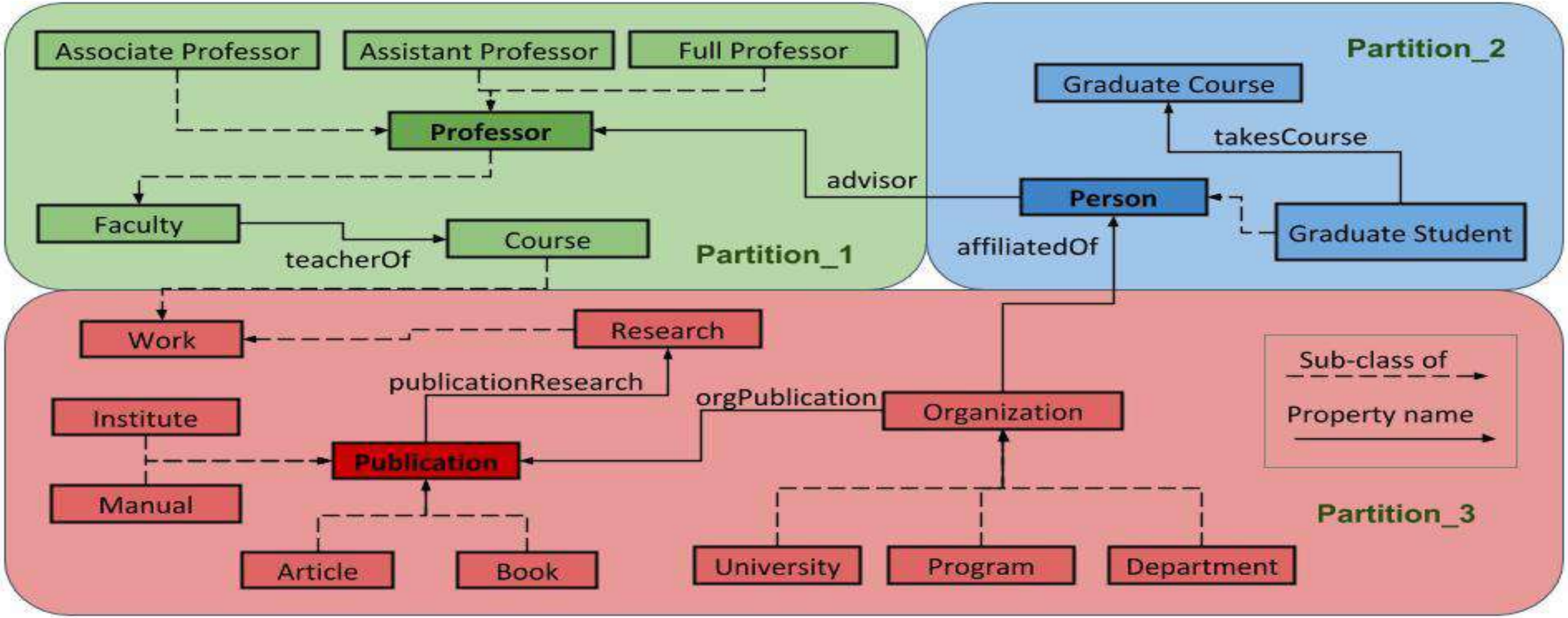
Locality Aware Partitioning: Centroid Identification



