

RDF graph summarization: principles, techniques and applications

Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu

FORTH-ICS, University of Cergy-Pontoise, INRIA and Ecole Polytechnique

EDBT Conference, March 26-29, 2019

Outline

- 1 **Introduction & Motivation** (Ioana)
- 2 **Preliminaries: RDF & RDFS** (Ioana)
- 3 **Applications & Dimensions** (Haris)
- 4 **Generic (non-RDF) Summarization Approaches** (Ioana)
- 5 **Structural Summarization**
 - **Quotient RDF summarization** (Ioana)
 - **Non-quotient summarization** (Haris)
- 6 **Pattern-based RDF Summarization** (Dimitris)
- 7 **Statistical Summarization** (Dimitris)
- 8 **Other Summarization Methods** (Haris)
- 9 **Conclusions & Future Work** (Dimitris)

Part I

Motivation: data discovery in RDF
graphs

Big Data needs semantics

AI Magazine, Spring 2015



The image shows two side-by-side screenshots of the Data.gov website's search results page. Both screenshots show the 'DATA.CATALOG' header and navigation links for 'DATA', 'TOPICS', 'IMPACT', 'APPLICATIONS', 'DEVELOPERS', and 'CONTACT'. The left screenshot shows search results for 'Natural Disaster', displaying 93 datasets found. The right screenshot shows search results for 'Earthquakes', displaying 243 datasets found. Both pages include a map of the United States, a filter by location dropdown, and a list of dataset results with details such as dataset type, organization, and a 'Show More Dataset Type' link.

RDF summaries

Simplified views of an RDF graph [CGK⁺18]

- Most often, a summary is also a **graph**, and/or: **statistics**, **patterns**...
- Summarize: the **data** (structure and/or content), the **ontology**, **both**
- Many prior works on graph summarization (see also [LSDK18]) applied to RDF

RDF summaries

Simplified views of an RDF graph [CGK⁺18]

- Most often, a summary is also a **graph**, and/or: **statistics, patterns...**
- Summarize: the **data** (structure and/or content), the **ontology, both**
- Many prior works on graph summarization (see also [LSDK18]) applied to RDF

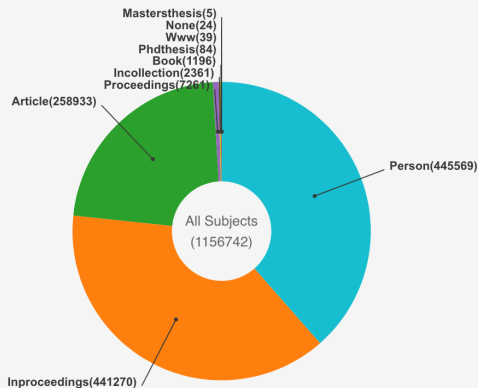
Summary uses:

- 1 For **query processing**: give direct access to a group of nodes summarized together, detect empty queries...
- 2 For **data discovery**: help identify interesting structure or patterns in the data

RDF graphs are often structurally heterogeneous

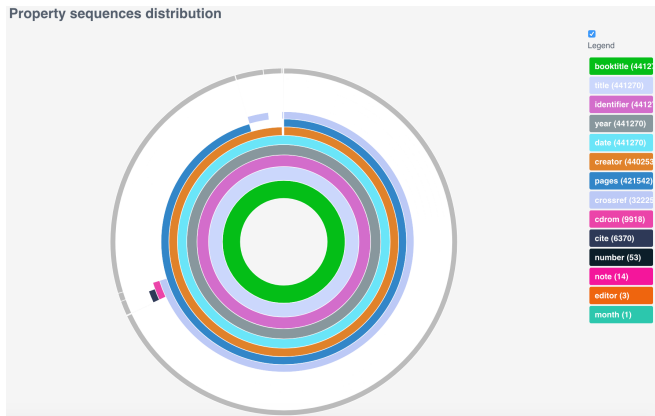
Subject types in DBLP bibliographic data:

Type distribution (Click *All Subjects* or a certain type below for further exploration.)



RDF graphs are often structurally heterogeneous

Data properties of DBLP conference articles:

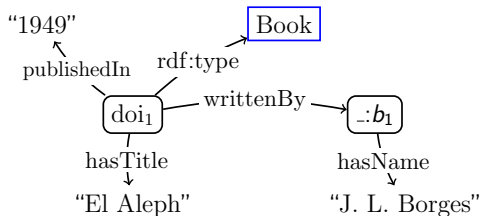
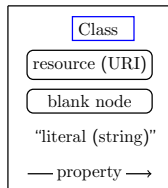


Part II

RDF and RDFS

The Resource Description Framework (RDF)

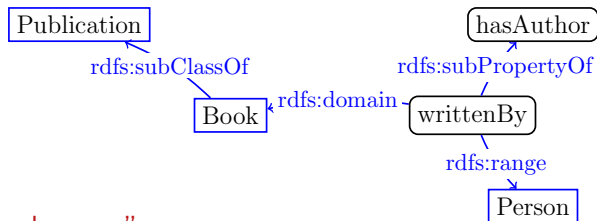
RDF graph: set of triples



RDF Schema

We consider **RDFS** deductive constraints, stating connections between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$

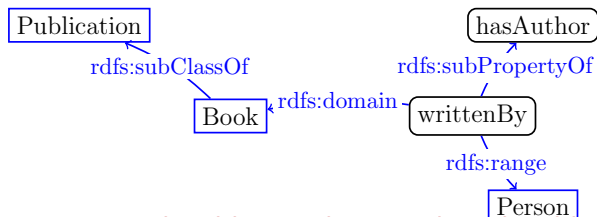


“Any c_1 is also a c_2 ”

RDF Schema

Simple language of deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$

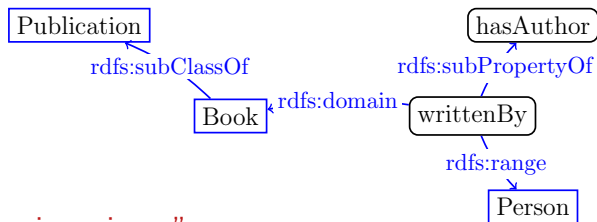


“If two resources are related by p_1 , they are also related by p_2 ”

RDF Schema

Simple language of deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$

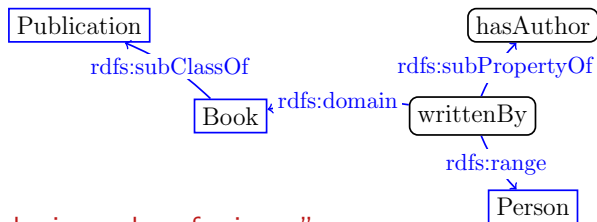


“Anyone having p is a c ”

RDF Schema

Simple language of deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$



“Anyone who is a value of p is a c ”

Open-world assumption and RDF entailment

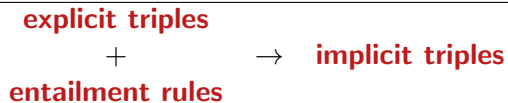
RDF data model based on the open-world assumption.

Deductive constraints lead to **implicit triples**:
part of the graph even though not explicitly present

Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

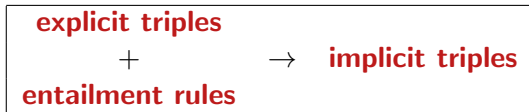
Deductive constraints lead to **implicit triples**:
part of the graph even though not explicitly present



Open-world assumption and RDF entailment

RDF data model based on the open-world assumption.

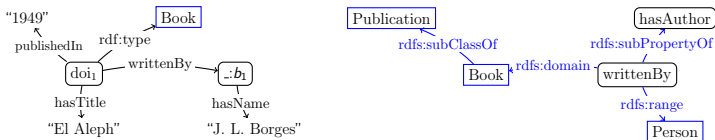
Deductive constraints lead to **implicit triples**:
part of the graph even though not explicitly present



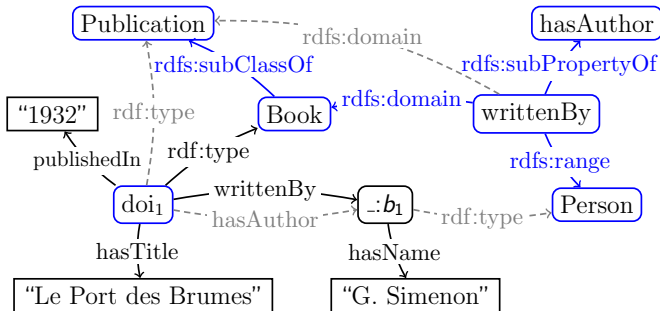
Exhaustive application of entailment leads to **saturation (closure)**

The semantics of an RDF graph G is its saturation G^∞

RDF data graph and RDF schema graph:



Saturation of the graph union:



Part III

Applications & Dimensions

RDF Graph Summaries and their use

Two **generic views** of an RDF summary:

- 1 A **compact information**, extracted from the original RDF graph
 - Summarization extracts meaning from data, while reducing its size
- 2 A **graph**, which some applications can **exploit instead of the original RDF graph**, to perform some tasks more efficiently.
 - The summary stands for the graph in specific settings.

Applications

- 1 **Indexing** - Lead directly to nodes used by the query.

Applications

- ① **Indexing** - Lead directly to nodes used by the query.
- ② **Estimating Query Cardinalities** - Compute how many nodes match certain query parts.

Applications

- 1 **Indexing** - Lead directly to nodes used by the query.
- 2 **Estimating Query Cardinalities** - Compute how many nodes match certain query parts.
- 3 **Making BGPs more specific** - Replace wildcard paths with specific ones.

Applications

- ① **Indexing** - Lead directly to nodes used by the query.
- ② **Estimating Query Cardinalities** - Compute how many nodes match certain query parts.
- ③ **Making BGPs more specific** - Replace wildcard paths with specific ones.
- ④ **Source selection** - Detect whether a graph is likely to contain a certain kind of data.

Applications

- 1 **Indexing** - Lead directly to nodes used by the query.
- 2 **Estimating Query Cardinalities** - Compute how many nodes match certain query parts.
- 3 **Making BGPs more specific** - Replace wildcard paths with specific ones.
- 4 **Source selection** - Detect whether a graph is likely to contain a certain kind of data.
- 5 **Graph visualization** - Support the user's discovery and exploration.

Applications

- 1 **Indexing** - Lead directly to nodes used by the query.
- 2 **Estimating Query Cardinalities** - Compute how many nodes match certain query parts.
- 3 **Making BGPs more specific** - Replace wildcard paths with specific ones.
- 4 **Source selection** - Detect whether a graph is likely to contain a certain kind of data.
- 5 **Graph visualization** - Support the user's discovery and exploration.
- 6 **Vocabulary usage analysis** - Based on actual ontology use, designers can make decisions about future versions.

