User-oriented exploration of semi-structured datasets

Nelly Barret

4th year PhD student
Supervised by Ioana Manolescu
Inria Saclay and Institut Polytechnique de Paris

January 25, 2024

Context: data is the new gold (1/3)

Context: data is the new gold (1/3)

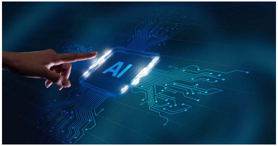
Context: data is the new gold (2/3)

Our digital world comes:

- In **various contexts**: science, health, political life
- At **various scales**: home, city, country, world
- By **different actors**: scientists, businesses, policy makers
- With **different needs**, constraints, abilities

We are **overwhelmed** by (raw) data, we need:

- Data-driven applications
- Data journalism
- Knowledge graphs
- Artificial "intelligence"
- ...



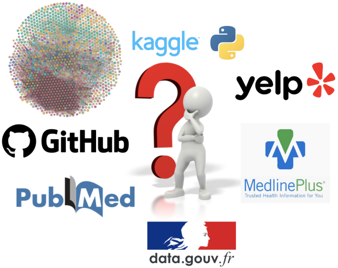
Context: data is the new gold (3/3)

Very **heterogeneous** data:

- Mainly RDF (1K datasets in the LODC)
- Also: XML, JSON, relational, Property Graph...

Detection of **entities** of interest:

- People, Place, email, ...



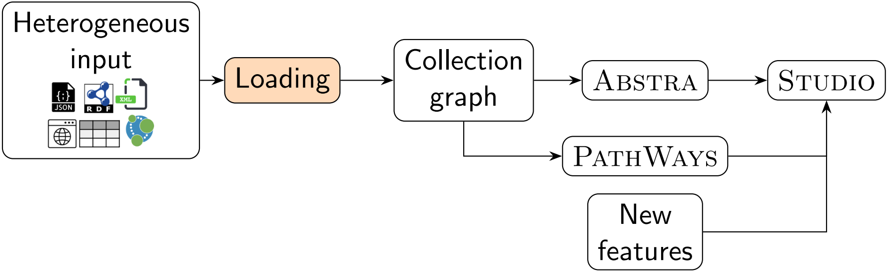
With heterogeneous data, users need:

1. A **uniform** integration, view over the data
2. **Efficient** algorithms and applications
3. A global **understanding**, description
4. Interesting **entity connections**

Create a unique data graph

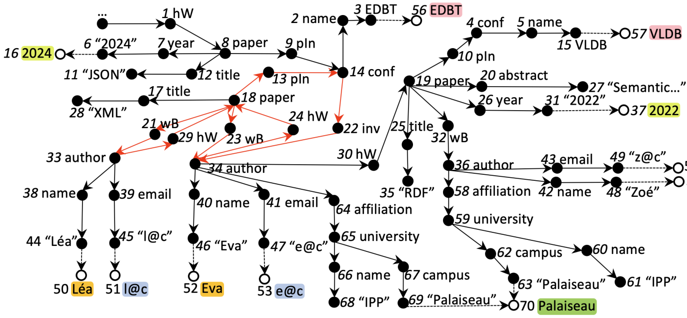
“A **uniform** integration, view”

Angelos Anadiotis — IPP, EPFL
Oana Balalau — Inria, IPP
Ioana Manolescu — Inria, IPP
et al... — INSEC, ...



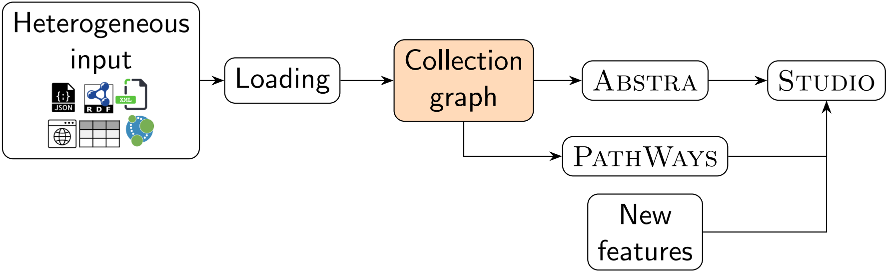
Graph construction

- Ingest any dataset into a **directed graph** (•, →)
- Extract **named entities**, NEs, from the graph values (○, - - →):
  - Temporal: date, time reference
  - Web: URI, email address, hashtag, Twitter citation
  - Complex entities: People, Place, Organization