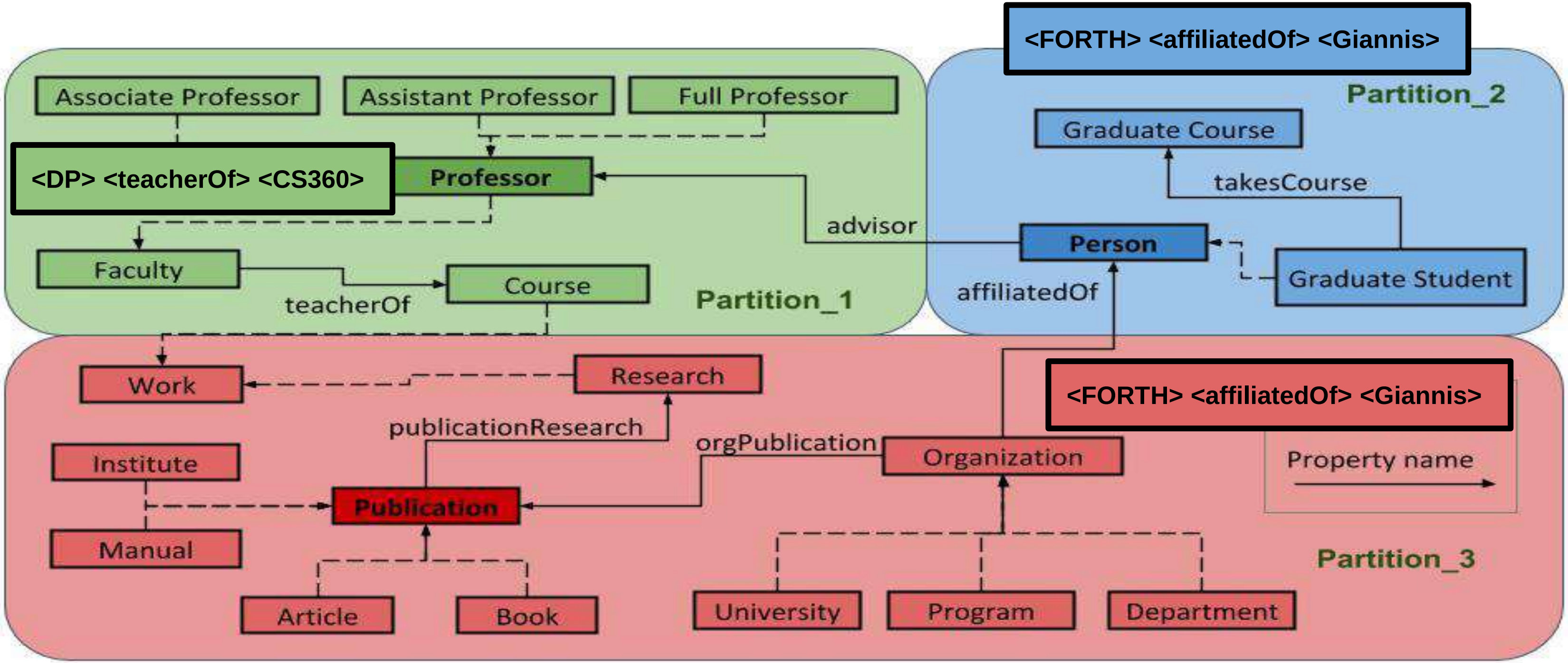
Locality Aware Partitioning: Dependence



Locality Aware Partitioning: Schema Partitioning



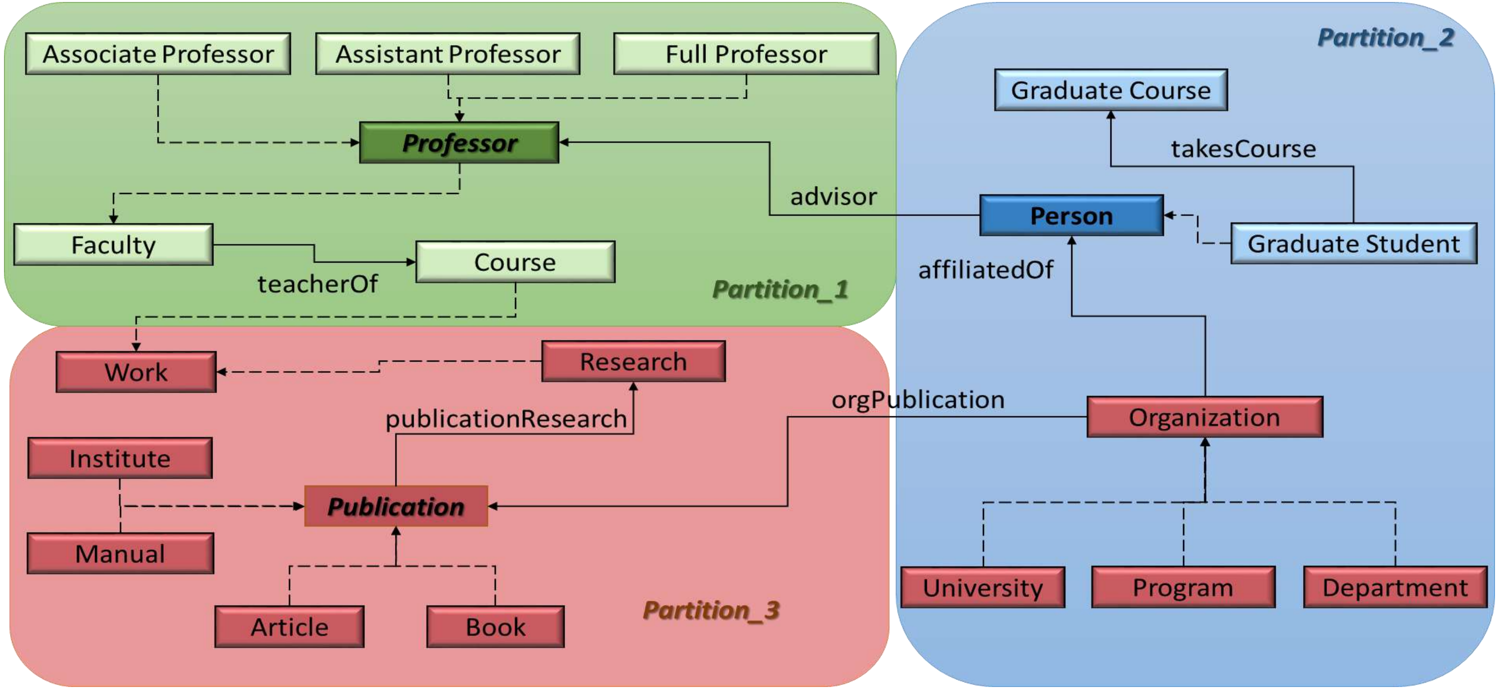
Locality Aware Partitioning: Instance Partitioning