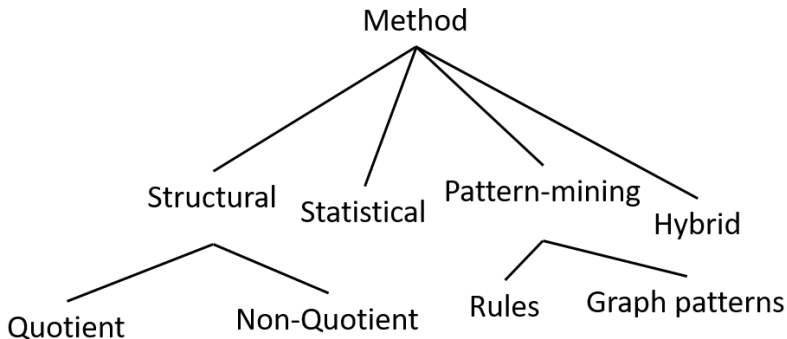
Applications

- 1 **Indexing** - Lead directly to nodes used by the query.
- 2 **Estimating Query Cardinalities** - Compute how many nodes match certain query parts.
- 3 **Making BGP's more specific** - Replace wildcard paths with specific ones.
- 4 **Source selection** - Detect whether a graph is likely to contain a certain kind of data.
- 5 **Graph visualization** - Support the user's discovery and exploration.
- 6 **Vocabulary usage analysis** - Based on actual ontology use, designers can make decisions about future versions.
- 7 **Schema (or ontology) discovery** - When an ontology is not present, it could be extracted from the graph.

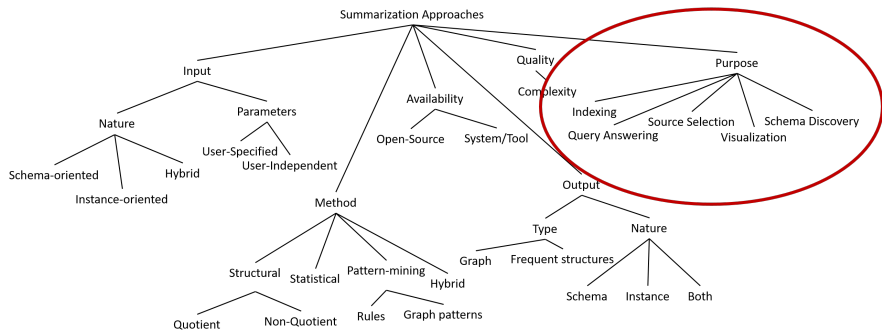
Other applications?

Can you suggest other applications?

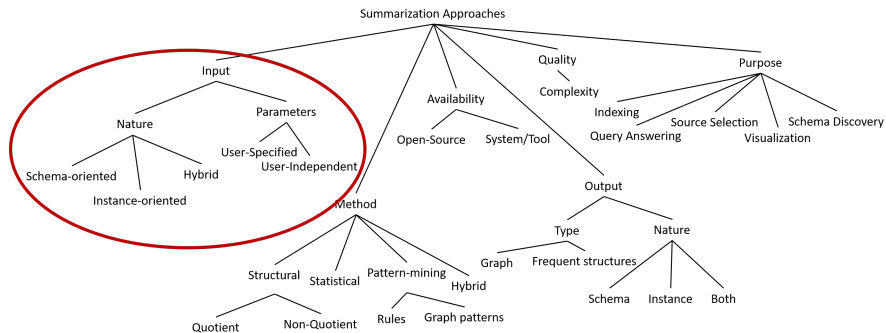
According to the summarization method



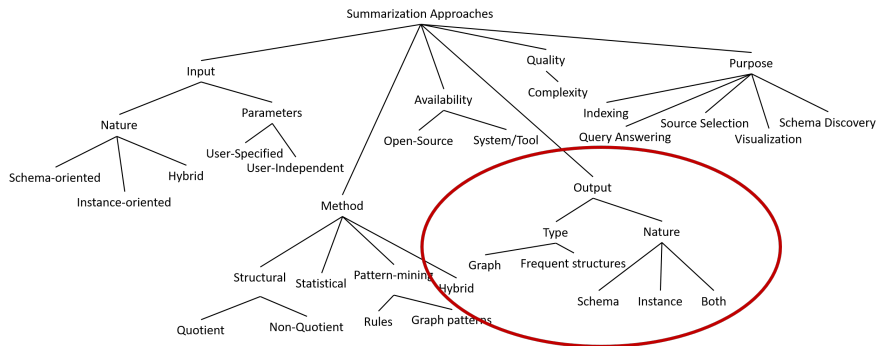
According to the summary purpose



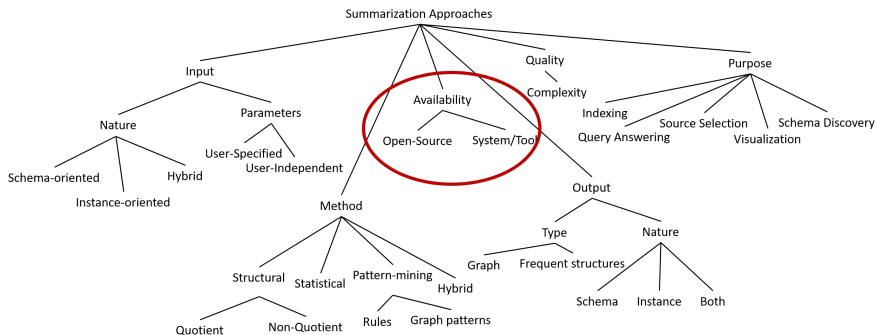
According to the summarization input



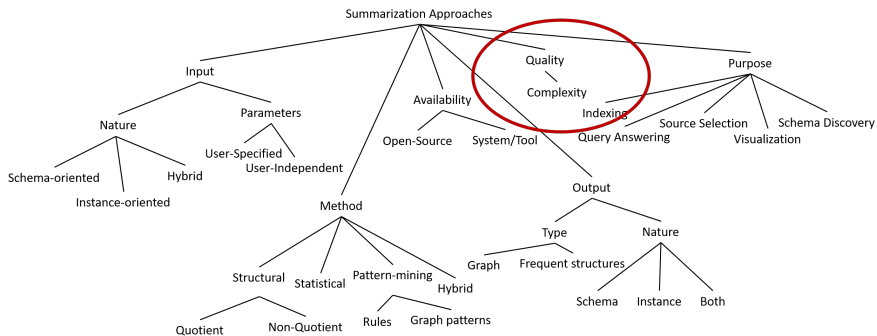
According to the summarization output



According to the availability of the summarization tool



According to the computational complexity



Part IV

Generic Graph Summarization

Summarization principle: quotient graphs

Let \equiv be an equivalence relation on the nodes of G .

The **quotient G_{\equiv} of a directed graph G by \equiv** is a graph defined as follows:

- G_{\equiv} nodes: one for \equiv equivalence class of V
- G_{\equiv} edges: $n_{\equiv}^1 \xrightarrow{a} n_{\equiv}^2$ iff $\exists n_1 \xrightarrow{a} n_2 \in G$ such that n_1 represented by n_{\equiv}^1 , n_2 represented by n_{\equiv}^2

Summarization principle: quotient graphs

Let \equiv be an equivalence relation on the nodes of G .

The **quotient G_{\equiv} of a directed graph G by \equiv** is a graph defined as follows:

- G_{\equiv} nodes: one for \equiv equivalence class of V
- G_{\equiv} edges: $n_{\equiv}^1 \xrightarrow{a} n_{\equiv}^2$ iff $\exists n_1 \xrightarrow{a} n_2 \in G$ such that n_1 represented by n_{\equiv}^1 , n_2 represented by n_{\equiv}^2

Quotients have interesting summary qualities:

- 1 **Property completeness:** All G properties appear in G_{\equiv}
- 2 **Size guarantees:** By definition, G_{\equiv} is at most as large as G (usually much smaller)
- 3 **Structure representativeness:** Given a query q , if its **structure-only** version is empty on G_{\equiv} , then q is empty on G

Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \sim_{fw} , \sim_{bw} , \sim_{fb}

Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \sim_{fw} , \sim_{bw} , \sim_{fb}

Problem: Bisimilarity compresses/summarizes very little!

Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \sim_{fw} , \sim_{bw} , \sim_{fb}

Problem: Bisimilarity compresses/summarizes very little!

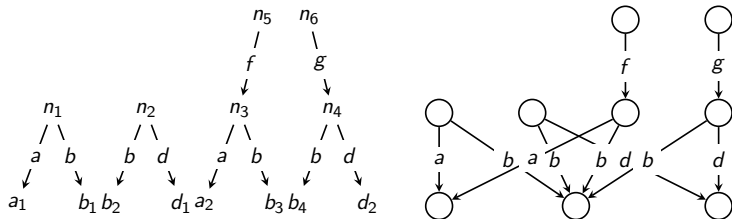
Solution: Bounded bisimilarity [KBNK02], e.g., \sim_{1fb}

Common graph quotients: bisimilarity [HHK95]

Two nodes are forward (resp. backward) bisimilar if they have exactly the same incoming (resp. outgoing) paths; \sim_{fw} , \sim_{bw} , \sim_{fb}

Problem: Bisimilarity compresses/summarizes very little!

Solution: Bounded bisimilarity [KBNK02], e.g., \sim_{1fb}



Still: > 130 property combinations on conf. papers in DBLP

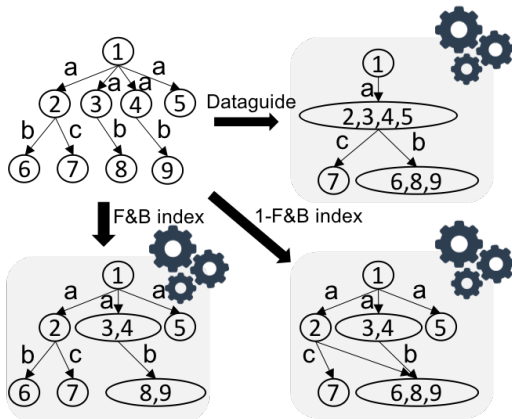
More on bisimilarity quotient summarization

- The simulation relation can be computed in $O(M * \log(M))$ [PT87] or $O(N * M)$ [HHK95]; linear for acyclic graphs.
- The Template Index (T-index) [MS99] is based on backward bisimilarity.
- [LFH⁺13] provides an I/O efficient external memory based algorithm for k -bisimulation. I/O complexity: $O(k * \text{sort}(M_p) + k * \text{scan}(N_p) + \text{sort}(N_p))$, where M_p, N_p are the numbers of pages storing graph edges (resp. nodes).

