

Tutorial on Semantic Schema Discovery: principles, methods and future research directions Part 4

Kenza Kellou-Menouer, Nikolaos Kardoulakis, Georgia
Troullinou, Zoubida Kedad, Dimitris Plexousakis,
Haridimos Kondylakis

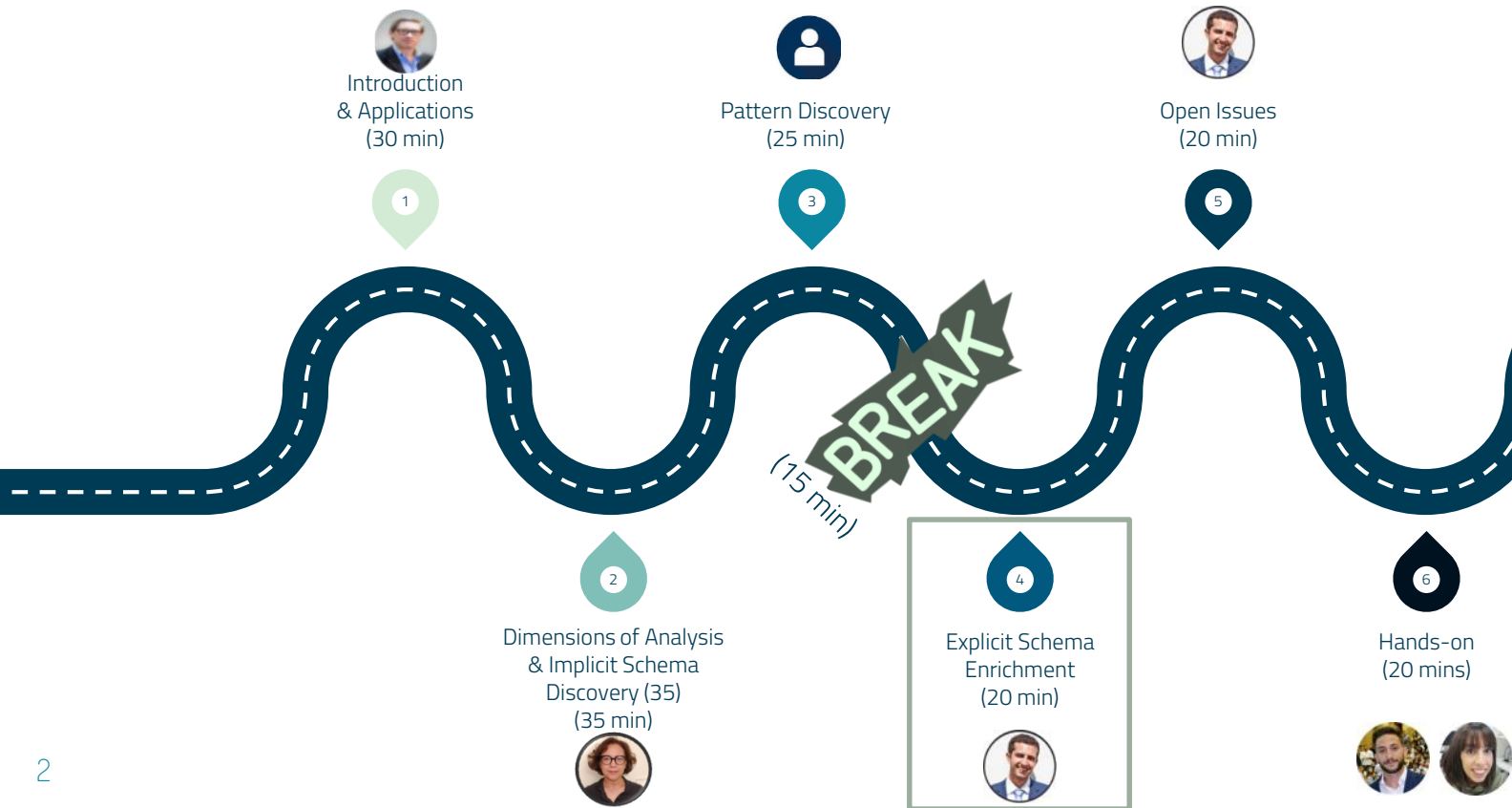
BREAK



Équipes Traitement
de l'Information
et Systèmes

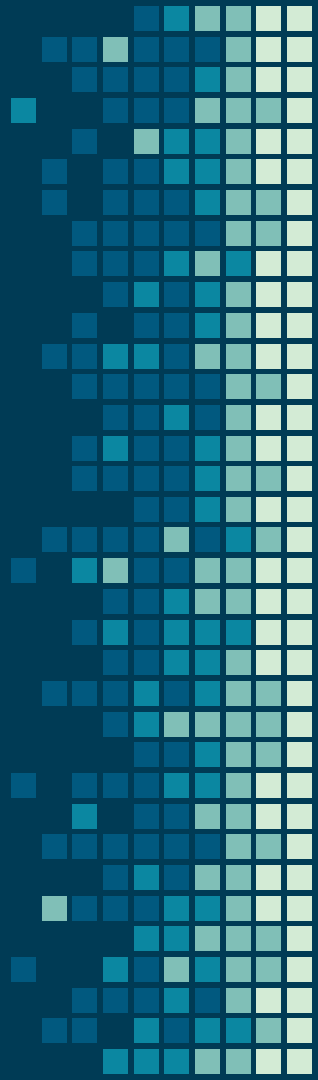


ROADMAP



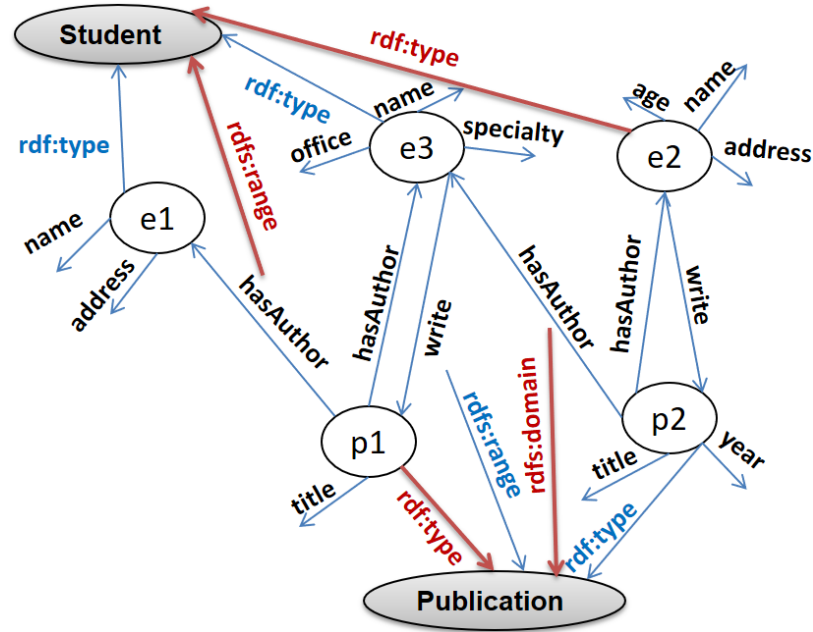


Explicit Schema Enrichment



Explicit Schema Enrichment

- Use existing statements on the schema and **complement** or **enrich** them



Main Techniques


- Using Machine Learning
- Statistical Analysis



	Inputs		Techniques	Outputs
	Data description	Setting		
Volker et al. [95]	RDF data: Incoming and outgoing properties, <i>rdf : type</i> statements	Support and confidence thresholds for association rules	Unsupervised Learning using association rules discovery algorithm	Semantic links, hierarchical links, OWL axioms
Paulheim [70]	RDF data: Incoming and outgoing properties, part of <i>rdf : type</i> statements	Support and confidence thresholds for association rules	Unsupervised Learning using association rules discovery algorithm	Types (additional <i>rdf : type</i> statements)
Nuzzolese et al. [68]	RDF/OWL data: Wikilinks, <i>owl : sameAs</i> statements	Number of neighbors K in K -NN	Supervised learning using K -NN	Types (additional <i>rdf : type</i> statements)
Zong et al. [110]	RDF data: Incoming and outgoing properties, <i>rdf : type</i> statements	No required setting	Unsupervised Learning using Hierarchical clustering - VSM	Hierarchical links (<i>rdfs : subclassOf</i>)