Create a compact representation of the data graph

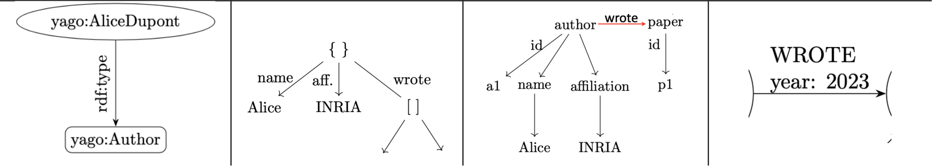
"**Efficient** algorithms and applications"



A uniform view of data formats

Each data format has its own specificities:

RDF (URIs)	JSON (ϵ labels)	XML (ID-IDREF)	PG (edge attrs)



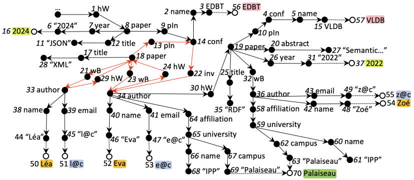
But, we **encode** the same logic:

- **Record**: piece of data, an object
- **Value**: record with no children
- **Same-kind records**: schema or “intuitive” order
- **Relationship**: how records relate

Compact representation (summarization)

Three **equivalence relations**:

- Per **label** for XML
- Per **path** for JSON and relational data
- Per **type or edge neighbourhood** for RDF and PG [GGM20]



$$\left\{\begin{array}{lc} EC_1 & \{N_8, N_{18}, N_{19}\} \\ EC_2 & \{N_{33}, N_{34}, N_{36}\} \\ EC_3 & \{N_4, N_5\} \\ EC_4 & \{N_2, N_5, N_{38}, N_{40}, N_{42}\} \\ EC_i & \dots \end{array}\right\}$$

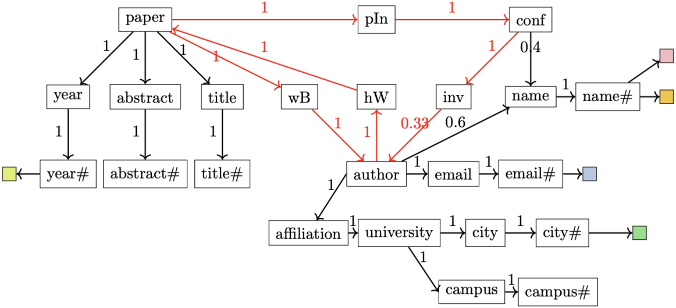
The collection graph

One **collection node** for each equivalence class

One **collection edge** $C_s \to C_t$:

- Between two collection nodes if a data edge exists
- **Edge transfer factor**: $\frac{|C_t \to C_s|}{|C_t|}$
- **At-most-one**: 1:1 cardinality

An **entity profile** for each **leaf collection node**: presence of entities



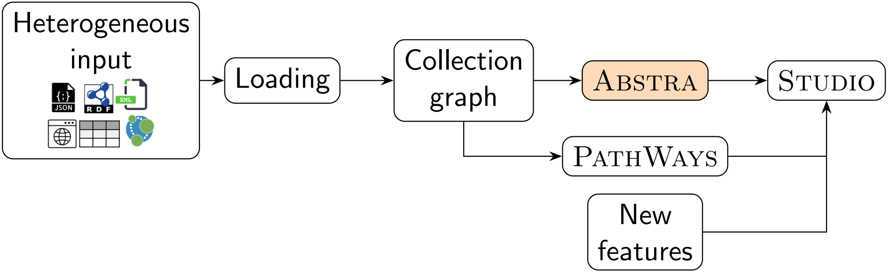
Build an Entity-Relationship model

"A global **understanding, description**"

Nelly Barret — Inria, IPP

Prajna Upadhyay — Inria

Ioana Manolescu — Inria, IPP



ABSTRA: get an overview of the data

Problem statement

How to produce a compact and expressive description out of any dataset?

1. A **high-level, global description**, easy to grasp for **NTUs**
2. Focus on the data **meaning** more than the **syntax**

$\Longrightarrow$ Retrieve / build **the Entity-Relationship model** behind any dataset

ABSTRA: get an overview of the data

Problem statement

How to produce a compact and expressive description out of any dataset?