Non-quotient summarization

DataGuides [GW97]: two nodes are equivalent if they are **reachable by a common path**.

In general, this is **not** an equivalence relation (non transitive).
For trees only, coincides with backward bisimilarity.



Other graph summarization techniques

- SNAP [THP08] produces a structural summary based on properties of interest given by the users, also seeking same attribute values in a group
- k SNAP summarization [THP08, TP10] controls the number k of SNAP summary nodes, provides roll-up, drill-down
- [LTH⁺14] seeks to find super-nodes and super-edges such that each super-edge is an all-to-all connection between the respective data nodes; distributed implementation in Giraph

Closing remarks

Graphs are a natural, popular data model \Rightarrow numerous summarization methods proposed [LSDK18]

Database research has considered summarization especially for **indexing**

- Group (equivalence classes) or nodes determined in relation with a set of queries
- An index provides direct access to the **extent** of a group

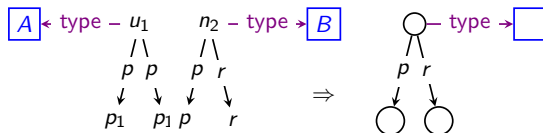
Main other summarization use case: **mining** for frequent structures in the graph

Part V

Structural RDF Summarization

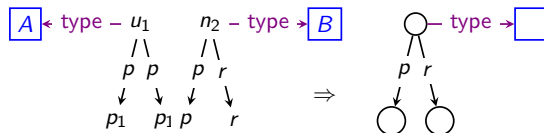
Can we summarize an RDF graph by a quotient?

Sample graph G and a possible quotient:



Can we summarize an RDF graph by a quotient?

Sample graph G and a possible quotient:

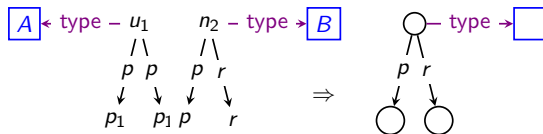


Possible loss of class and property names

Can we summarize an RDF graph by a quotient?

What about type and schema triples?

Sample graph G and a possible quotient:

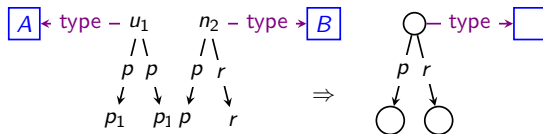


Possible loss of schema triples

Can we summarize an RDF graph by a quotient?

What about type and schema triples?

Sample graph G and a possible quotient thereof:



Possible loss of implicit triples

RDF equivalence relation and RDF summaries [ČGGM17]

Define:

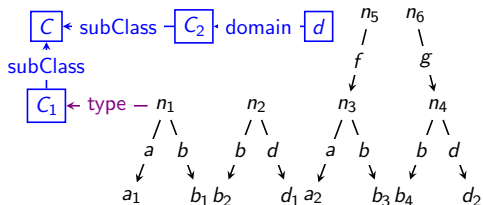
- 1 **RDF equivalence relation:** an equivalence relation on RDF graph nodes such that any class or property node is only equivalent to itself
- 2 **RDF summary:** a quotient of a graph G by an RDF equivalence relation such that any class or property node is represented by itself.

Consequence: For any RDF equivalence relation \equiv and RDF graph G , the schema of $G_{/\equiv}$ is the schema of G .

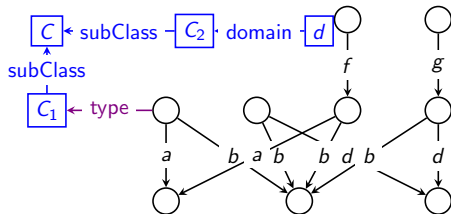
\Rightarrow No schema compression! (to be rediscussed briefly)

Summarization through an RDF equivalence relation

E.g., let \equiv_{1fb} to be the RDF node equivalence obtained from \sim_{1fb} .
Sample graph G :

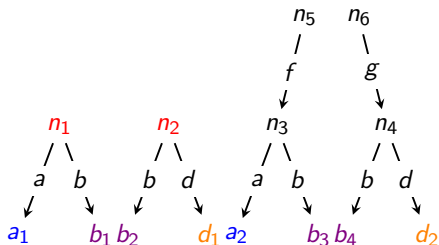


Its quotient through the RDF node equivalence \equiv_{1fb} :



RDF node equivalence based on property cliques [ČGM15, ČGGM17, GGM19]

Intuition: n_1, n_2 are “of the same kind”; similarly b_1, b_2, b_3

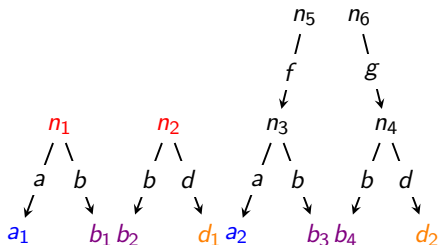


n_3, n_4 may or may not be of the same kind as n_1, n_2 .

RDF node equivalence based on property cliques

Output property cliques: $\{a, b, d\}$; $\{f\}$; $\{g\}$; \emptyset

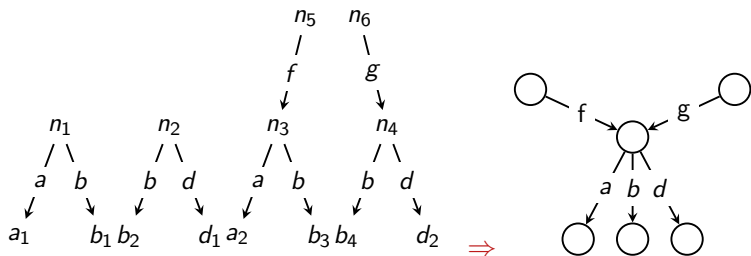
Input property cliques: $\{a\}$; $\{b\}$; $\{d\}$; $\{f\}$; $\{g\}$; \emptyset



Weak clique-based summaries

Two nodes are weakly equivalent ($\equiv_{/W}$) iff they have **the same input clique** **or** **the same output clique** **or** are weakly equivalent to a third one.

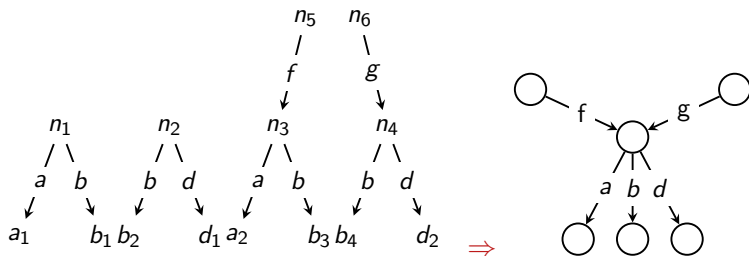
Weak summary $G_{/W}$ of the sample RDF graph G :



Weak clique-based summaries

Two nodes are weakly equivalent ($\equiv_{/W}$) iff they have **the same input clique** **or** **the same output clique** **or** are weakly equivalent to a third one.

Weak summary $G_{/W}$ of the sample RDF graph G :

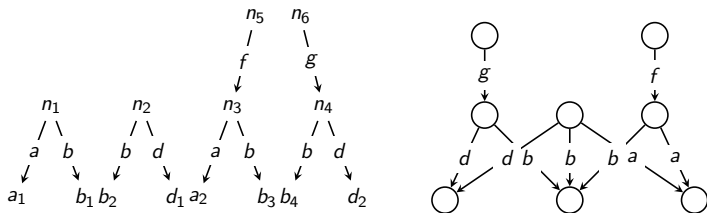


Property: In $G_{/W}$, each data property appears exactly once \Rightarrow its nodes are “source of p , target of p ” for each p [ČGM15].

Strong clique-based summaries

Two nodes are strongly equivalent (\equiv_S) iff they have **the same input clique** **and** **the same output clique**.

Strong summary $G_{/\equiv_S}$ of the same G :



Which role should node types play in summarization?

Having the same type(s) is orthogonal w.r.t. having the same structure.

Which role should node types play in summarization?

Having the same type(s) is orthogonal w.r.t. having the same structure. Two alternatives:

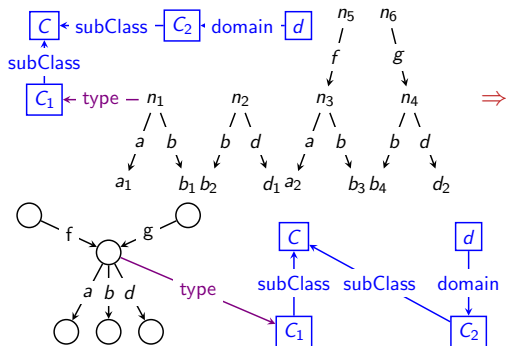
- 1 **Data-then-type:** group nodes first by their data triples, then carry the types from each \equiv group to its representative.

Which role should node types play in summarization?

Having the same type(s) is orthogonal w.r.t. having the same structure. Two alternatives:

- 1 **Data-then-type:** group nodes first by their data triples, then carry the types from each \equiv group to its representative.

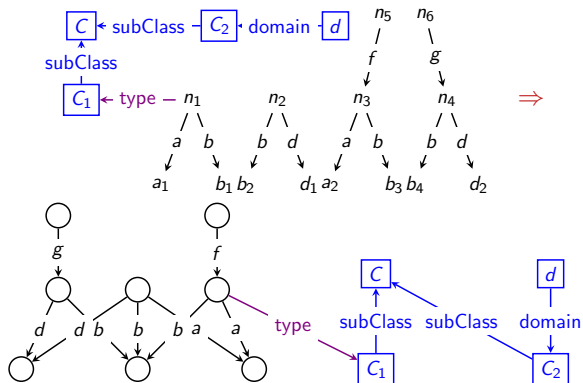
Extended Weak summary:



Adding types after data summarization

- Data-then-type:** group nodes first by their data triples, then carry the types from each \equiv group to its representative.

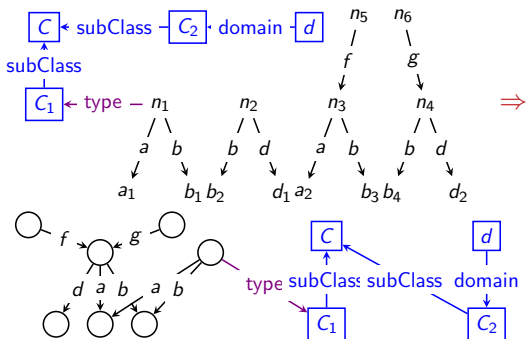
Extended Strong summary:



Giving prominence to types

- ② **Type-then-data:** Group nodes by their type set, and **untyped** nodes by their data properties.