Bühmann et al. <i>DL-Learner</i> [17,18]	RDF(S)/OWL data: Incoming and outgoing properties, part of schema (<i>rdf : type</i> , <i>rdfs : domain</i> , <i>rdfs : range</i> , <i>rdfs : subclassOf</i>), OWL axioms for reasoning if provided	Threshold for candidate axioms	Supervised Learning using different algorithms	Additional OWL axioms with scores

GoldMiner (Volkel et al. 2011)

- Main assumptions:
 - the semantics of any RDF resource, is revealed by **patterns**
 - association rules** are inferred on RDF triples to acquire knowledge at the schema level



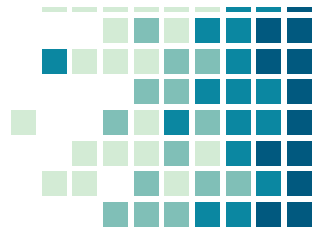
Instances
Classes
Incoming Properties
Outgoing Properties

	C ₁	C ₂	C ₃	P _{d1}	P _{d2}	P _{d3}	P _{r1}	P _{r2}	P _{r3}
e ₁	0	1	0	1	0	1	1	1	0
e ₂	1	0	1	0	1	1	1	0	1
e ₃	1	0	1	1	0	1	0	0	1
e ₄	0	1	0	0	1	0	1	1	0
e ₅	1	0	0	0	1	1	0	1	1

C₃ → C₁
 C₁ → P_{d3}
 C₂ → P_{r1}



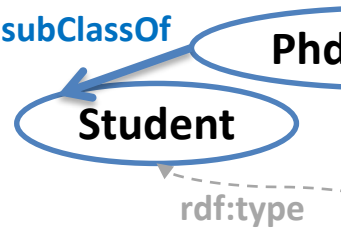
C₃ rdfs : subClassOf C₁
 P₃ rdfs : domain C₁
 P₁ rdfs : range C₂



Paulheim, 2012

- Type information might be missing for instances too!
- Association rule mining (Apriori algorithm) to **complete the type information** for the data
- Input:** instances and their types
- Algorithm:** Find patterns of co-occurring types

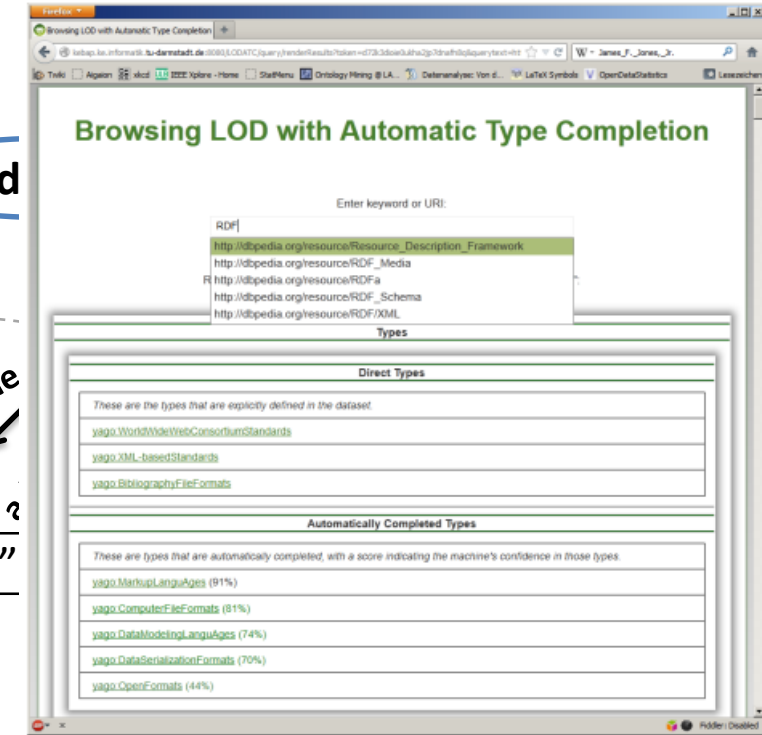
rdfs:subClassOf



“Georgia”

“Heraklion”

name



{ Student , Author } → { PhdStudent }.

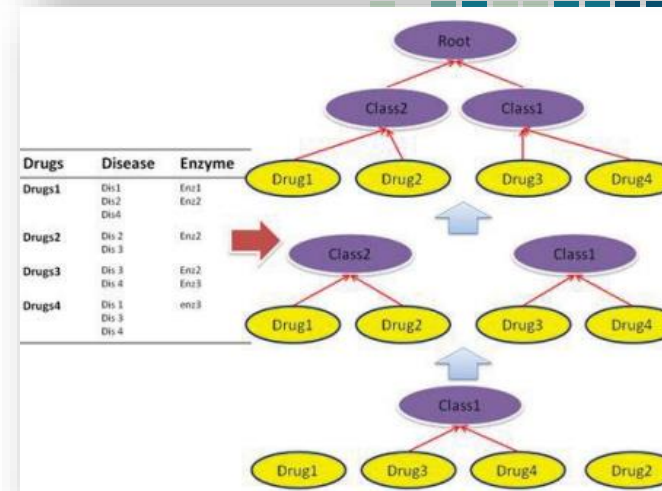
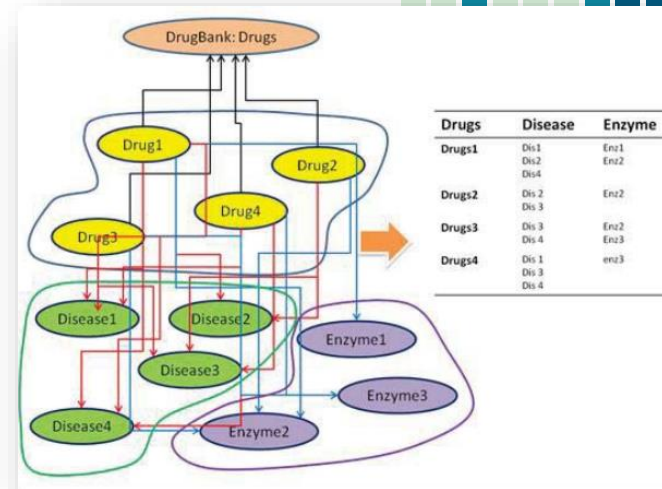
	Mammal	Scientist	Company	Drug	City	Magazine	Person
Steve_Jobs	0	0	1	0	1	1	Person