1. A **high-level, global description**, easy to grasp for **NTUs**
2. Focus on the data **meaning** more than the **syntax**

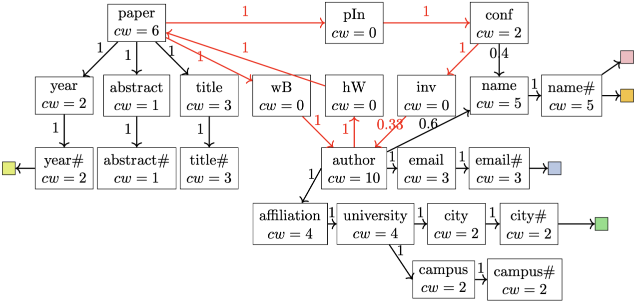
$\Longrightarrow$ Retrieve / build **the Entity-Relationship model** behind any dataset

	Data Summarization	Schema inference	Abstra
Several data formats	×	∼	✓
Content and structure	∼	∼	✓
No syntactic detail	✓	×	✓
First-sight discovery	∼	×	✓

Main collections selection

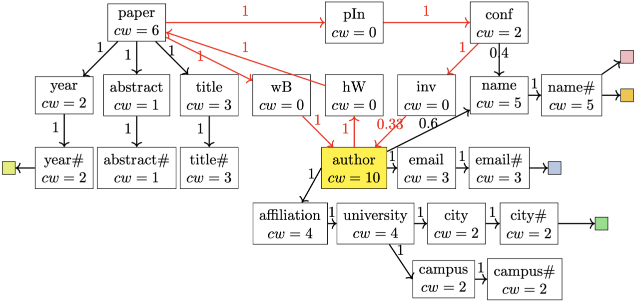
Election of few main collections, representing mostly the dataset

1. Assign a **weight** to each collection
2. While less than E_{max} main collections or data coverage $< cov_{min}$
   1. Pick C^*, the next heaviest collection
   2. Compute the **boundary** of C^*
   3. **Update** the collection graph to reflect the selection of C^*
   4. Recompute the weights



Collection weight

- w_{desc_k}
- w_{leaf_k}
- w_{DAG}
- $w_{PageRank}$
- $w_{dwPageRank}$



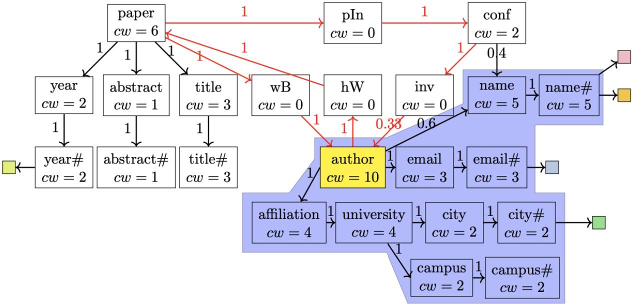
Collections weights, boundaries and graph updates

Collection weight

- w_{desc_k}
- w_{leaf_k}
- w_{DAG}
- $w_{PageRank}$
- $w_{dwPageRank}$

Boundary

- $bound_{desc}$
- $bound_{leaf}$
- $bound_{DAG}$
- $bound_{flood}$
- $bound_{acyclic-flood}$



Collections weights, boundaries and graph updates

Collection weight

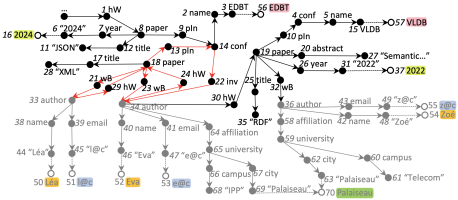
- w_{desc_k}
- w_{leaf_k}
- w_{DAG}
- $w_{PageRank}$
- $w_{dwPageRank}$

Boundary

- $bound_{desc}$
- $bound_{leaf}$
- $bound_{DAG}$
- $bound_{flood}$
- $bound_{acyclic-flood}$