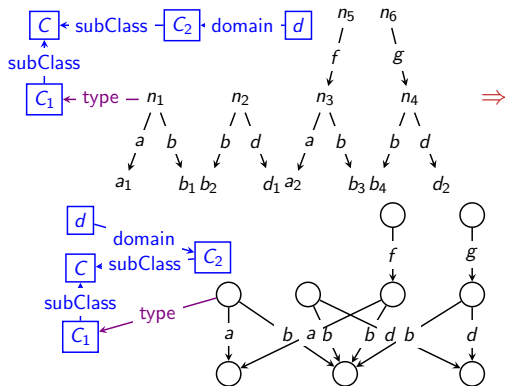
Typed Weak summary $G_{/\equiv TW}$ of the sample graph:



Giving proeminence to types

- Type-then-data:** Group nodes first by their types. Only untyped nodes are grouped by their data properties.

Typed Strong summary $G_{\equiv} \text{TS}$ of the sample graph:



Clique-based RDF summaries outline

Summary	Weak?	Strong?	Types first?
$G/\equiv W$	✓		
$G/\equiv S$		✓	
$G/\equiv TW$	✓		✓
$G/\equiv TS$		✓	✓

Quotient RDF summaries outline

Summary	Weak?	Strong?	FW bisim?	BW bisim?	Types first?
$G/\equiv W$	✓				
$G/\equiv S$		✓			
$G/\equiv TW$	✓				✓
$G/\equiv TS$		✓			✓
$G/\equiv fw$			✓		
$G/\equiv bw$				✓	
$G/\equiv fb$			✓	✓	
$G/\equiv fw, T$			✓		✓
$G/\equiv bw, T$				✓	✓
$G/\equiv fb, T$			✓	✓	✓

Summarizing the saturated graph G^∞

With an RDF Schema, the semantics of G is $G^\infty \Rightarrow$.

How to compute $(G^\infty)_{/\equiv}$?

- ① Saturate G , then summarize
- ② **Shortcut theorems** [ČGM17]
 - For the summaries $G_{/\equiv} \sqsubseteq W$, $G_{/\equiv} \sqsubseteq S$, $G_{/\equiv} \sqsubseteq fw$, $G_{/\equiv} \sqsubseteq bw$, $G_{/\equiv} \sqsubseteq fb$ [ČGGM17]:

$(G^\infty)_{/\equiv}$ is the same as $((G_{/\equiv})^\infty)_{/\equiv}$
 - **Sufficient condition** for any \equiv to admit the shortcut [ČGM17].

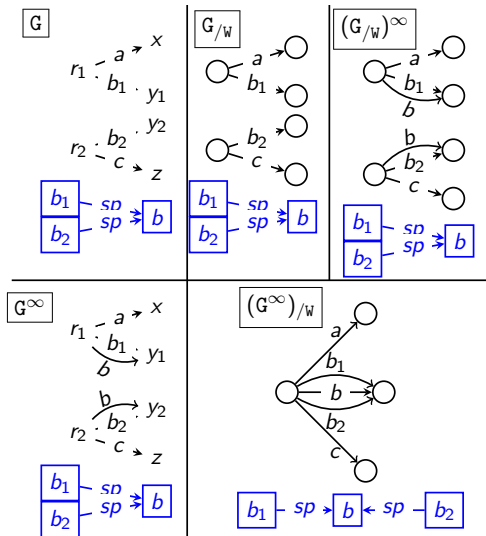
Shortcut path to G^∞

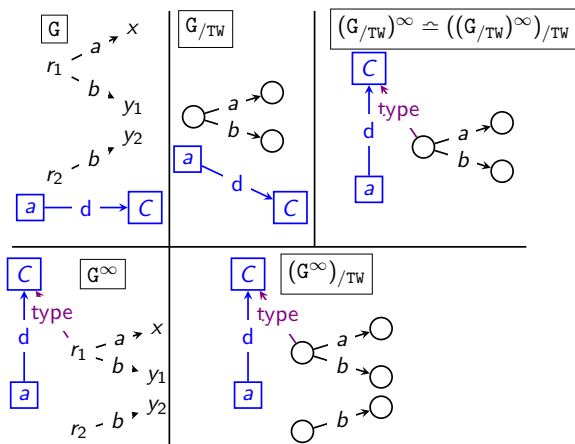
Direct $G \rightarrow \mathbf{sat.} \rightarrow G^\infty \rightarrow \mathbf{summ.} \rightarrow (G^\infty)_\equiv$

Shortcut $G \rightarrow \mathbf{summ.} \rightarrow G_\equiv \rightarrow \mathbf{sat.} \rightarrow (G_\equiv)^\infty \rightarrow \mathbf{summ.} \rightarrow ((G_\equiv)^\infty)_\equiv$

If G_\equiv is much smaller than G , **the shortcut may be faster!**

Up to 20 times in our experiments [ČGGM17]

Shortcut example: $G \not\equiv W$ 

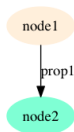
Shortcut counter-example: $G/\equiv TW$ 

Quotient summarization algorithms

- 1 **Global algorithms:** visit all G , compute \equiv relation, then traverse G again and represent each triple in G/\equiv
- 2 **Incremental algorithms:** visit G , compute \equiv and summary based on knowledge gained so far; **adjust** summary.
The challenge is to **simultaneously**:
 - Build the node equivalence relation \equiv
 - Represent nodes in the partial summary
 - Decisions may have to be undone: **split**, **merge**

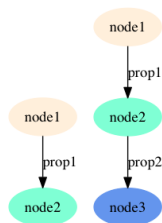
Example: weak incremental summarization (1) [GGM19]

Each color corresponds to a different \equiv_W class



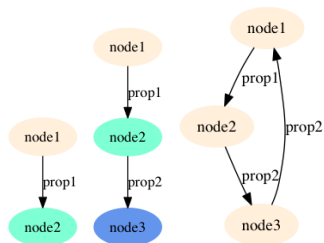
Example: weak incremental summarization (1) [GGM19]

Each color corresponds to a different \equiv_W class



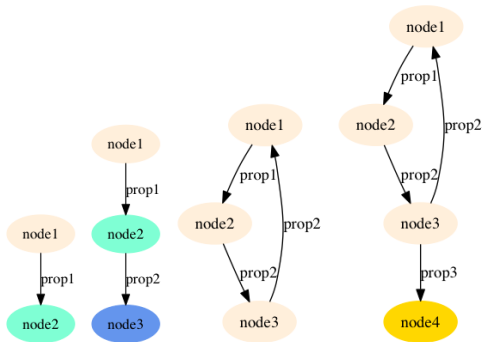
Example: weak incremental summarization (1) [GGM19]

Each color corresponds to a different \equiv_W class



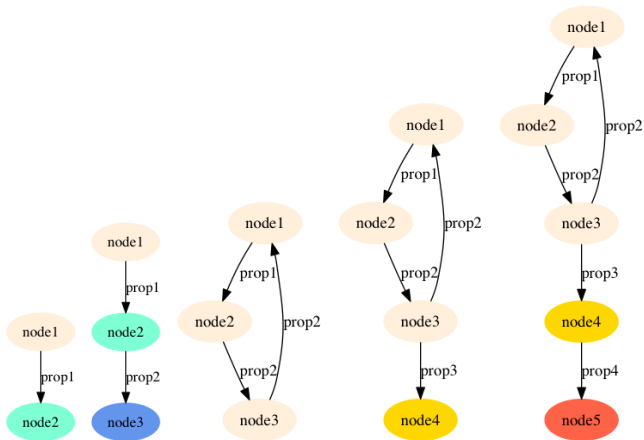
Example: weak incremental summarization (1) [GGM19]

Each color corresponds to a different \equiv_W class



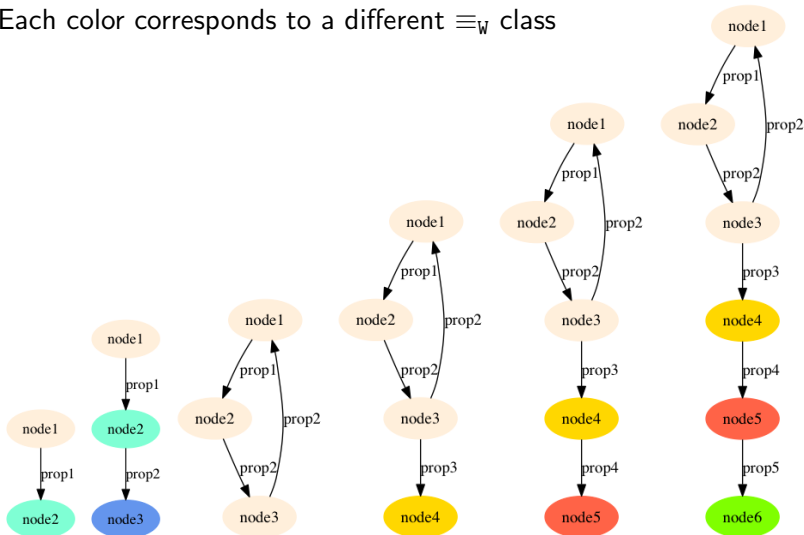
Example: weak incremental summarization (1) [GGM19]

Each color corresponds to a different \equiv_W class



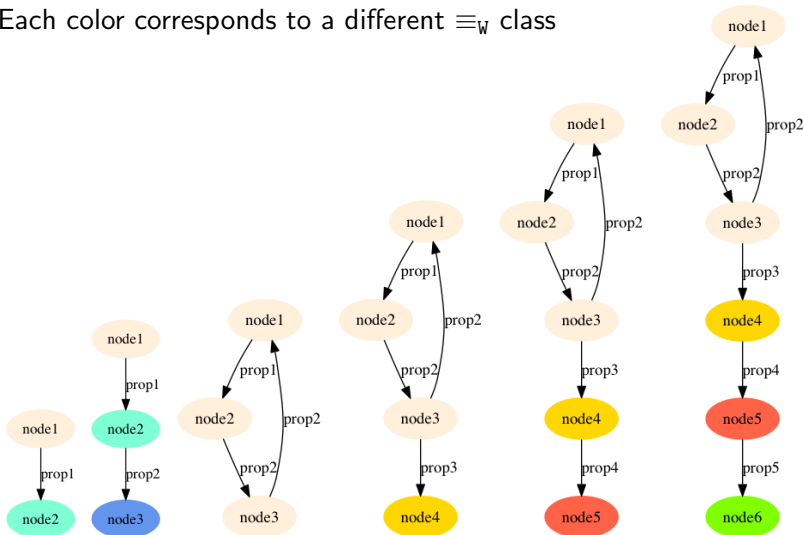
Example: weak incremental summarization (1) [GGM19]