Nuzzolese et al. 2012

- DBpedia specific approach: [exploit wikilinks](#) to infer the missing types
 1. [Analyze](#) the links from and to the Wikipedia page that the entity is described
 2. [Exploit instances](#) types from datasets linked to DBpedia (owl : sameAs)
 3. Use [K-NN to predict the types](#) of an entity, based on the characteristics of the related types
- Show the bias that can be induced by the incomplete ontological coverage of the DBpedia ontology

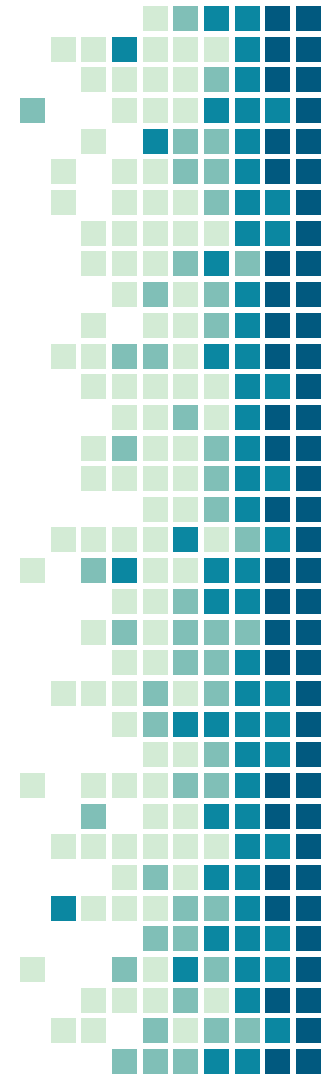
Zong et al. 2012

- Use a **hierarchical clustering** algorithm to construct a hierarchy of types declared in RDF datasets.
- The **Vector Space Model (VSM)**, is used as well as the **Jaccard Coefficient** to formalize the structure of the hierarchy after preprocessing the data.
- The method does **not discover the types**, but **discovers only the hierarchical links** between types



DL-Learner (Bühmann et al. 2018)

- A platform for facilitating the **implementation and evaluation of supervised structured machine learning methods**
- An approximation of the solution is retrieved, the degree of which is managed by thresholds representing the tolerance to noise/errors
- A group of the learning algorithms focus on knowledge base enrichment.
 - These algorithms **analyze the instance data** in order to enrich or revise the existing schemata in a semi-automatic manner.
 - The process is not restricted to classes but can also be applied to object and data properties.
 - The generated axioms consider the **class hierarchy as well as the domain and range** of a property.

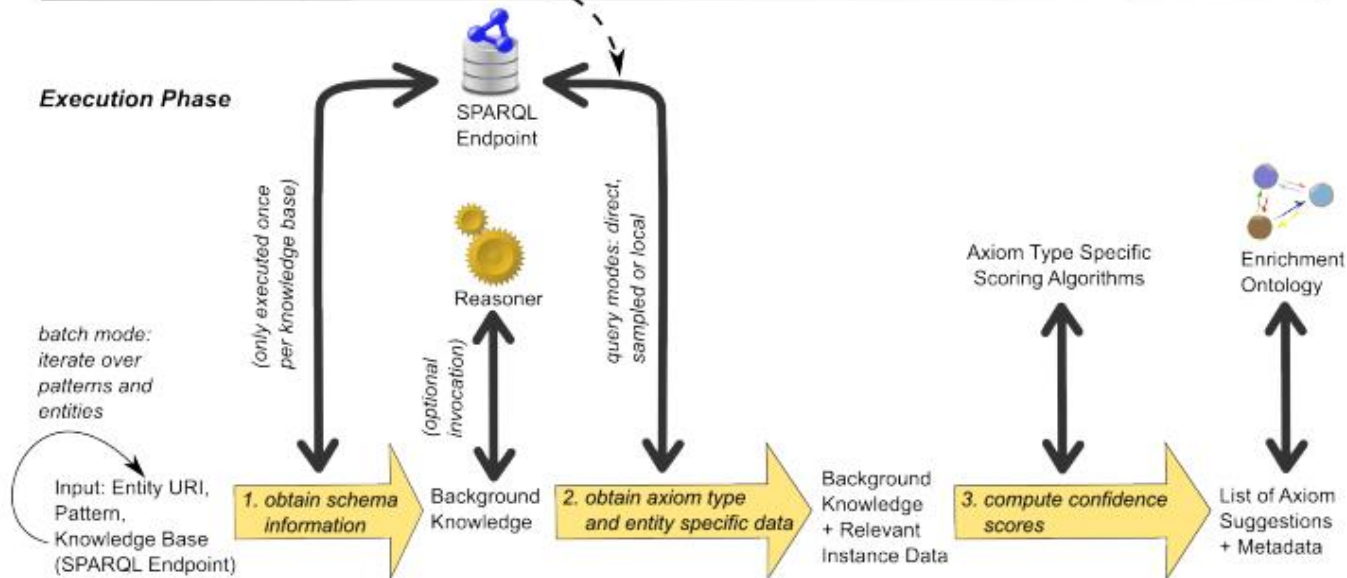


DL-Learner (Bühmann et al. 2018)