Graph update

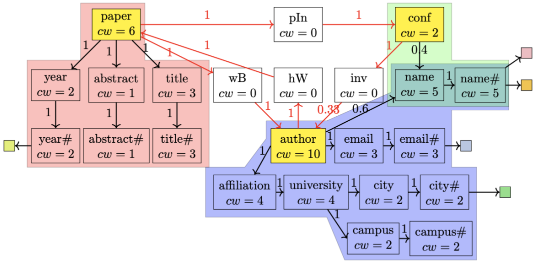
- $update_{boolean}$
- $update_{exact}$



Find relationships between main collections

Possible relationships

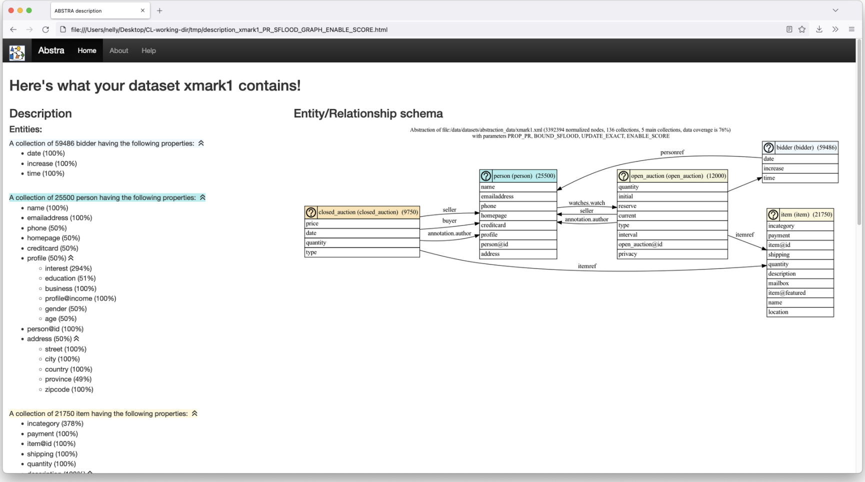
The **set of relationships** connecting a pair of collections is the set of their paths.



- paper → wB → author
- paper → pIn → conf
- author → hW → paper
- conf → inv → author

The final output in ABSTRA

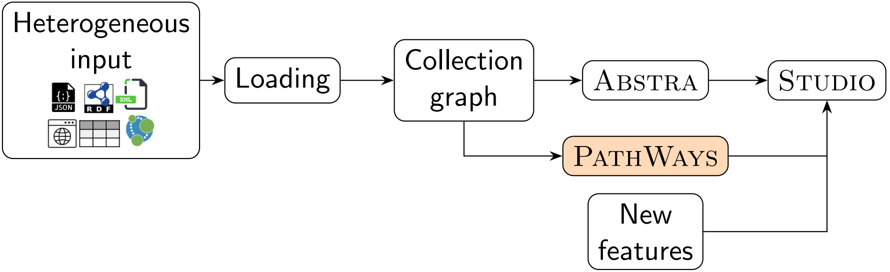
https://team.inria.fr/cedar/projects/abstra/



Enumerate entity paths

"Interesting **entity connections**"

Nelly Barret — Inria, IPP
Antoine Gauquier — IMT
Jia Jean Law — IPP
Ioana Manolescu — Inria, IPP



PathWays: find interesting connections in the data

Problem statement

How to interactively explore entity connections in heterogeneous datasets?

1. No query writing, nor prior knowledge
2. A **tabular, high-level output**, easy to grasp for **NTUs**
3. Do it **efficiently** even if the data graph is large

$\Longrightarrow$ Connect **named entities** (People, Places, ...) **in and across** datasets.

PathWays: find interesting connections in the data

Problem statement

How to interactively explore entity connections in heterogeneous datasets?

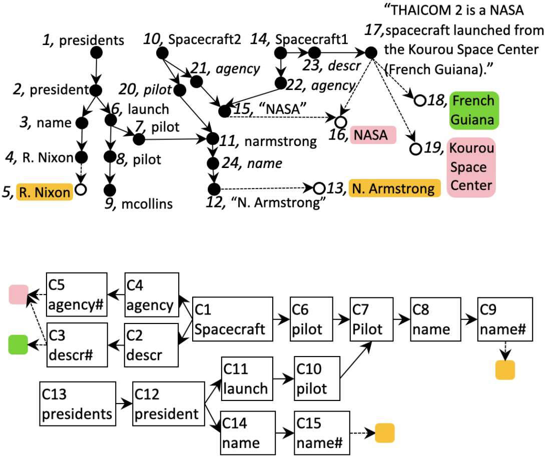
1. No query writing, nor prior knowledge
2. A **tabular, high-level output**, easy to grasp for **NTUs**
3. Do it **efficiently** even if the data graph is large

$\Longrightarrow$ Connect **named entities** (People, Places, ...) **in and across** datasets.