Each color corresponds to a different \equiv_W class

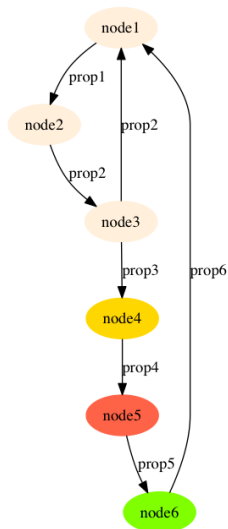


Example: weak incremental summarization (1) [GGM19]

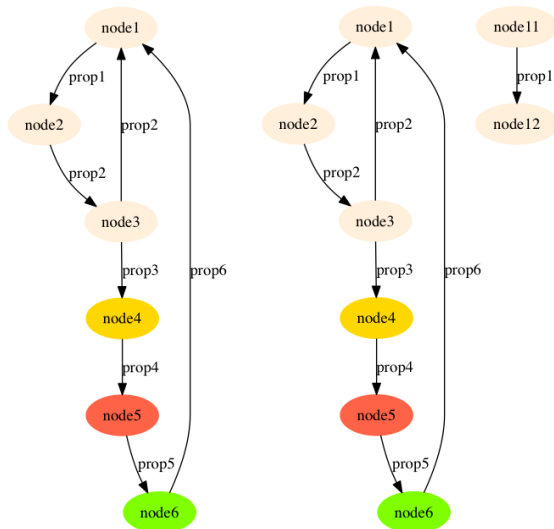
Each color corresponds to a different \equiv_W class



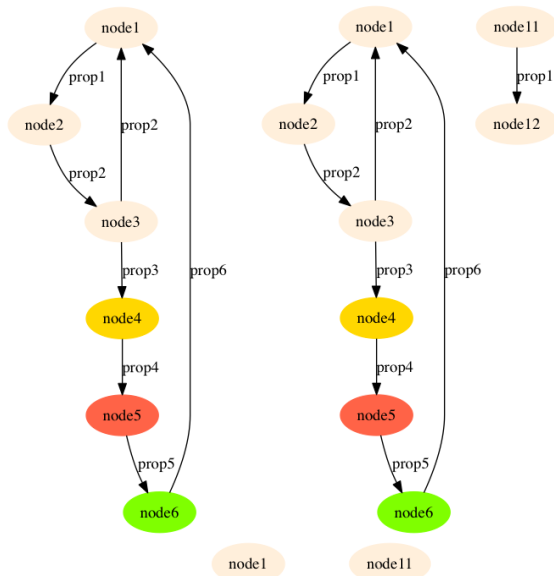
Example: weak incremental summarization (2) [GGM19]



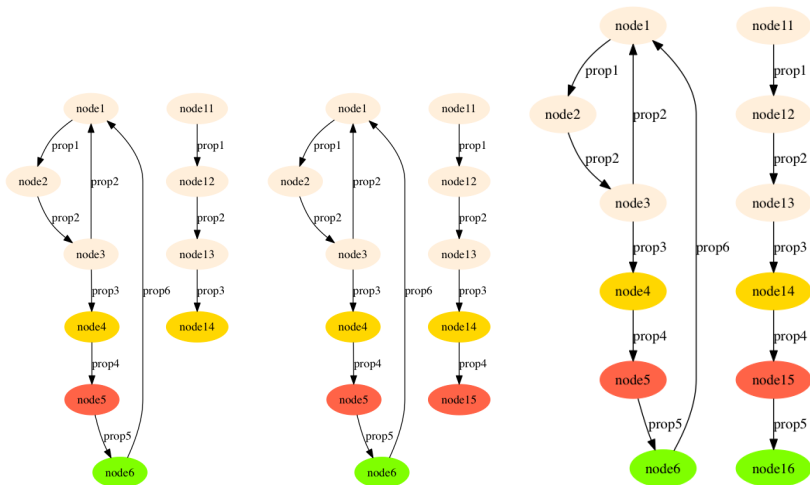
Example: weak incremental summarization (2) [GGM19]



Example: weak incremental summarization (2) [GGM19]

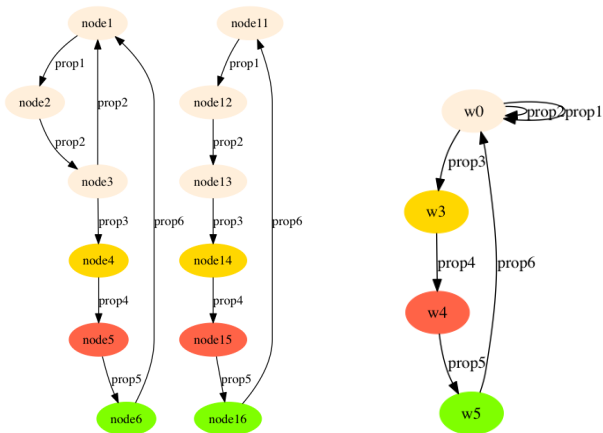


Example: weak incremental summarization (3) [GGM19]



Example: weak incremental summarization (end) [GGM19]


Full graph and its summary:

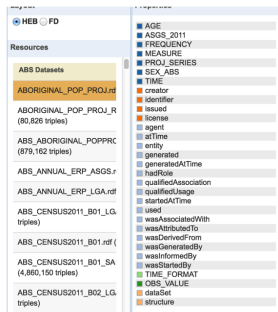


Visualizing summaries (1)

Summary-enabled LOD cloud exploration [PGA⁺18], online at <http://lodatlas.lri.fr/>

Use summary to derive visualisation instead of the original graph (smaller, faster)

abs-linked-data : Australian Bureau of Statistics (ABS) Linked Data 



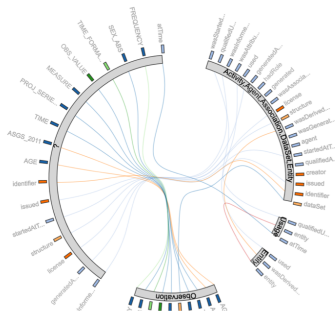
Resources

ABS Datasets

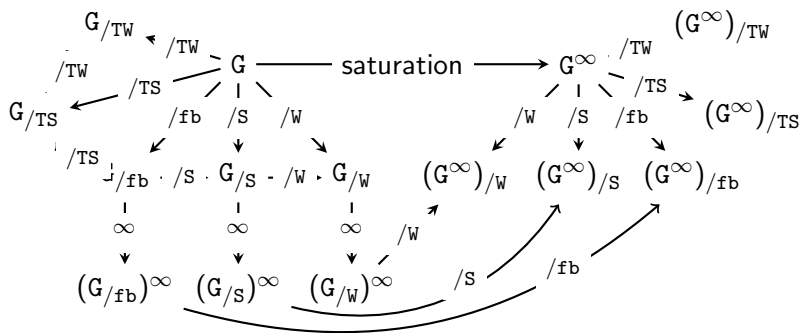
- ABORIGINAL_POP_PROJ.rdf
- ABORIGINAL_POP_PROJ_R (80,826 triples)
- ABS_ABORIGINAL_POPPRC (879,162 triples)
- ABS_ANNUAL_ERP_ASGS.r
- ABS_ANNUAL_ERP_LGA.rdf
- ABS_CENSUS2011_B01_LG (triples)
- ABS_CENSUS2011_B01.rdf (
- ABS_CENSUS2011_B01_SA (4,860,150 triples)
- ABS_CENSUS2011_B02_LG (triples)

Properties

- AGE
- ASGS_2011
- FREQUENCY
- MEASURE
- PROJ_SERIES
- SEX_ABS
- TIME
- creator
- identifier
- issued
- license
- agent
- atTime
- entity
- generated
- generatedAtTime
- hadRole
- qualifiedAssociation
- qualifiedUsage
- startedAtTime
- used
- wasAssociatedWith
- wasAttributedTo
- wasDerivedFrom
- wasGeneratedBy
- wasInformedBy
- wasStartedBy
- TIME_FORMAT
- OBS_VALUE
- dataSet
- structure



Relations between quotient RDF summaries [ČGGM17]



Clique-based summaries often orders of magnitude smaller than bisimulation-based ones.

Welcome back to the RDF summarization tutorial

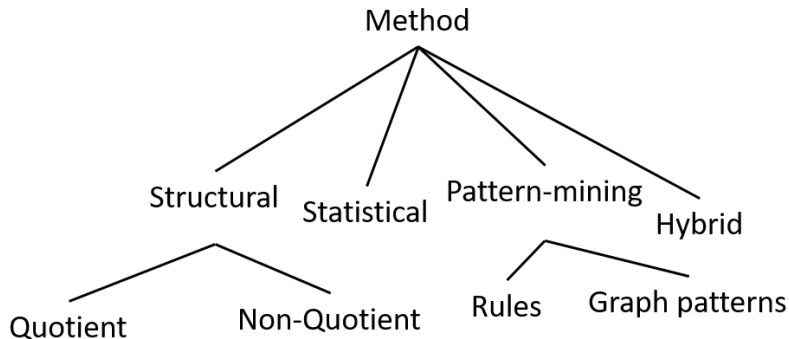
We covered:

- RDF basics
- Generic RDF summarization method
- Structural RDF summarization through quotient methods

We now present:

- Structural RDF summarization through non-quotient methods
- Pattern-based summarization
- Statistical summarization
- Hybrid summarization
- Conclusion

A taxonomy of summarization methods



Text summarization and information retrieval

As in text summaries the objective here is to select the most **significant** subject/part of the RDF graph using

- 1 **cognitive**
- 2 **lexical**
- 3 **topological** measures

Usually they heavily involve end-users.

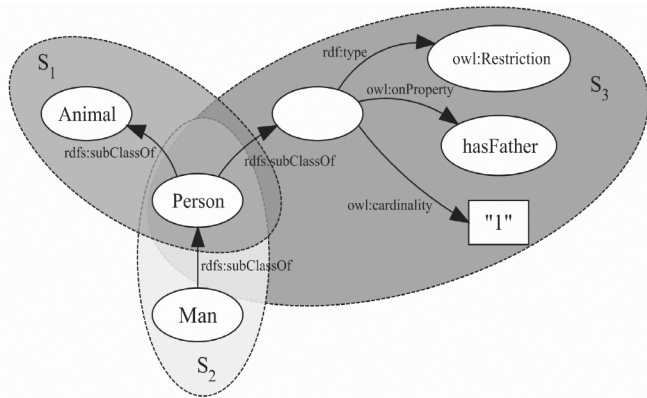
Text summarization and information retrieval

For example an approach focusing only on instance graph [SPS13]:

- 1 Users select an entity
- 2 Identify triples favoring **closeness** to the target entity
- 3 Extend this selection with criteria based on:
 - **diversity** - include edges with different labels
 - **popularity** - favour frequently occurring edge labels

Bag of sentences

Bag of sentences approach [ZCQ07], only for the schema graph.