Preparation Phase

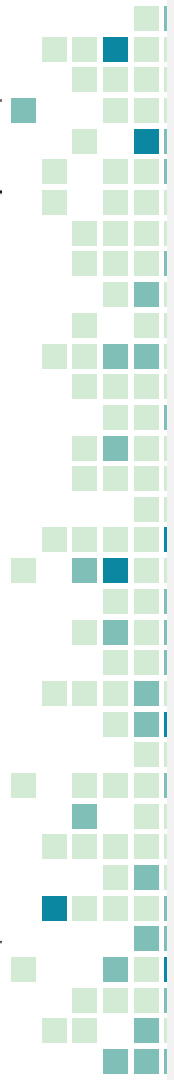


Execution Phase



	Inputs		Techniques	Outputs	Pros & Cons
	Data description	Setting			
Volker et al. [95]	RDF data: Incoming and outgoing properties, <i>rdf : type</i> statements	Support and confidence thresholds for association rules	Unsupervised Learning using association rules discovery algorithm	Semantic links, hierarchical links, OWL axioms	+ Unsupervised – Builds a transaction table for discovery of association rules that can be very large
Paulheim [70]	RDF data: Incoming and outgoing properties, part of <i>rdf : type</i> statements	Support and confidence thresholds for association rules	Unsupervised Learning using association rules discovery algorithm	Types (additional <i>rdf : type</i> statements)	+ Multiple typing + Online processing + Scalable
Nuzzolese et al. [68]	RDF/OWL data: Wikilinks, <i>owl : sameAs</i> statements	Number of neighbors <i>K</i> in <i>K-NN</i>	Supervised learning using <i>K-NN</i>	Types (additional <i>rdf : type</i> statements)	+ Exploits <i>owl : sameAs</i> resources – Specific to <i>DBpedia</i>
Zong et al. [110]	RDF data: Incoming and outgoing properties, <i>rdf : type</i> statements	No required setting	Unsupervised Learning using Hierarchical clustering - VSM	Hierarchical links (<i>rdfs : subclassOf</i>)	+ Apply to biomedical datasets extracted from relational databases – Build the hierarchy on the provided concepts
Bühmann et al. <i>DL-Learner</i> [17,18]	RDF(S)/OWL data: Incoming and outgoing properties, part of schema (<i>rdf : type</i> , <i>rdfs : domain</i> , <i>rdfs : range</i> , <i>rdfs : subclassOf</i>), OWL axioms for reasoning if provided	Threshold for candidate axioms	Supervised Learning using different algorithms	Additional OWL axioms with scores	+ Scalability enhancements through statistical sampling

	Inputs		Techniques	Outputs
	Data description	Setting		
Paulheim et al. <i>SDType</i> [72]	<p>RDFS data: Incoming properties, part of schema (<i>rdf : type</i>, <i>rdfs : domain</i>, <i>rdfs : range</i>, <i>rdfs : subclassOf</i>)</p>	Confidence threshold for a type	Analyze distribution of properties on types	Types (Additional <i>rdf : type</i> statements with a degree of confidence)
Fang et al. [29]	<p>RDF data: Incoming and outgoing properties, <i>dcterms : subject</i>, part of <i>rdf : type</i> statements</p>	Threshold for candidate selection	Analyze distribution of categories on types	Types (Additional <i>rdf : type</i> statements)



SDType (Paulheim & Bizer, 2013)

```
- ?x a ?t1. ?t1 rdfs:subClassOf ?t2 entails ?x a ?t2  
- ?x ?r ?y . ?r rdfs:domain ?t entails ?x a ?t  
- ?y ?r ?x . ?r rdfs:range ?t entails ?x a ?t
```

- A type inference heuristic, able to handle **noisy** and **incorrect** data
- It exploits **links** from and to an instance as indicators for the resource's types
- Type inference rules are exploited
- Exploiting only instance links are not always valid → a single irrelevant triple is enough to infer an incorrect type.
- SDType also considers the relevance of a property to a type → Weighted voting approach that considers many links, trying to avoid the propagation of errors of irrelevant instances.

subject	predicate	object
dbpedia:Mannheim	dbpedia-owl:federalState	dbpedia:Baden-Württemberg
dbpedia:Steffi:Graf	dbpedia-owl:birthPlace	dbpedia:Mannheim
...

① Input data

resource	type
dbpedia:Mannheim	dbpedia-owl:Place
dbpedia:Mannheim	dbpedia-owl:Town
...	...

② Compute basic distributions

resource	predicate	frequency
dbpedia:Mannheim	dbpedia-owl:federalState	1
dbpedia:Mannheim	dbpedia-owl:birthPlace ⁻¹	140
...

type	apriori probability
dbpedia-owl:Place	0.3337534
dbpedia-owl:Town	0.0523772
...	...

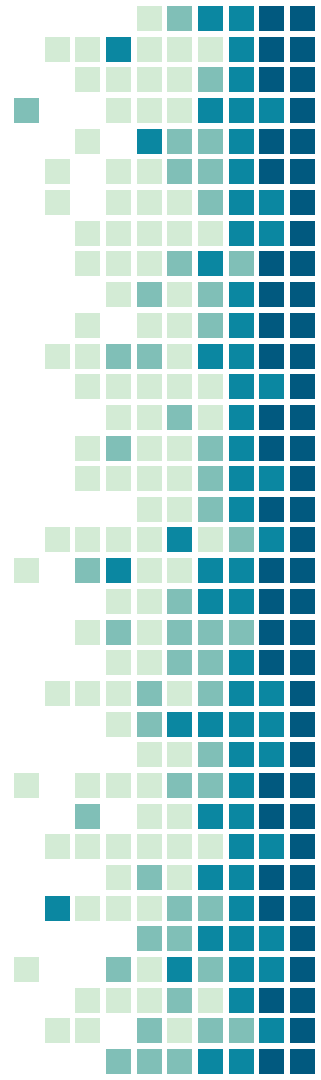
③ Compute weights and conditional probabilities

predicate	weight
dbpedia-owl:federalState	0.3337534
dbpedia-owl:birthPlace ⁻¹	0.0523772
...	...

predicate	type	probability
dbpedia-owl:federalState	dbpedia-owl:Place	1.0000000
dbpedia-owl:birthPlace ⁻¹	dbpedia-owl:Town	0.1760390
...

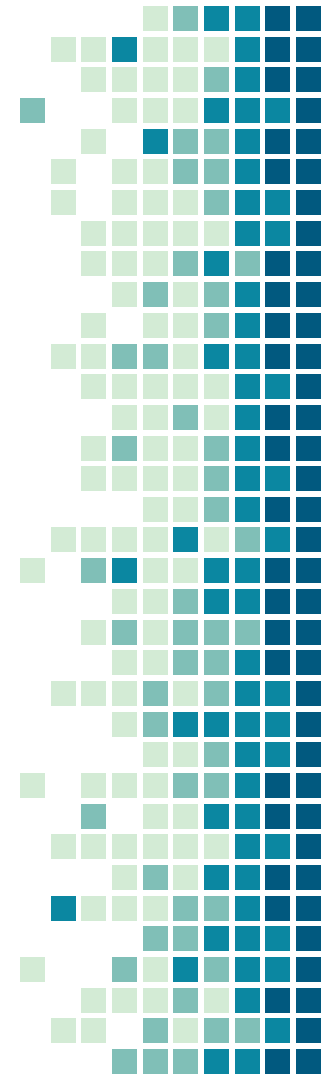
④ Materialize missing types

resource	type	score
dbpedia:Heinsberg	dbpedia-owl:Place	0.8856929
dbpedia:Heinsberg	dbpedia-owl:PopulatedPlace	0.8110996
...



Fang et al.