	Keyword search	**Graph query**	**Reachability query**	PathWays
No query writing	✓	×	×	✓
Tabular output	∼	∼	×	✓
Efficient	×	✓	✓	✓

Scenario and terminology

- A **data (entity) path** is a path in the data graph
- A **collection (entity) path** is a path in the collection graph
- The evaluation of a collection path leads to a set of data paths

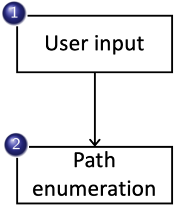


Collection (entity) path enumeration

1. User input

- (τ_1, τ_2); max path length; non-specific connections

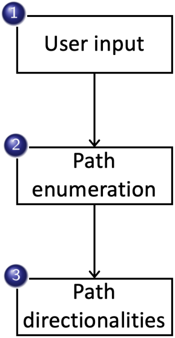
Collection (entity) path enumeration



- (τ_1, τ_2); max path length; non-specific connections

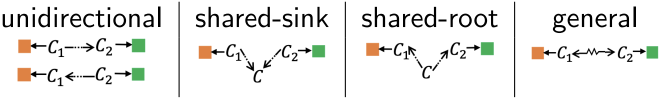
- **Enumerate** all collection paths using the user input
- **Regardless of edge direction**

Collection (entity) path enumeration

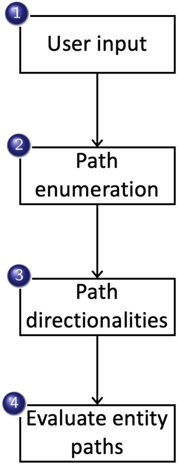


- (τ_1, τ_2); max path length; non-specific connections

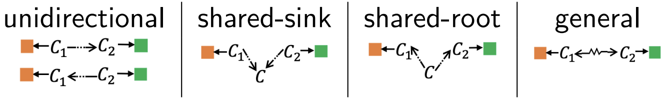
- **Enumerate** all collection paths using the user input
- **Regardless of edge direction**



Collection (entity) path enumeration



- (τ_1, τ_2); max path length; non-specific connections
- **Enumerate** all collection paths using the user input
- **Regardless of edge direction**



- **Evaluate** selected collection paths into data paths

Collection paths interestingness

- Many enumerated paths are non-interesting
- Humans can "see/feel it"

- How to quantify the interestingness of a path?
  1. Compute the **reliability** r of its extracted NEs
     $\rightarrow$ How many NEs of a given type are extracted from a leaf collection?
  2. Compute the **force** f of each structural collection edge
     $\rightarrow$ What is the maximal data edge cardinality behind a collection edge?
  3. Rank paths on their reliability, then their force
  4. Take the top-k or those having $r \geq \theta$

$$\text{P} \xleftarrow{1} \# \xleftarrow{1} \text{name} \xleftarrow{1} \text{author} \xrightarrow{1} \text{affiliation} \xrightarrow{1} \# \xrightarrow{0.9} \text{O}$$

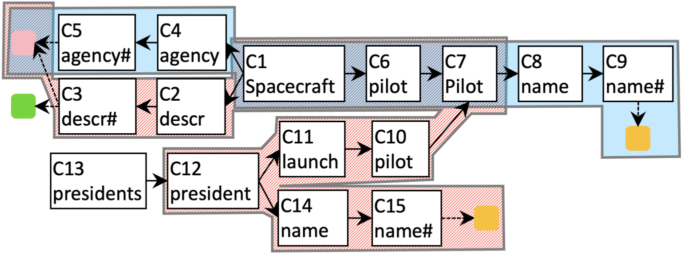
$$\text{P} \xleftarrow{1} \# \xleftarrow{1} \text{name} \xleftarrow{1} \text{author} \xleftarrow{0.02} \text{list} \xleftarrow{1} \text{article} \xrightarrow{1} \text{journal} \xrightarrow{1} \# \xrightarrow{0.4} \text{O}$$

$$\text{P} \xleftarrow{0.04} \# \xleftarrow{1} \text{title} \xleftarrow{1} \text{article} \xrightarrow{1} \text{list} \xrightarrow{1} \text{author} \xrightarrow{1} \text{affiliation} \xrightarrow{1} \# \xrightarrow{0.9} \text{O}$$

Optimized data paths computation (1/2)

Assumption: enumerated collection paths (largely) overlap

- There exist common sub-paths between collection paths
- Common sub-paths should be evaluated only once as <u>views</u>
  - → Saves computation time
- Collection paths are rewritten using <u>views</u>
  - → Reduces the number of joins



