

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 (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

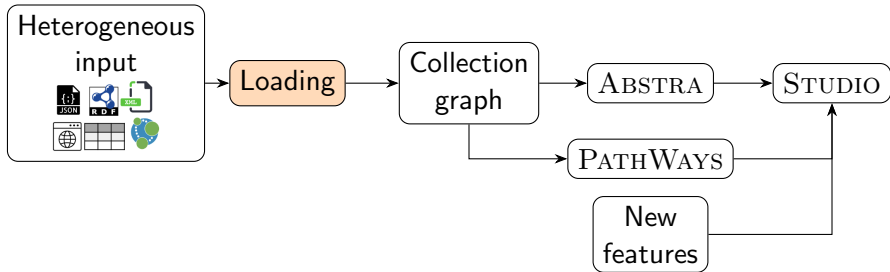
“A **uniform** integration, view”

Angelos Anadiotis
IPP, EPFL

Oana Balalau
Inria, IPP

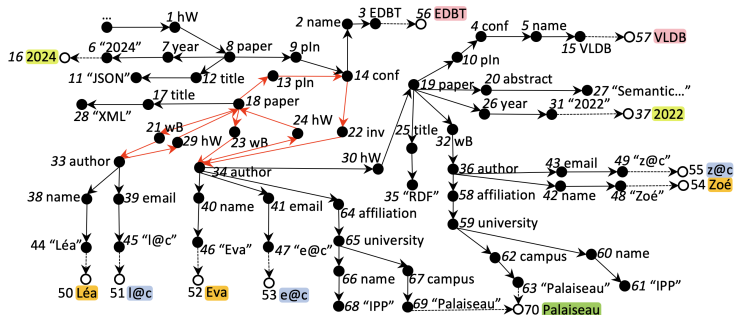
Ioana Manolescu
Inria, IPP

et al...
INSEC, ...



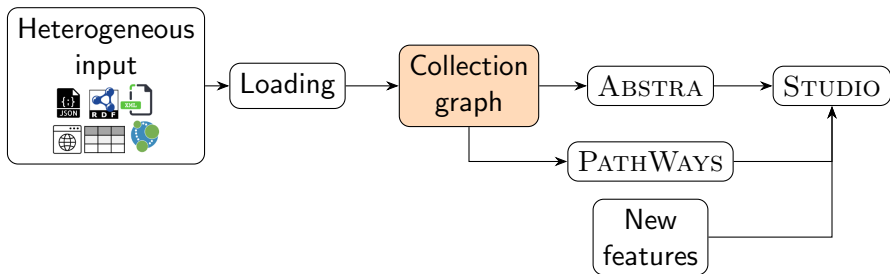
Graph construction

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



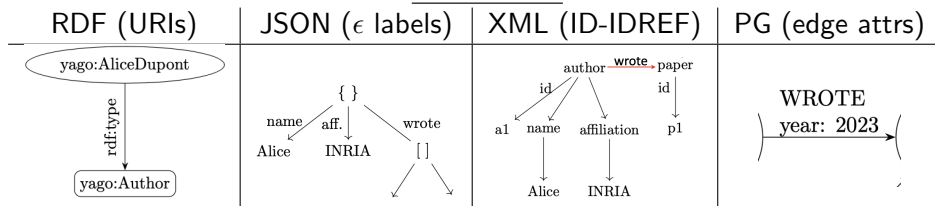
Create a compact representation of the data graph

“Efficient algorithms and applications”



A uniform view of data formats

Each data format has its own specificities:



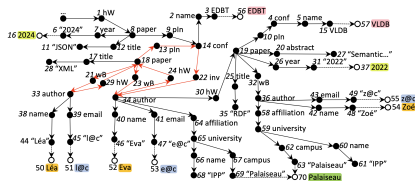
But, we **encode** the same logic:

- **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]