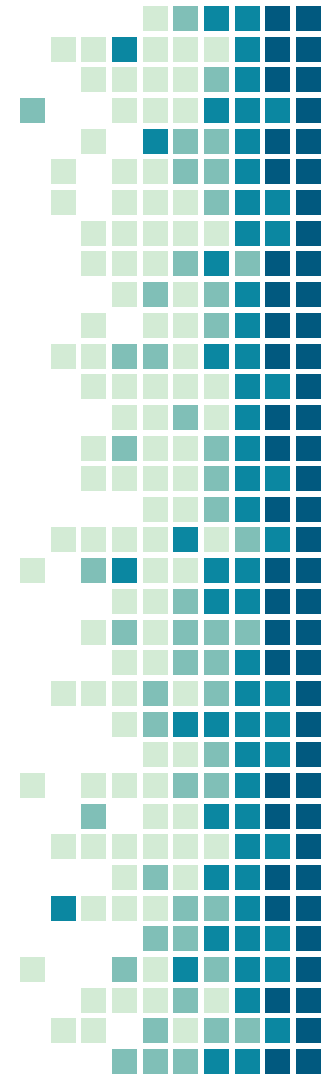
- Focus on DBPedia
- Identify the type of an RDF instance based on its category information provided through `dcterms:subject` declarations.
- **Algorithm:**
 1. The **statistical distribution of each category** is first calculated on all types.
 2. Then, for a given instance, **candidate types** are generated according to the distribution probability of its category.
 3. Finally, the correct **type is identified based on the probability of distribution, keywords in the category, and summary of the instance.**



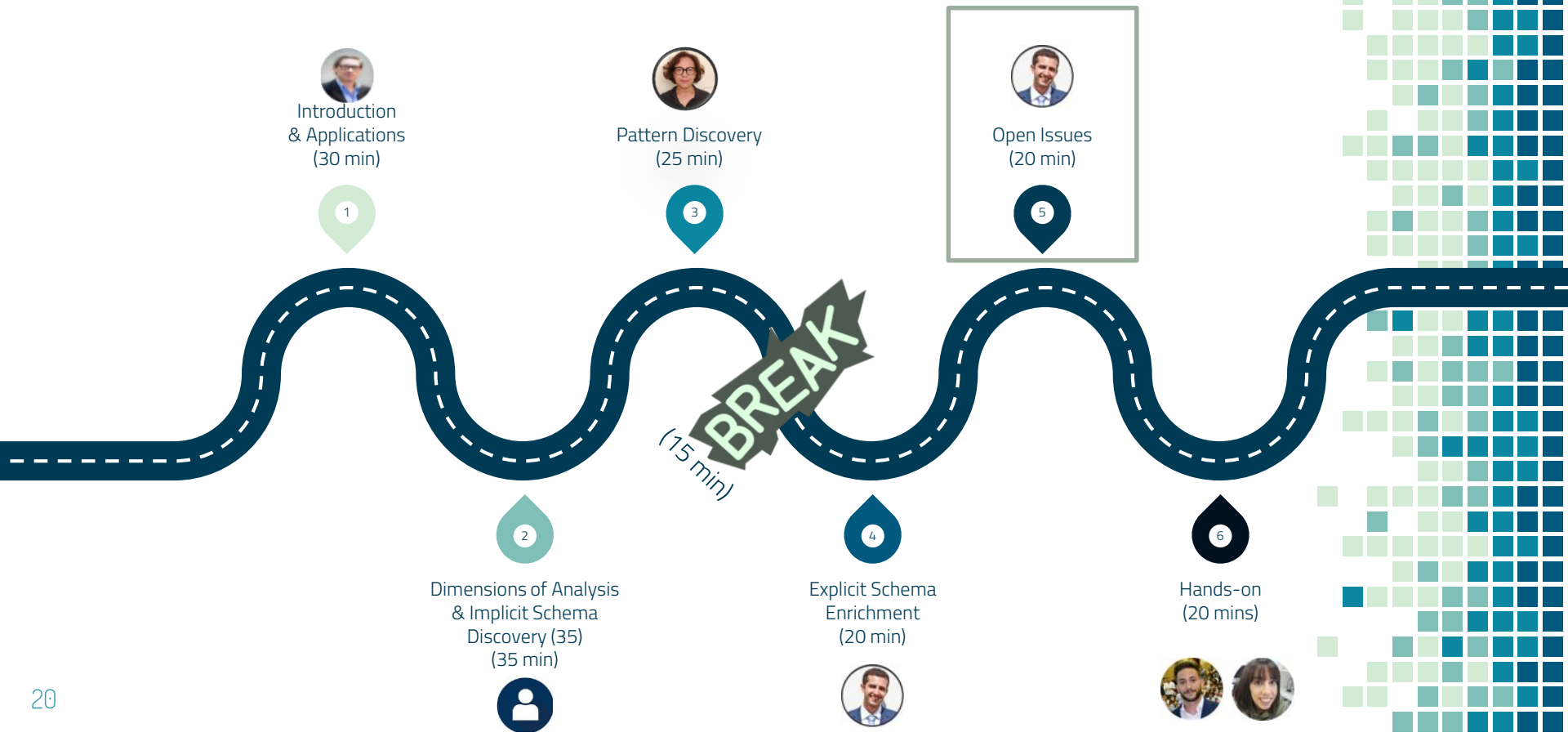
Inputs		Techniques	Outputs	Pros & Cons	
Data description	Setting				
Paulheim et al. <i>SDType</i> [72]	RDFS data: Incoming properties, part of schema (<i>rdf : type</i> , <i>rdfs : domain</i> , <i>rdfs : range</i> , <i>rdfs : subclassOf</i>)	Confidence threshold for a type	Analyze distribution of properties on types	Types (Additional <i>rdf : type</i> statements with a degree of confidence)	<ul style="list-style-type: none"> + Not restricted to classes but can also be applied to object and data properties + Scalable + Implements a weighted voting approach - Extracted rules might not be valid
Fang et al. [29]	RDF data: Incoming and outgoing properties, <i>dcterm : subject</i> , part of <i>rdf : type</i> statements	Threshold for candidate selection	Analyze distribution of categories on types	Types (Additional <i>rdf : type</i> statements)	<ul style="list-style-type: none"> + Scalable - Specific to <i>DBpedia</i>

Takeaways

- All approaches that enrich the existing schema require specific **statements on the schema already to exist**
- These approaches enrich the existing schema **with different additional declarations**
- Most of these approaches **require the specification of parameters**
 - the support and confidence threshold for association rules, the number of neighbors and the confidence threshold for a type
- **Cannot enrich data with types that do not exist in the dataset**