Optimized data paths computation (2/2)

Greedily select the most profitable views to materialize

Input: collection paths $\mathcal{P}$, candidate views $\mathcal{V}$

Output: a set of views, a set of rewritings

1. While there are some $v \in \mathcal{V}$:
   1. For each pair (p, v), compute $ben(p, v) \leftarrow costEval(p) - costEval(p, v)$
   2. Store v_{max}, the view maximizing $ben(v) \leftarrow \sum ben(p, v) - costMat(v)$
   3. For each path p, rewrite it, if possible, using v_{max}

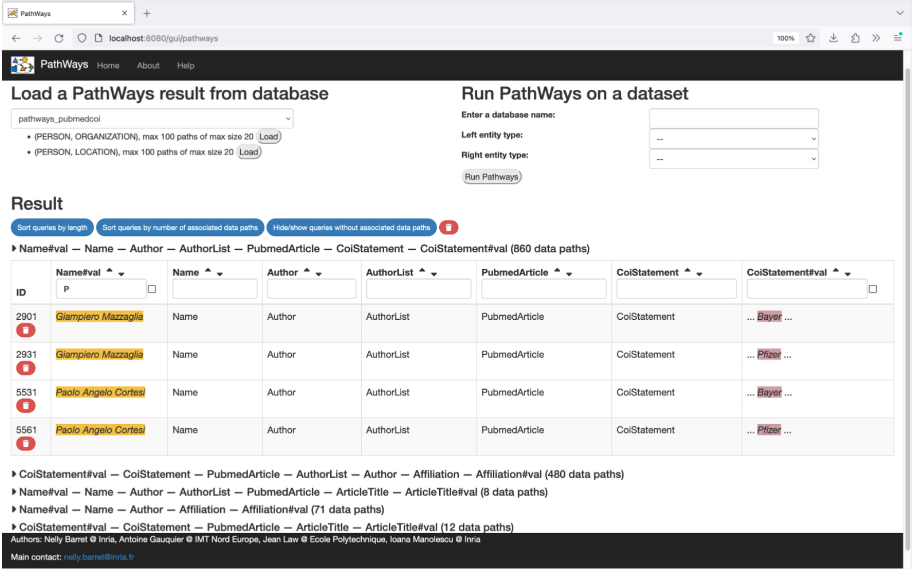
■ ← agency# ← agency ← **Spacecraft** → **pilot** → **Pilot** → name → name# → ■

■ ← agency# ← agency ← **v** → name → name# → ■

```
SELECT le.label, C5.label, C4.label, v.C1label, v.C6label, v.C7label, C8.label, C9.label, re.label
FROM nEntities le, nodes C5, edges C4, view v, edges C8, nodes C9, nEntities re
WHERE le.leafId=C5.id and C4.t=C5.id and C4.s=v.C1id and C8.s=v.C7id and C8.t=C9.id and re.leafId=C9.id
      and le.type = ■ and re.type = ■;
```

Data path results in PathWays

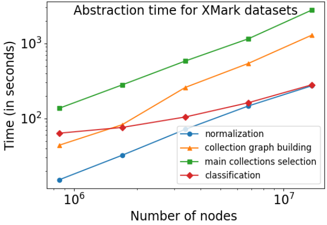
https://team.inria.fr/cedar/projects/pathways/



Quick overview of experiments

On widely-used **open data formats**: JSON, RDF, XML and PG.

Abstra	PathWays

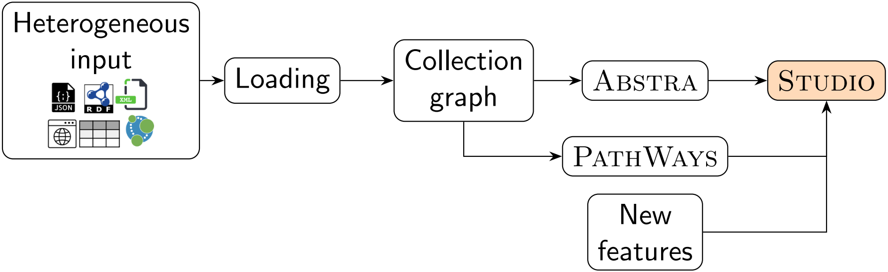


(τ_1, τ_2)	T_0	$T = T_R + T_{Q_{NV}}$	$s = T_0/T$
(τ_P, τ_O)	250.36	4.10	61×
(τ_P, τ_L)	37.29	19.38	2×
(τ_L, τ_O)	151.29	20.47	7×
(τ_P, τ_P)	152.59	44.27	3×
(τ_L, τ_L)	169.64	71.63	2×
(τ_O, τ_O)	317.92	23.24	13×