Focusing on Centrality measures

Approaches that focus on centrality measures from graph theory to identify the most important nodes.

RDFDigest+ [TKDP17], [TKDP15]

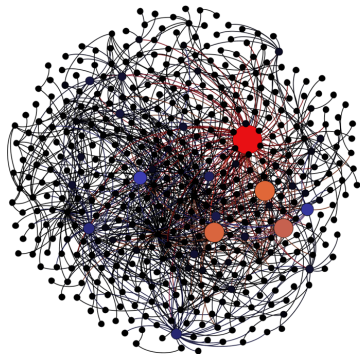
- 1 Identify the **most important nodes**
- 2 **Link** those nodes
- 3 Enable **exploration operations**

RDFDigest+

- 1 Identify the **most important nodes**
 - **But How?**
- 2 Link those nodes
- 3 Enable exploration operations

RDFDigest+

- 1 Identify the **most important nodes**
 - Using **centrality measures** like Relevance, Degree, Betweenness, Bridging Centrality, Harmonic Centrality, Radiality, HITS, Pagerank etc.
- 2 Link those nodes
- 3 Enable exploration operations



RDFDigest+

- 1 Identify the **most important nodes**
 - Using centrality measures like Relevance, Degree, Betweenness, Bridging Centrality, Harmonic Centrality, Radiality, HITS, Pagerank etc.
 - **But what about instances?**
- 2 Link those nodes
- 3 Enable exploration operations

RDFDigest+

- 1 Identify the **most important nodes**
 - Using **adapted** centrality measures like Relevance, Degree, Betweenness, Bridging Centrality, Harmonic Centrality, Radiality, HITS, Pagerank etc.
- 2 Link those nodes
- 3 Enable exploration operations

RDFDigest+

- 1 Identify the most important nodes
 - Using adapted centrality measures like Relevance, Degree, Betweenness, Bridging Centrality, Harmonic Centrality, Radiality, HITS, Pagerank etc.
- 2 Link those nodes
 - But How? Which would be your objective here?
- 3 Enable exploration operations

RDFDigest+

- 1 Identify the most important nodes
 - Using adapted centrality measures like Relevance, Degree, Betweenness, Bridging Centrality, Harmonic Centrality, Radiality, HITS, Pagerank etc.
- 2 Link those nodes
 - Find the minimum weight spanning tree that connects them - Minimum Spanning Tree problem
 - Reduce additional nodes introduced - Steiner Tree problem
- 3 Enable exploration operations

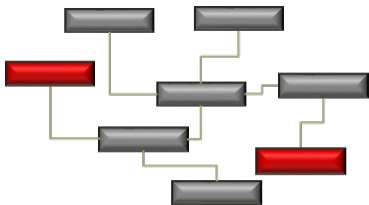
RDFDigest+

- 1 Identify the most important nodes
 - Using adapted centrality measures like Relevance, Degree, Betweenness, Bridging Centrality, Harmonic Centrality, Radiality, HITS, Pagerank etc.
- 2 Link those nodes
 - Find the minimum weight spanning tree that connects them - Minimum Spanning Tree problem
 - Reduce additional nodes introduced - Steiner Tree problem
- 3 What about evaluation?
- 4 Enable exploration operations

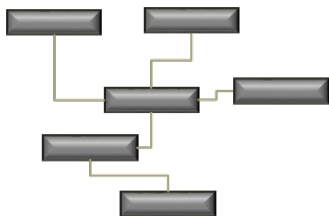
RDFDigest+

- 1 Identify the most important nodes
- 2 Link those nodes
- 3 Enable exploration operations
 - Zoom
 - Extend

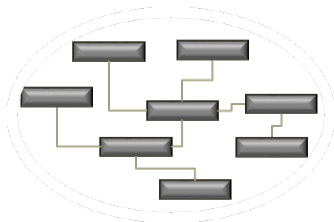
Zoom-Out



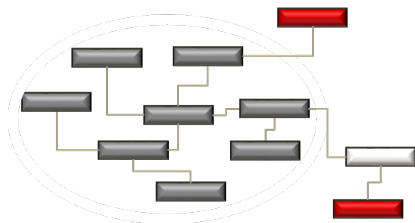
Zoom-Out



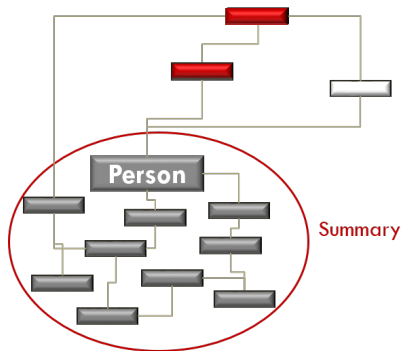
Zoom-In



Zoom-In



Extend

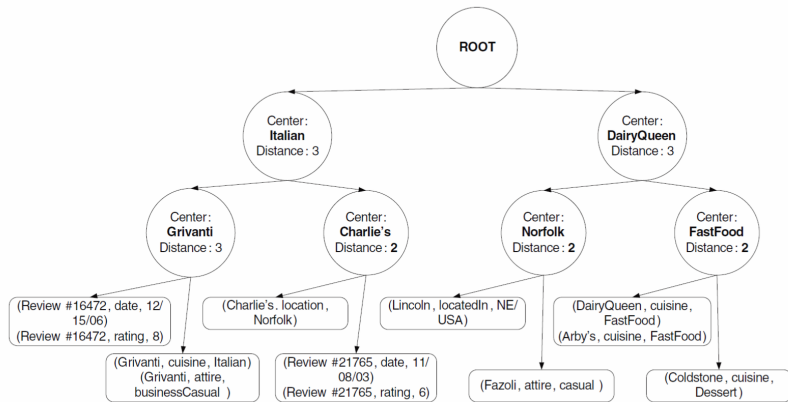


Indexing

A GRIN index [UPS07] is a **hierarchical clustering** of the RDF instance graph, modeled as a balanced binary tree.

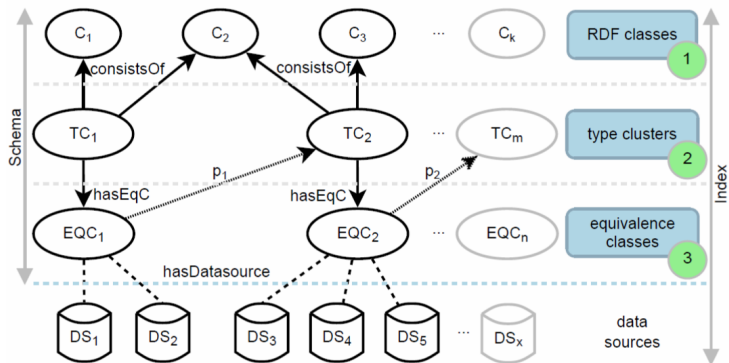
- 1 Each **leaf node** represents a set of resources.
- 2 **Interior nodes** implicitly represent the set of all graph nodes that are **within R units of distance** from a center triple.
- 3 Each inner node reflects the resources of the triples of the nodes it is an ancestor of.
- 4 At query time, inequality constraints derived from the query are evaluated against the index nodes, to identify the smallest sub-graph that contains answers to the input query.

Indexing



Schema extraction

SchemEX [KGSS12] is an indexing and schema extraction tool for the LOD cloud.



Closing remarks

- Many structural summaries based on quotients
 - Special treatment of schema nodes is needed
 - Interplay between summarization and saturation
 - Bisimilarity-based equivalence: accurate summarization, large summaries
 - Clique-based equivalence: tolerant of heterogeneity, compact summaries
- Other approaches try to identify the most important parts and focus on those
 - Based on text content
 - Centrality measures
 - User input

Part VI

Pattern-based RDF Summarization

Pattern-based RDF Summarization

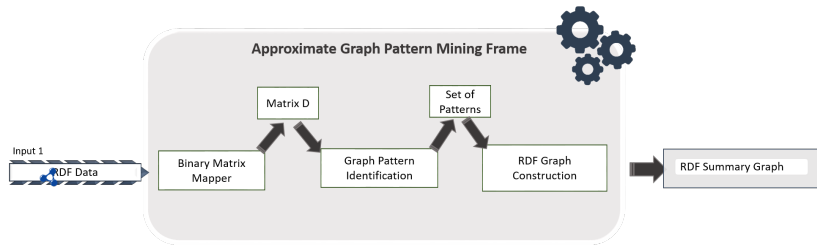
Based on data mining techniques: extract *frequent patterns* from the RDF graph and use these patterns to represent the original RDF graph.

- A frequent pattern or *knowledge pattern* characterizes a set of instances in an RDF graph that share a common set of types and a common set of properties.
- Modeled as a star BGP of the form $\{x \tau c_1, \dots, x \tau c_n, x Pr_1 ?b_1, \dots, x Pr_m ?b_m\}$ denoting some resource x having types c_1, \dots, c_n and properties Pr_1, \dots, Pr_m .
- Given an RDF graph G , a pattern KP identifies all the G resources that match x in the embeddings of KP into G ; the number of such embeddings is called the support of KP in G .