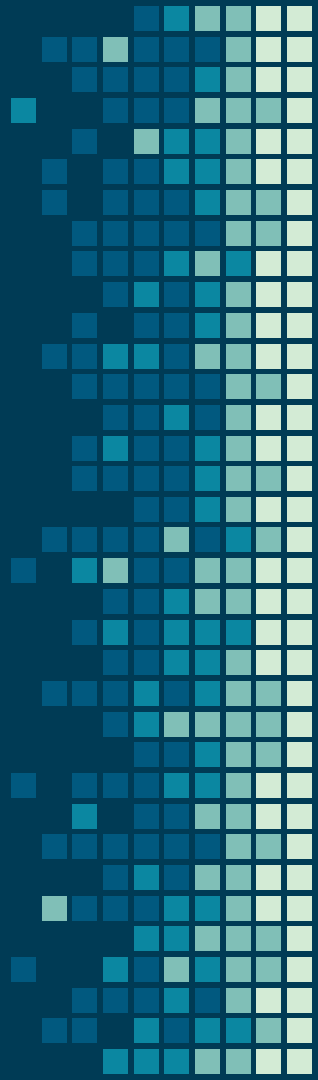


ROADMAP



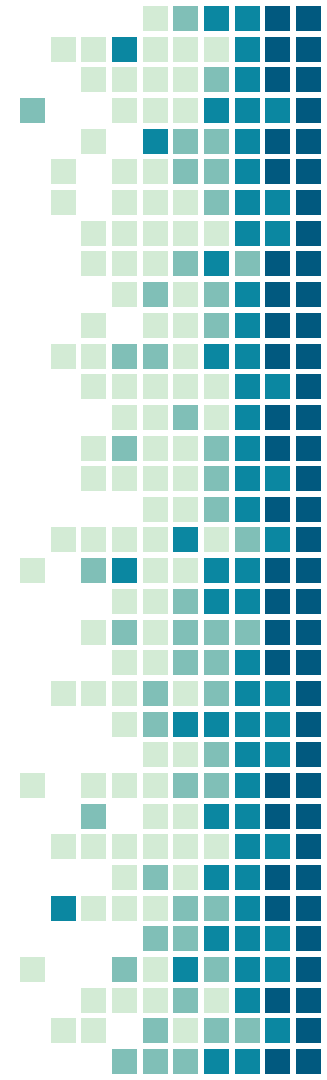


Open Issues



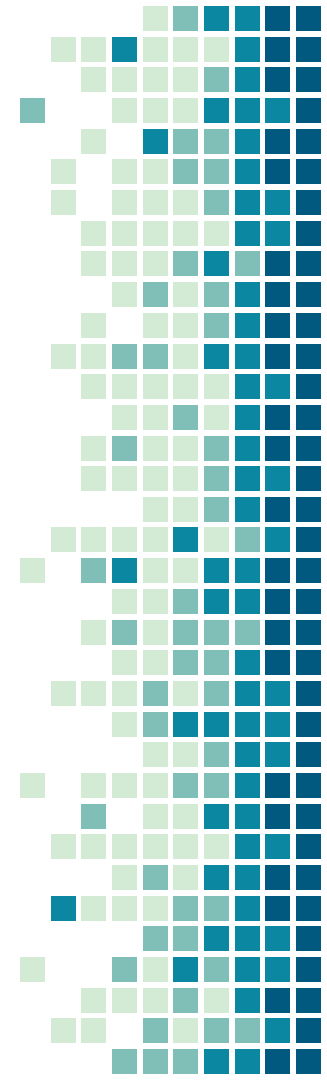
Open Issues

- Which issues?
- Why is it important to address these issues?
- To which extent have they been dealt with so far?
- What are the possible future research directions?



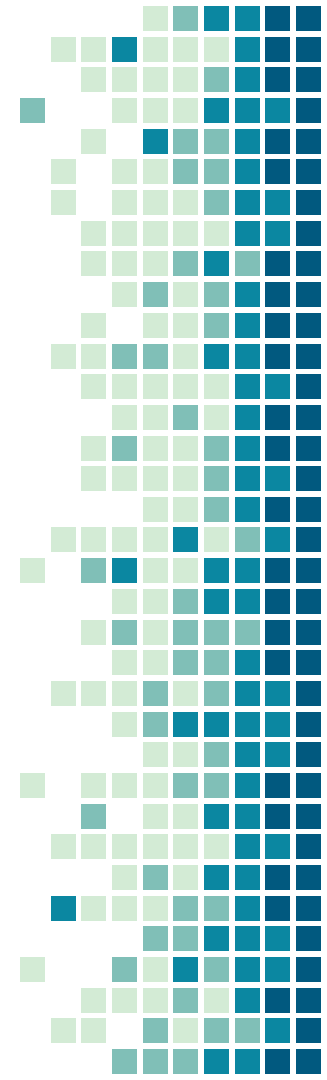
Hybrid approaches currently missing (1/2)

- Schema enrichment approaches require explicit schema statements in the dataset to enrich and complete them.
 - → However, these statements are sometimes completely missing.
- On the other hand, approaches that discover the implicit schema of a linked dataset
 - → do not exploit potentially available declarations on the schema.



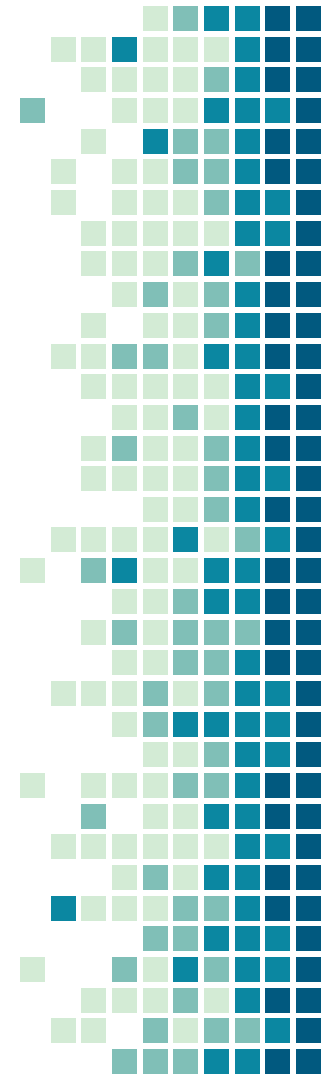
Hybrid approaches currently missing (2/2)

- **Solution:**
 - exploit the available typing information
 - introduce new types where missing
- Existing literature **lacks a hybrid approach**
 - Explicit statements on the schema could be used to guide implicit schema discovery.
 - They can also be used to validate the discovered schema



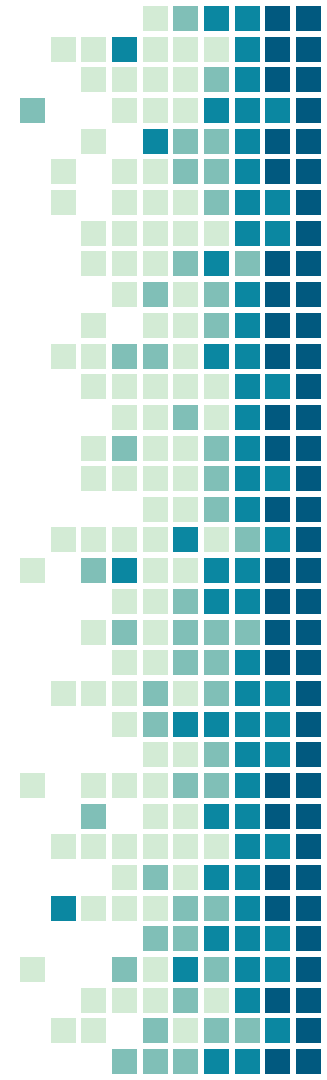
Incremental opportunities (1/3)