- User study
- Comparison to schemas

- # paths: 0 to very high
- Filter spurious paths (path interestingness)

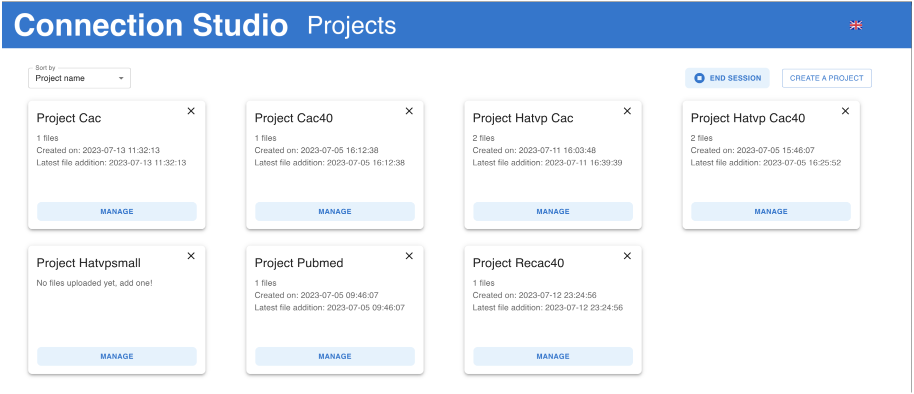
Future work, takeaways and open questions



Future work: Studio

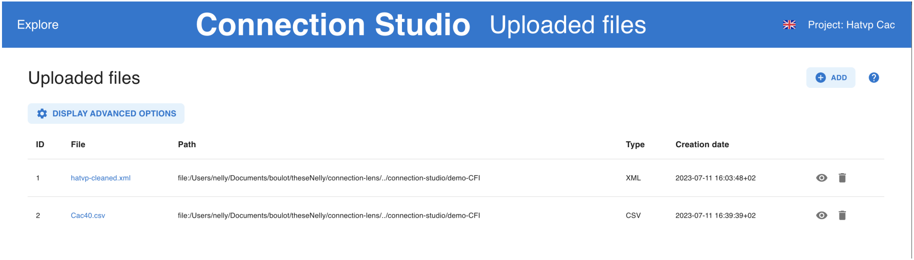
Studio: a data lake for ingesting, querying, cleaning and understanding heterogeneous data

- French media are interested (DataJournos, CFI)



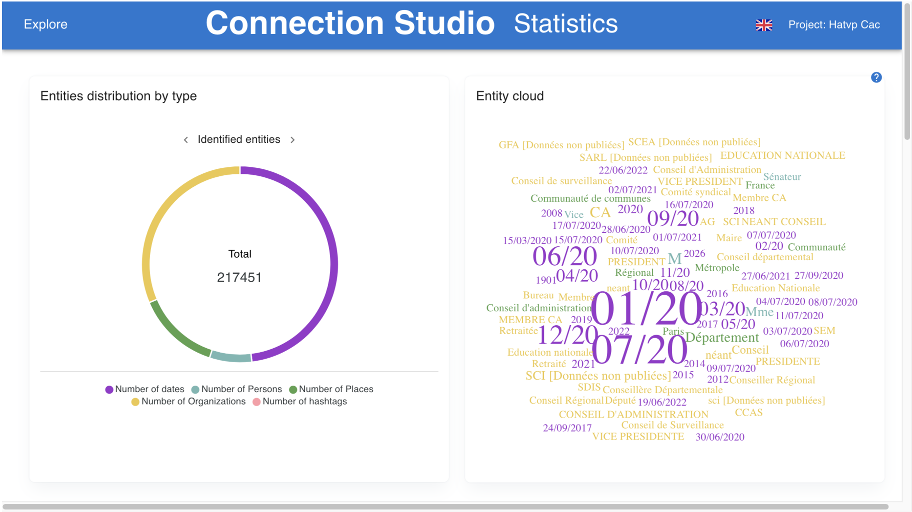
Future work: STUDIO

STUDIO: a data lake for ingesting, querying, cleaning and understanding heterogeneous data