Pattern-based RDF Summarization

- Algorithms in this category work by trying to optimize (maximize or minimize) a cost function
- Cost functions considered:
 - Noise / erroneous identified data
 - Similarity
- Greedy algorithms with considerable time complexity, use of heuristics

Using approximate graph pattern mining [ZLVK16, ZLVK15]



- Transforms RDF graph into a binary matrix
- Uses a calibrated version of the PaNDa+ [LOP14] algorithm, to find the best approximate RDF graph patterns; supports different cost functions
- Stops when no improvement of the cost function is reported
- Reconstructs summary from patterns

Using approximate graph pattern mining [ZLVK16, ZLVK15]

- Greedy algorithm: tests all possible combinations of improvement of the identified knowledge patterns, until none is found
- Time complexity: $O(M * N + K * M^2 * N)$
 - K is the maximum number of patterns to be extracted,
 - M , the number of distinct properties and N , the number of distinct subjects/resources in the original KB

Using pattern similarity [SWD16]

- Maximize an *informativeness* measure/cost function (input) to produce k summaries (input)
- Computes the d -similarity among all nodes and will choose those that are more similar for the d -summaries
 - d -similarity: is the similarity of the extended (of distance d) neighborhood
- Compute the maximal d -summaries by mining patterns
 - d -summary (pattern): an itemset of d -similar nodes
 - Pick maximal patterns (those with greater *support*)
- Greedily tests summary pairs of d -summaries to find the one that maximizes *informativeness* until it reaches k

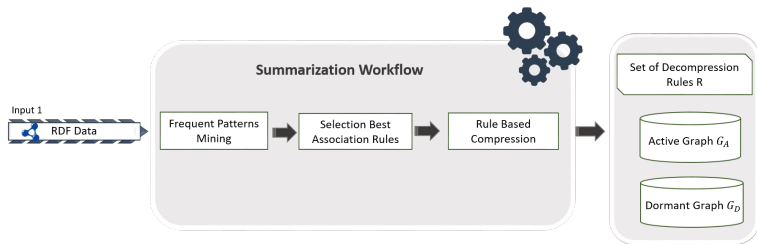
Using pattern similarity [SWD16]

- Greater flexibility: looking into the extended neighborhood of a node
- Greedy optimization of the cost function
- Time complexity: $O(S * (b + N) * (b + M) + \frac{K}{2} * S^2)$,
 - N is the total number of nodes (subject and objects) and M is the total number of edges (triples) of the original RDF graph
 - S is the number of possible d -summaries whose size is bounded by b

Mining Rules

- Rule mining techniques to extract rules to be used as the summary of the RDF graph
- Important limitation: the summary is by definition not an RDF graph and thus cannot be queried by standard RDF tools
- Used a lot in RDF graph compression

Mining Rules: Graph compression techniques [JHD13]



- Transforms RDF Graph G to $R(G_A) \cup G_D$, where
 - G_A is the active graph containing the triples that adhere to certain logical rules R ,
 - G_D is the dormant graph, which contains the set of triples of the original graph that cannot be described by a rule
- Use of the Apriori or FP-Growth frequent pattern mining algorithms to identify sets of association rules

Mining Rules: Graph compression techniques [JHD13]

- Sample rule: $\forall x, (x, p, k) \rightarrow \bigwedge_{i=1}^n (x, p, v_i)$, stating that the subjects that carry the value k for property p , carry also the values u_i for the same property
- Left triple is kept in the summary, right triples removed
- Works well when we have many nodes with similar neighbors (e.g. same literal values)
- Extension: add two variables instead of one; more patterns represented by a rule, semantic similarity/coherence decreased
- Time complexity: $O(M * R + N_p * O_v^2 * N_s)$, where:
 - M is the total number of triples
 - R is the number of the generated logical rules
 - N_p and N_s are respectively the number of distinct properties and subjects/resources in the graph
 - O_v is the average number of different objects/values that are assigned to a property p

Closing Remarks

- Identify the most "frequent" graph patterns
- Try to join those patterns together in an effort to optimize a given cost function
- Join the grouped/optimized patterns together to form the RDF summary

Part VII

Statistical RDF Summarization

Statistical RDF Summarization

- Source selection problem: where to direct your queries to get back the important results?
- Identify if in a source there is significant information concerning a graph pattern of interest
- Different than the pattern mining category: does not necessarily care for the structural completeness; reduced computational cost

SPARQL ASK queries and beyond [BB10, SHH⁺11, HS12]

- Use SPARQL ASK queries to identify if a triple pattern exists or not
- Extend these queries to return a concise summary of the results (e.g. as Bloom filters)
- Function that estimates the benefit for retrieving results for a triple pattern; ignore sources with low benefit
- *Sketches* = summaries with statistical information on the instances
- No input required from the users

Concept and Relation Ranking [WLFW08]

- Find **important** concepts and relations from the schema
- Importance of concepts = number of relations starting from it + number of relations to important concepts + weights of these relations
- Weight of the relation: the more important the concept at the source of a relation, the higher the weight
- Iterative approach: important concepts and important relations reinforce each other
- Does not use any instance information!

OWL Summarizer [PQKS10]

Used for peer clustering: an incoming peer with its own local ontology searches for similar peers by comparing schema summaries

- **Summary:** top- k concepts grouped together by adjacency
- Concept weight: centrality + frequency
 - *Centrality*: degree centrality; different weights assigned to user-defined and RDFS properties
 - *Frequency*: the ratio between the number of concept appearances and the number of distinct local ontologies
- Non-adjacent groups: examine first k -paths; pick the best
- No data or implicit triples are considered!

LODSight [DSM15, MDTs16]

LODSight: RDF Summary Visualization Tool

- Uses results of iterative SPARQL queries; no user input is required
- Combines types and predicates
- **Summary:** a collection of the types and predicates that appear the most
- Implicit information is considered if return by the endpoint

LODSight [DSM15, MDTs16]

LODSight Extended

- Instantiate the summary patterns
 - Random: random examples for all summary paths
 - Distinct: examples that are further away for each path (distance)
 - Representative: diversity and representativeness criteria
- **Summary**: a collection of the types and predicates that appear the most
- Implicit information is considered if return by the endpoint

Using an organizational ontology [PAA⁺11]

- Use of an ontology to organize the statistical information around the dataset
 - triples
 - paths: types and properties
 - statistics for each path
- Core (=frequencies + position in paths) types and properties extracted
- no implicit triples accounted for
- Used in query answering

Closing remarks

- Calculating and using statistics to identify the most "prominent" types and properties to include in the summary
- Some methods do not use data and rely only on the schema information (what if schema information does *not* exist or is *incomplete*?)
- Applications in source selection and visualization

Part VIII

Other RDF Summarization Methods

Other Summarization Methods

- 1 Approaches that combine **structural**, **statistical** and **pattern-mining methods** in order to get better results.
- 2 Methods going beyond RDF graph summaries, for example summarizing **DL ontologies**.

Hybrid structural summarization [AL13]

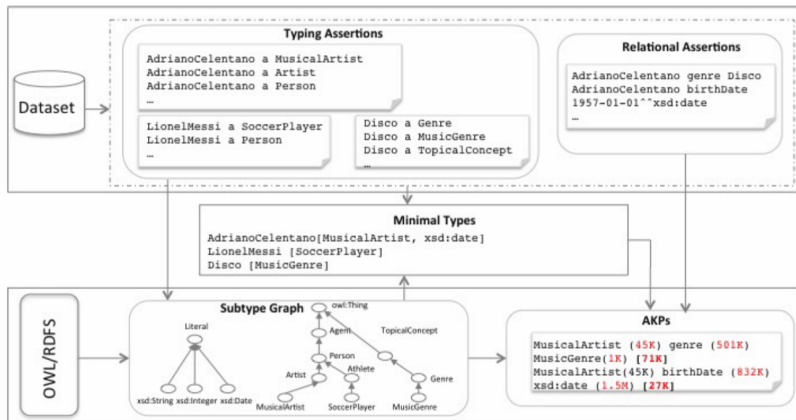
Identify similar structured inside RDF-Graphs: Reduce graph size while retaining the structure as much as possible.

- 1 First step: **bounded forward bisimulation**, building a graph representing all the N nodes and M edges of the input graph
- 2 Second step: **hierarchical clustering**, fusing root nodes of similar depth-bounded tree subgraphs.

ABSTAT [SPP⁺16]

Combines **statistical** and **pattern mining** methods aiming at reflecting how class instances are related through properties.

- A summary is a set of **abstract knowledge patterns** (AKPs) (c_1, p, c_2) representing the (s, p, o) graph triples with c_1 (resp. c_2) one of the most specific types of s (resp. o).
 - 1 Compute for every value present in the graph all its types.
 - 2 Prune out the redundant ones.
 - 3 For each property assertion (s, p, o) , build an AKP (c_1, p, c_2) if c_1 (resp. c_2) is a most specific type for value s (resp. o).

ABSTAT [SPP⁺16]

Estimating the cardinality of conjunctive queries [SMK17]

- 1 Group together nodes having exactly the same set of types, same outgoing and same incoming properties
- 2 A summary edge is labeled with the number of edges of G that have been collapsed due to merging
- 3 Reduce a potentially very large summary to a target size specified by the user, by **merging** nodes having similar incoming and outgoing properties.
 - The similarity is determined by a Jaccard index approximation.

Mining equivalent structure patterns

Common to have different graph structures, sharing the same meaning. Can we automatically exploit those [ZZP⁺16]?

- 1 Rewrite the input query to one considering semantic equivalences
- 2 Find the subgraphs minimizing the semantic graph edit distance
- 3 A *semantic summary graph* is built off-line enabling a two-level pruning at query time.
 - A multi-layer graph where the first layer is consisted of the linked types of the instances, and the other layers above replace in each layer classes with their superclass.

Summarizing Horn- \mathcal{ALCHOI} description logic KBs

[GKL⁺14]

Compressing the ABox of a Horn- \mathcal{ALCHOI} description logic KBs, using the notion of **ABox abstraction**.

- ① Given an ABox \mathcal{A} , for each \mathcal{A} value v , a type pattern of the form $tp(v) = (tp_{\downarrow}, tp_{\rightarrow}, tp_{\leftarrow})$ is computed
 - ① tp_{\downarrow} denotes the explicit types v has in \mathcal{A}
 - ② tp_{\rightarrow} the outgoing properties v has in \mathcal{A}
 - ③ tp_{\leftarrow} the incoming properties v has in \mathcal{A}
- ② These type patterns are then used to build the abstraction \mathcal{B} of the ABox, which is an ABox itself
 - each such type pattern is used to represent all the ABox values that match it

Summarizing *SHIN* KBs

ABox summaries in *SHIN* KBs have also been considered for

- 1 Consistency checking [FKM⁺06b], [FKM06a]
- 2 Query answering [DFK⁺07], [DFK⁺09]

In these works, the notion of a summary ABox is very general: an ABox \mathcal{A}' is a summary of another ABox \mathcal{A} w.r.t. some function f that maps \mathcal{A} values to \mathcal{A}' ones whenever f defines a **homomorphism** from \mathcal{A} to \mathcal{A}' .

Part IX

Conclusions & Future Work

Conclusions

- Summarizing RDF graphs has many useful applications: from data understanding to query answering and from RDF data indexing to RDF graph visualization
- Diverse set of algorithms coming from different domains based on different concepts
- Effort to homogenize the way we look at those algorithms and proposed a taxonomy to navigate the space of alternatives

Conclusions - Cont.