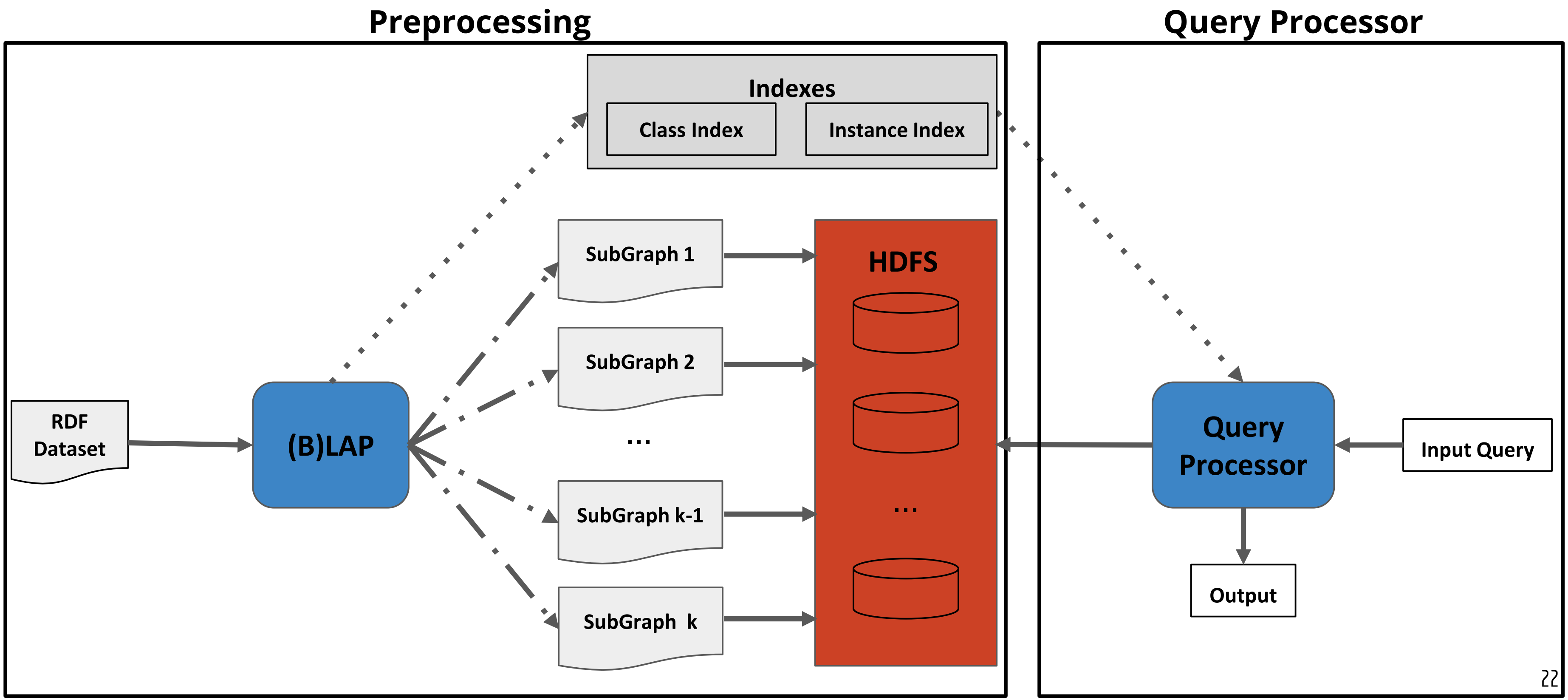
RDF Instance Triples

<FORTH> <affiliatedOf> <Giannis>	<DP> <teacherOf> <CS360>

Bounding Locality Aware Partitioning (BLAP)



LAWA Overview



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**

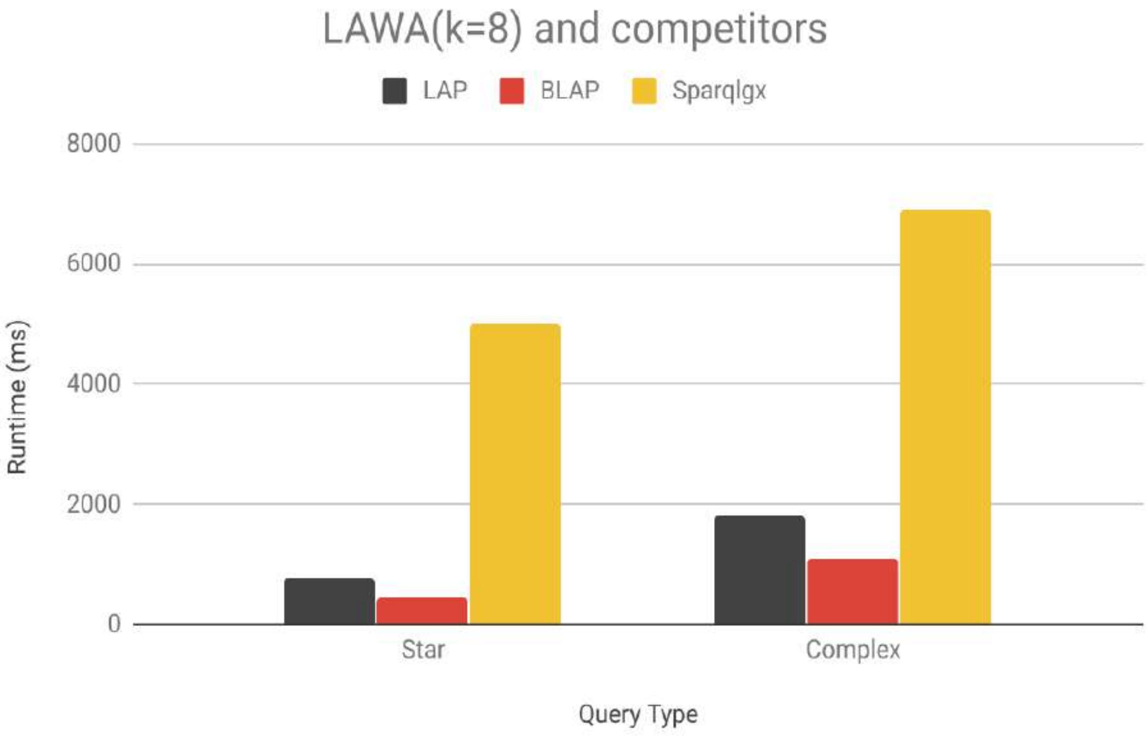
Dataset	Num. Triples	Storage Size
SWDF	304,583	49,2M
LUBM1K	13,405,454	2,3G
DBpedia	176,593,742	24,7G