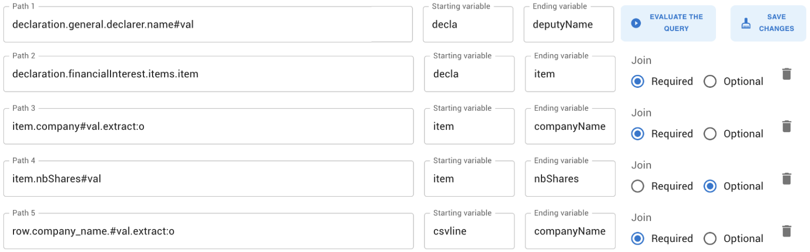
Future work: Studio

Studio: a data lake for ingesting, querying, cleaning and understanding heterogeneous data



Future work: Studio

Studio: a data lake for ingesting, querying, cleaning and understanding heterogeneous data

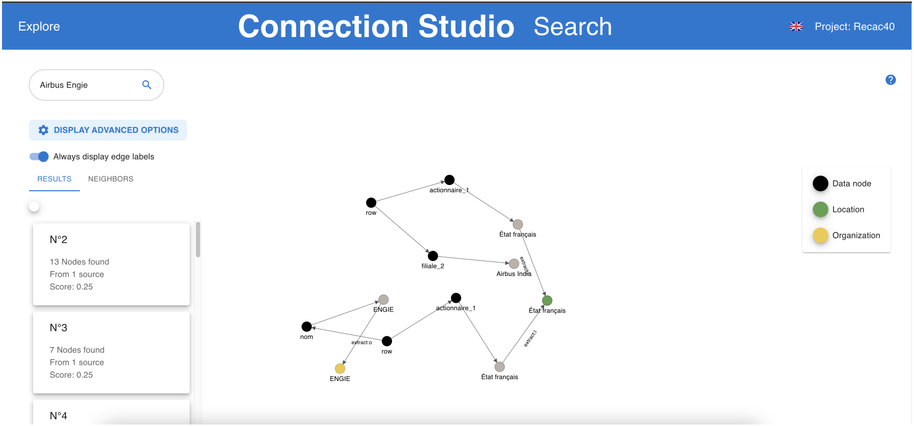


COLUMNS FILTERS DENSITY EXPORT

decla	deputyname	item	companyname	nbshares	csvline
2660	alain pierre marie rousset	2743	sanofi	1200	352
1470	edouard courtial	1511	lvmh	29013	248
1470	edouard courtial	1543	michelin	162179	261

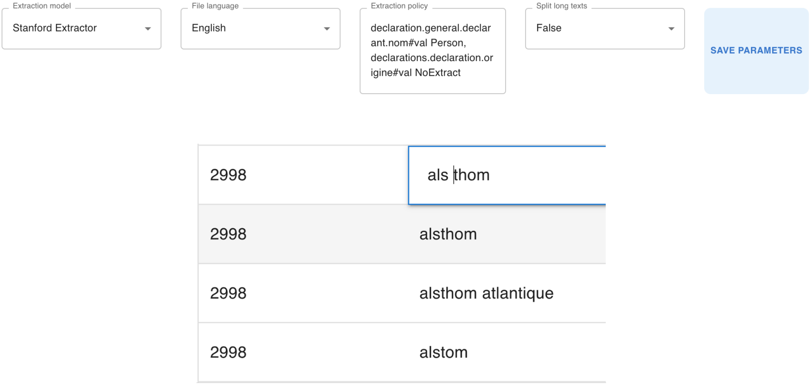
Future work: Studio

Studio: a data lake for ingesting, querying, cleaning and understanding heterogeneous data



Future work: Studio

Studio: a data lake for ingesting, querying, cleaning and understanding heterogeneous data



Takeaways and open questions

- ABSTRA: a dataset abstraction system for heterogeneous data
- PATHWAYS: an entity-focused exploration system
- STUDIO: a user-oriented data lake for data exploration

ABSTRA	PATHWAYS	STUDIO
EDBT 2024	ADBIS 2023	CoopIS 2023

Further opportunities

Nelly BARRET

nelly.barret@inria.fr

https://pages.saclay.inria.fr/nelly.barret/

Inria Saclay & Institut Polytechnique de Paris
Palaiseau

→

→

POLITECNICO
MILANO 1863

→

ÉCOLE CENTRALE LYON · LIRIS · Inria · ENS DE LYON · UNIVERSITÉ DE LYON