- A **taxonomy** of the works in this area has been proposed, for practitioners and researchers to easily find the algorithm(s) more appropriate for their use case
- Identified 4 main categories:
 - structural
 - quotient (indexing and query answering through graph reduction)
 - not-quotient (visualization, schema discovery and data understanding)
 - pattern mining (RDF graph compression, RDF schema discovery)
 - statistical (RDF schema understanding)
 - hybrid
- Used diverse criteria like: input, output, availability, purpose

More: *Cebiric, S., Goasdoué, F., Kondylakis, H., Kotzinos, D., Manolescu, I., Troullinou, G., & Zneika, M. (2018). Summarizing Semantic Graphs: A Survey. The VLDB Journal. [CGK⁺18]*

Future Work

- How to compare summaries / assess the **quality** of a summary
 - few works in the area, e.g. [ZVK18]
- Construction of golden standards
 - difficult process, even experts disagree
 - highly dependent on the application
- Work to improve the speed of the summarization
 - parallelization of the algorithms
 - Linked Data
- Account for the dynamic nature of the data
 - schema evolution [KP11], [KP12]
 - data updates
- Propagating graph updates to the summary

References

- [AL13] Anas Alzogbi and Georg Lausen. Similar structures inside rdf-graphs. In Proceedings of the WWW2013 Workshop on Linked Data on the Web, Rio de Janeiro, Brazil, 14 May, 2013, 2013.
- [BB10] Cosmin Basca and Abraham Bernstein. Avalanche: Putting the spirit of the web back into semantic web querying. In Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, Shanghai, China, November 9, 2010, 2010.
- [ČGGM17] Šejla Čebirić, François Goasdoué, Pawel Guzewicz, and Ioana Manolescu. Compact Summaries of Rich Heterogeneous Graphs. Research Report RR-8920, INRIA Saclay ; Université Rennes 1, June 2017.
- [CGK⁺18] Sejla Cebiric, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. Summarizing semantic graphs: A survey. The VLDB Journal, 2018.
- [ČGM15] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Query-oriented summarization of RDF graphs. PVLDB, 8(12):2012–2015, 2015.
- [ČGM17] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. A framework for efficient representative summarization of RDF graphs. In International Semantic Web Conference (ISWC), 2017.

References (cont.)

- [DFK⁺07] Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Aaron Kershenbaum, Edith Schonberg, Kavitha Srinivas, and Li Ma. Scalable semantic retrieval through summarization and refinement. In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada, pages 299-304, 2007.
- [DFK⁺09] Julian Dolby, Achille Fokoue, Aditya Kalyanpur, Edith Schonberg, and Kavitha Srinivas. Scalable highly expressive reasoner (SHER). J. Web Semant., 7(4):357-361, 2009.
- [DSM15] Marek Dudás, Vojtech Svátek, and Jindrich Mynarz. Dataset summary visualization with Iodsight. In The Semantic Web: ESWC 2015 Satellite Events - ESWC 2015 Satellite Events Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers, pages 36-40, 2015.
- [FKM06a] Achille Fokoue, Aaron Kershenbaum, and Li Ma. SHIN abox reduction. In Proceedings of the 2006 International Workshop on Description Logics (DL2006), Windermere, Lake District, UK, May 30 - June 1, 2006, 2006.

References (cont.)

- [FKM⁺06b] Achille Fokoue, Aaron Kershenbaum, Li Ma, Edith Schonberg, and Kavitha Srinivas. The summary abox: Cutting ontologies down to size. In The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings, pages 343–356, 2006.
- [GGM19] François Goasdoué, Pawel Guzewicz, and Ioana Manolescu. Incremental structural summarization of RDF graphs. In EDBT 2019 - 22nd International Conference on Extending Database Technology, Lisbon, Portugal, March 2019.
- [GKL⁺14] Birte Glimm, Yevgeny Kazakov, Thorsten Liebig, Trung-Kien Tran, and Vincent Vialard. Abstraction refinement for ontology materialization. In The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II, pages 180–195, 2014.
- [GW97] Roy Goldman and Jennifer Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. In VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece, pages 436–445, 1997.

References (cont.)

- [HHK95] Monika Rauch Henzinger, Thomas A. Henzinger, and Peter W. Kopke. Computing simulations on finite and infinite graphs. In FOCS, 1995.
- [HS12] Katja Hose and Ralf Schenkel. Towards benefit-based RDF source selection for SPARQL queries. In Proceedings of the 4th International Workshop on Semantic Web Information Management, SWIM 2012, Scottsdale, AZ, USA, May 20, 2012, page 2, 2012.
- [JHD13] Amit Krishna Joshi, Pascal Hitzler, and Guozhu Dong. Logical linked data compression. In The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings, pages 170–184, 2013.
- [KBNK02] Raghav Kaushik, Philip Bohannon, Jeffrey F Naughton, and Henry F Korth. Covering indexes for branching path queries. In SIGMOD, 2002.
- [KGSS12] Mathias Konrath, Thomas Gottron, Steffen Staab, and Ansgar Scherp. Schemex - efficient construction of a data catalogue by stream-based indexing of linked data. J. Web Sem., 16:52–58, 2012.
- [KP11] Haridimos Kondylakis and Dimitris Plexousakis. Ontology evolution in data integration: Query rewriting to the rescue. In Conceptual Modeling - ER 2011, 30th International Conference, ER2011, Brussels, Belgium, October 31 - November 3, 2011. Proceedings, pages 393–401, 2011.

References (cont.)

- [KP12] Haridimos Kondylakis and Dimitris Plexousakis. Ontology evolution: Assisting query migration. In *Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings*, pages 331–344, 2012.
- [LFH⁺13] Yongming Luo, George H. L. Fletcher, Jan Hidders, Yuqing Wu, and Paul De Bra. External memory k-bisimulation reduction of big graphs. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 919–928, 2013.
- [LOP14] Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. A unifying framework for mining approximate top- k binary patterns. *IEEE Trans. Knowl. Data Eng.*, 26(12):2900–2913, 2014.
- [LSDK18] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM Comput. Surv.*, 51(3):62:1–62:34, June 2018.

References (cont.)

- [LTH⁺14] Xingjie Liu, Yuanyuan Tian, Qi He, Wang-Chien Lee, and John McPherson. Distributed graph summarization. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, pages 799–808, 2014.
- [MDTS16] Jindrich Mynarz, Marek Dudás, Paolo Tomeo, and Vojtech Svátek. Generating examples of paths summarizing RDF datasets. In Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016., 2016.
- [MS99] Tova Milo and Dan Suciu. Index structures for path expressions. In Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings., pages 277–295, 1999.

References (cont.)

- [PAA⁺11] Valentina Presutti, Lora Aroyo, Alessandro Adamou, Balthasar A. C. Schopman, Aldo Gangemi, and Guus Schreiber. Extracting core knowledge from linked data. In Proceedings of the Second International Workshop on Consuming Linked Data (COLD2011), Bonn, Germany, October 23, 2011, 2011.
- [PGA⁺18] Emmanuel Pietriga, Hande Gözükan, Caroline Appert, Marie Destandau, Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Browsing linked data catalogs with LODAtlas. In Int'l. Semantic Web Conference (ISWC), Resources track, 2018.
- [PQKS10] Carlos Eduardo S. Pires, Paulo Orlando Queiroz-Sousa, Zoubida Kedad, and Ana Carolina Salgado. Summarizing ontology-based schemas in PDMS. In Workshops Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA, pages 239–244, 2010.
- [PT87] Robert Paige and Robert Endre Tarjan. Three partition refinement algorithms. SIAM J. Comput., 16(6):973–989, 1987.

References (cont.)

- [SHH⁺11] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. Fedx: Optimization techniques for federated query processing on linked data. In The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I, pages 601–616, 2011.
- [SMK17] Giorgio Stefanoni, Boris Motik, and Egor V. Kostylev. Estimating the Cardinality of Conjunctive Queries over RDF Data Using Graph Summarisation. Research report, University of Oxford, 2017.
- [SPP⁺16] Blerina Spahiu, Riccardo Porrini, Matteo Palmonari, Anisa Rula, and Andrea Maurino. ABSTAT: ontology-driven linked data summaries with pattern minimalization. In SumPre, 2016.
- [SPS13] Marcin Sydow, Mariusz Pikula, and Ralf Schenkel. The notion of diversity in graphical entity summarisation on semantic knowledge graphs. J. Intell. Inf. Syst., 41(2):109–149, 2013.
- [SWD16] Qi Song, Yinghui Wu, and Xin Luna Dong. Mining summaries for knowledge graph search. In IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain, pages 1215–1220, 2016.

References (cont.)

- [THP08] Yuanyuan Tian, Richard A. Hankins, and Jignesh M. Patel. Efficient aggregation for graph summarization. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008, pages 567–580, 2008.
- [TKDP15] Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. RDF digest: Efficient summarization of RDF/S kbs. In The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings, pages 119–134, 2015.
- [TKDP17] Georgia Troullinou, Haridimos Kondylakis, Evangelia Daskalaki, and Dimitris Plexousakis. Ontology understanding without tears: The summarization approach. Semantic Web, 8(6):797–815, 2017.
- [TP10] Yuanyuan Tian and Jignesh M Patel. Interactive graph summarization. In Link Mining: Models, Algorithms, and Applications, pages 389–409. Springer, 2010.
- [UPS07] Octavian Udrea, Andrea Pugliese, and V. S. Subrahmanian. GRIN: A graph based RDF index. In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada, pages 1465–1470, 2007.

References (cont.)

- [WLFW08] Gang Wu, Juanzi Li, Ling Feng, and Kehong Wang. Identifying potentially important concepts and relations in an ontology. In ISWC, pages 33–49, 2008.
- [ZCQ07] Xiang Zhang, Gong Cheng, and Yuzhong Qu. Ontology summarization based on rdf sentence graph. In Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, pages 707–716, 2007.
- [ZLVK15] Mussab Zneika, Claudio Lucchese, Dan Vodislav, and Dimitris Kotzinos. RDF graph summarization based on approximate patterns. In Information Search, Integration, and Personalization - 10th International Workshop, ISIP 2015, Grand Forks, ND, USA, October 1-2, 2015, Revised Selected Papers, pages 69–87, 2015.
- [ZLVK16] Mussab Zneika, Claudio Lucchese, Dan Vodislav, and Dimitris Kotzinos. Summarizing linked data RDF graphs using approximate graph pattern mining. In Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016., pages 684–685, 2016.

References (cont.)

- [ZVK18] Mussab Zneika, Dan Vodislav, and Dimitris Kotzinos. Quality Metrics For RDF Graph Summarization. Semantic Web Journal (SWJ), accepted, to appear, 2018.
- [ZZP⁺16] Weiguo Zheng, Lei Zou, Wei Peng, Xifeng Yan, Shaoxu Song, and Dongyan Zhao. Semantic SPARQL similarity search over RDF knowledge graphs. PVLDB, 9(11):840–851, 2016.