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

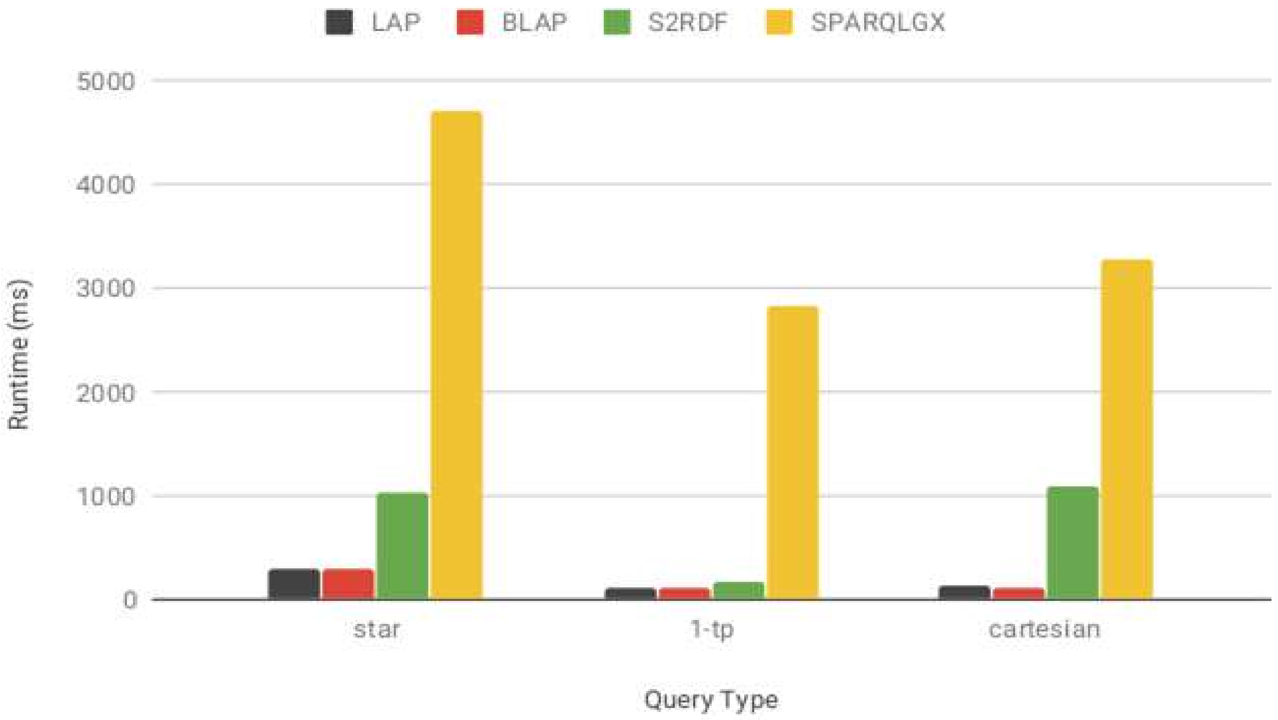
Outperforms by
1 order of magnitude
SPARQLX



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**

The Team

Haridimos
Kondylakis,
FORTH

Dimitris
Plexousakis,
FORTH

Kostas
Stefanidis
Tampere Univ, FL

Georgia
Troulinou,
FORTH

Georgia
Trouli,
Tech. Univ. Crete

Dimitris
Kotzinos
Université de
Cergy-Pontoise, FR

Giannis
Agathangelos,
Amadeus, FR

Alexandros
Pappas,
Market Logic, DE

Ioana
Manolescu
INRIA, FR

Evangelia
Daskalaki
FORTH

Giannis
Roussakis,
FORTH

Giorgos
Flouris,
FORTH