- Designing incremental algorithms for schema discovery is also essential as **many datasets are not static, but are daily updated**
 - enable the unobtrusive discovery of the types of the new instances and the potential introduction of new ones



Incremental opportunities (2/3)

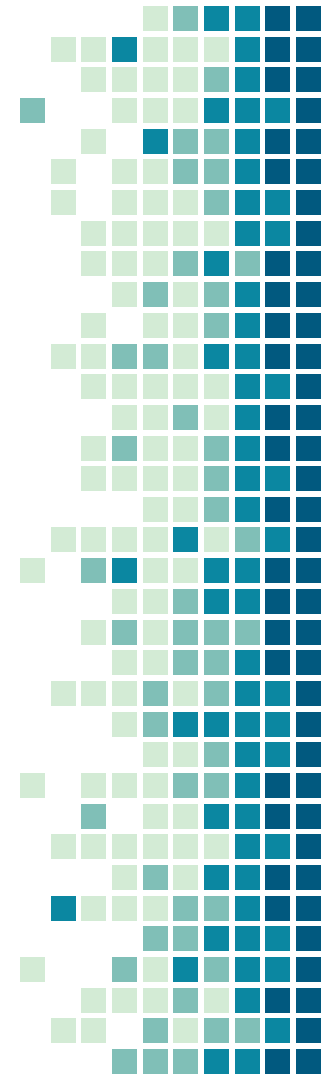
- Approaches using **statistics**
 - update the statistics to consider a new update in the data source
- Approaches **discovering frequent patterns** using association rules
 - explore the possible ways of achieving their incremental discovery



Incremental opportunities (3/3)

Assign incrementally a type to a new instance

- A supervised learning step which could be applied for any implicit schema discovery approaches grouping instances
 - Adding a classification step for new instances requires a training set & the result is very dependent on the content of the training set.
- Adapt [Locality Sensitive Hashing](#) (LSH) for schema discovery
- COBWEB that assigns incrementally types to new instances is not [stable](#)
- What about incremental algorithms like [CLASSIT](#) or [ARACHNE](#)?



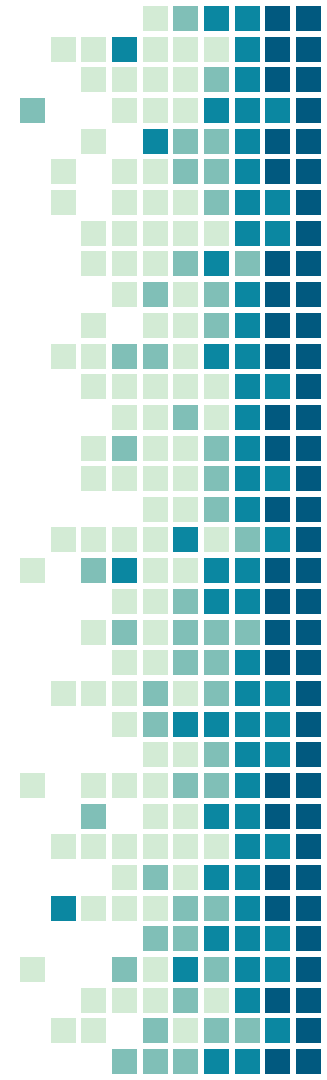
Scaling up (1/3)

- Existing approaches are **not suitable for massive datasets**
- Approaches proposed for implicit schema discovery are based on **groupings**
 - Requires an **exhaustive comparison between the instances**, and as such are inefficient



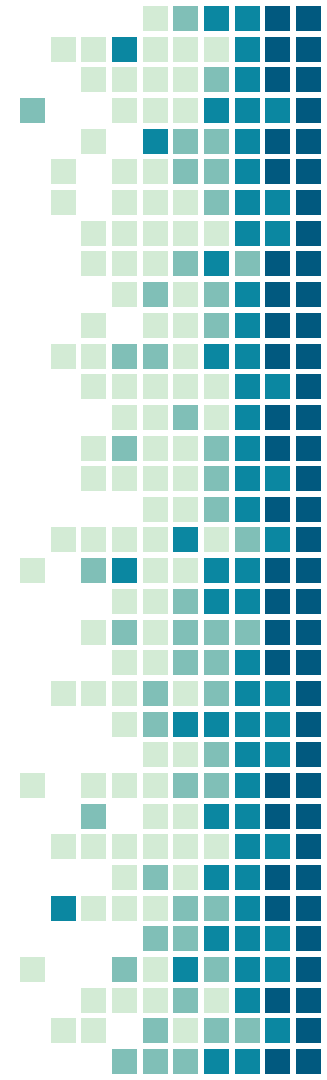
Scaling up – Solutions (2/3)

- Potential solutions
 - **Structural compression:** instances of the same type have a similar structure and many instances have exactly the same structure
 - **SC-DBSCAN:** a distributed, density-based clustering algorithm dealing with scalability issues
 - Extract patterns first then apply distributed clustering using SPARK
 - Could explore strategies for distributing effectively the patterns through the nodes with minimal replication



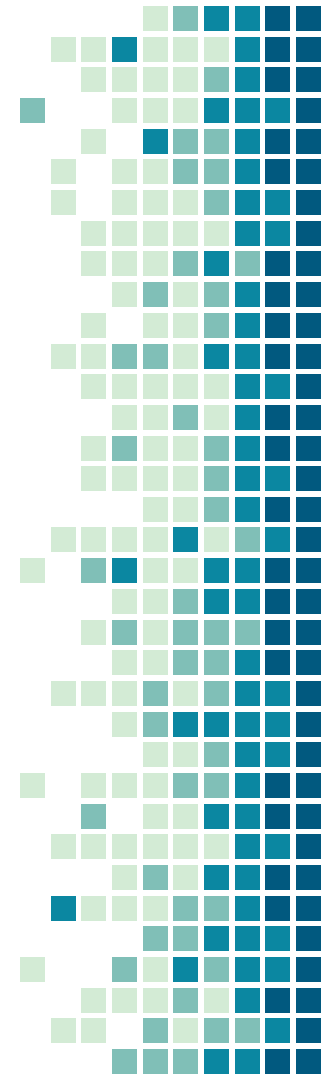
Scaling up – Solutions (3/3)

- Bisimulation with the [map/reduce paradigm](#):
 - bisimulation gives a schema as a larger path plan
- Explore other methods which [do not imply an exhaustive comparison](#) between the instances such as Locality Sensitive Hashing (LSH)
 - the [pattern](#) of an instance could be considered as a first key to allow the distribution of instances
 - then the [hash value of each pattern](#) could be considered as a second key to allow the processing of the patterns with Map/Reduce to generate the types
- Scalable techniques from [entity resolution](#) could also be exploited



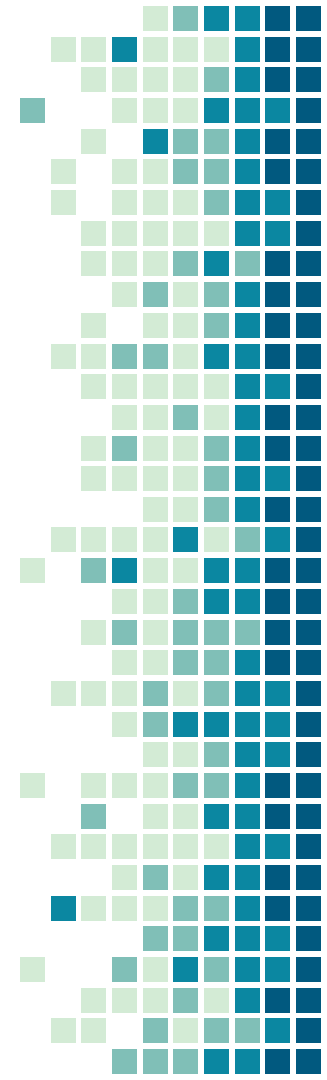
Online schema discovery

- It is not always possible for a **remote data source to be copied locally**
- it may be **expensive in terms of memory storage** and to keep the local source up to date with the remote source.
- Challenges:
 - Restrictions by the server such as **the timeout set on the query execution** and the limitation on the **number of queries that can be sent** in order to avoid server and network overload.
 - Discovering **exact pattern** online is more challenging because this requires sending more queries to get an exact result.
 - It would be useful to have an online approach for implicit or hybrid schema discovery.
 - We could, for example, **rely on some provided schema-related information or statistics** to query a remote source in order to discover the schema



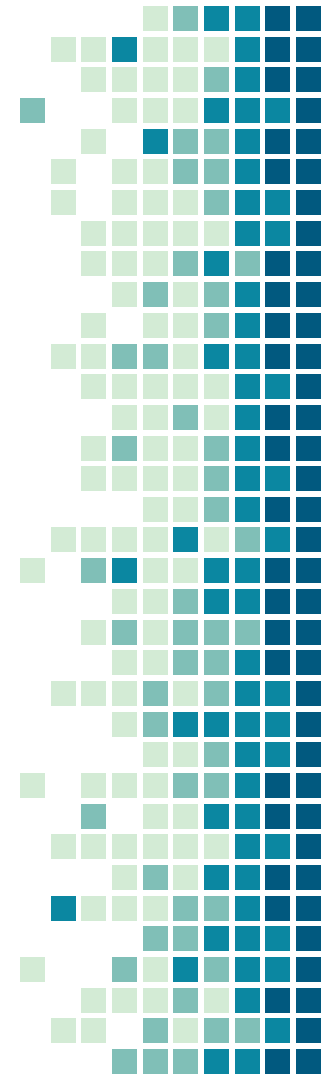
Automatic Setting (1/2)

- Many schema discovery approaches rely on machine learning techniques
- Several techniques require the specification of parameters
- The specification of the parameters is not obvious
 - For example, it is not possible to define the number of required classes before the schema discovery task.
- **Potential solution:**
 - The approach presented in (Kellou-Menouer & Kedad, 2016) propose the **automatic detection of the similarity threshold for grouping similar instances**.
 - This proposal could be applied for any schema discovery approaches which require the specification of the similarity threshold.
 - However, it needs **a first pass on the data** to automatically detect this setting.

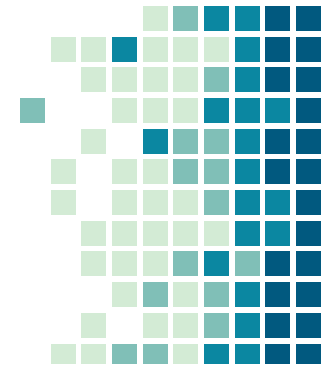


Automatic Setting (2/2)

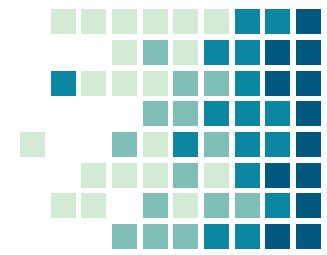
- HiINT (Kardoulakis et al, 2021)
 - LSH functions for an implicit schema discovery
 - Type information if available for an explicit schema enrichment
 - Incrementally identifies the patterns of the various instances, thus reducing instance comparisons to pattern comparisons.



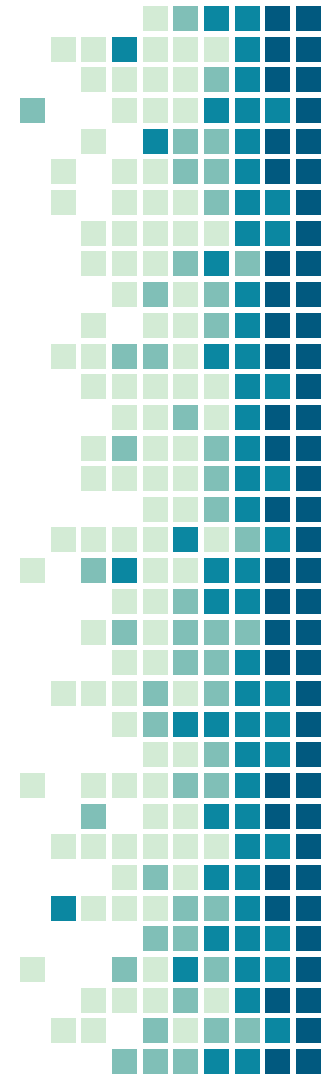
Overview



Approaches	Hybrid	Incrementality	Stability	Scalability	Online	Automatic setting
Implicit schema discovery [14,15,22,24,25,35,48,50, 54,60,66,67,78,91,101]	–	[50,101]	All except [101]	[14,15,78]	–	[50]
Explicit schema enrichment [17,18,29,68, 70,72,95,110]	–	–	All	[17,18,29,70,72]	[70]	–
Structural patterns discovery exact [11,51,52,57,77] approximate [8,9,12,13,21,42,100,109]	all except [11,100]	[8,9,11]	All	All except [21,42,100,109]	[12,13,51,52]	–



Takeaway



- An important challenge for schema discovery is
- **finding a single approach that could tackle these different issues at the same time.**
- The work in schema discovery gains more importance as the available data sources become larger and more connected, as it provides a natural way to understand their contents.

Conclusion



- We focus on approaches for schema extraction,
 - providing a taxonomy to help readers understand the various dimensions,
 - approaches discovering the **implicit** schema
 - approaches enriching the **explicit** schema
 - approaches discovering the **structural patterns** of instances of a dataset
 - shedding light to the various works in the area.
- Although being an active research area recently, **many challenges have still to be tackled in schema discovery**.
- Automatic schema discovery has attracted significant interest over the latest years, and given the many applications and the open topics in the domain, we expect many more works to appear in the following years.

THANKS!

Any questions?

You can find us at

[https://users.ics.forth.gr/~kondylak/
iswc_2022_tutorial/](https://users.ics.forth.gr/~kondylak/iswc_2022_tutorial/)