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

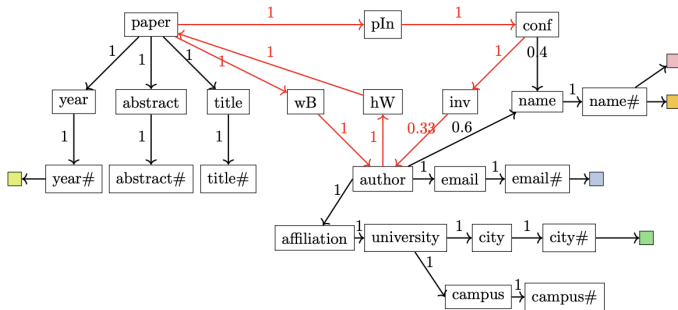
The collection graph

One **collection node** for each equivalence class

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

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

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



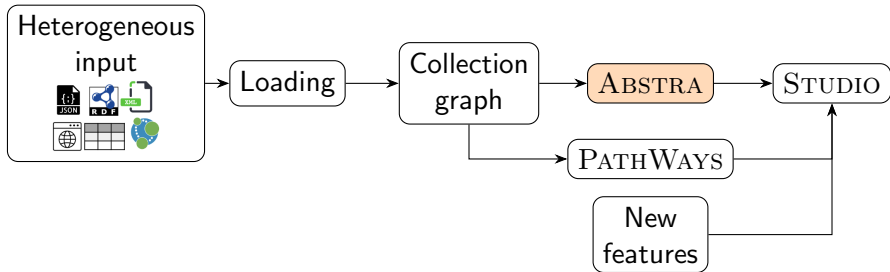
Build an Entity-Relationship model

“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**

⇒ 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**

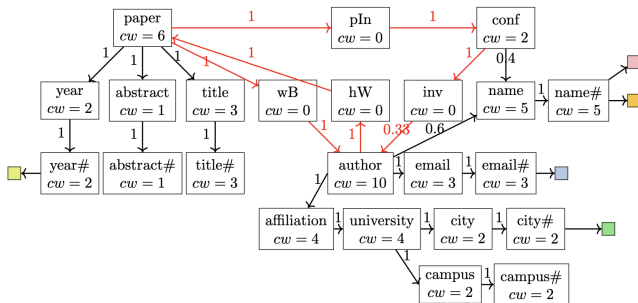
⇒ Retrieve / build **the Entity-Relationship model** behind any dataset

	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

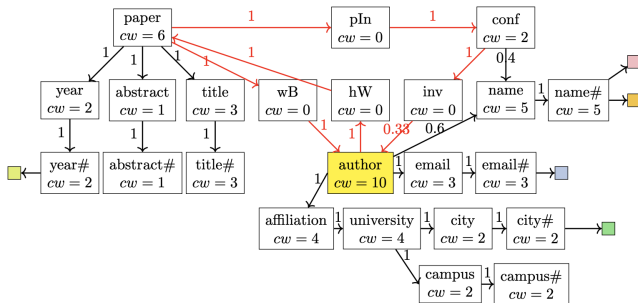
- 1 Assign a **weight** to each collection
- 2 While less than E_{max} main collections or data coverage $< cov_{min}$
 - 1 Pick C^* , the next heaviest collection
 - 2 Compute the **boundary** of C^*
 - 3 **Update** the collection graph to reflect the selection of C^*
 - 4 Recompute the weights



Collections weights, boundaries and graph updates

Collection weight

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



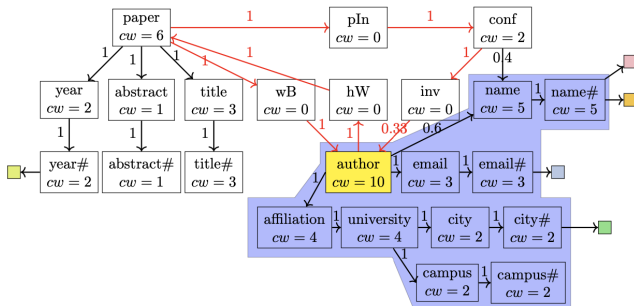
Collections weights, boundaries and graph updates

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

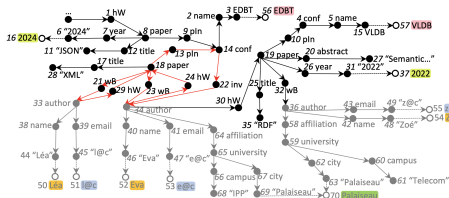
- W_{desc_k}
- W_{leaf_k}
- W_{DAG}
- $W_{PageRank}$
- $W_{dwPageRank}$

Boundary

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

Graph update

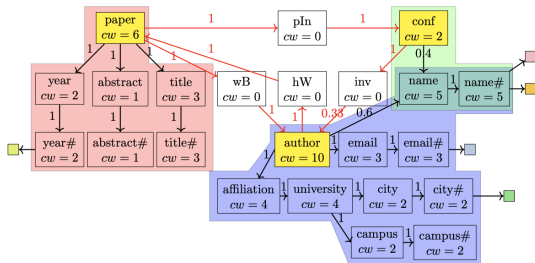
- $update_{boolean}$
- $update_{exact}$



Find relationships between main collections

Possible relationships

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



- paper \rightarrow wB \rightarrow author
- paper \rightarrow pIn \rightarrow conf
- author \rightarrow hW \rightarrow paper
- conf \rightarrow inv \rightarrow author

The final output in ABSTRA

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

The screenshot displays the ABSTRA web application interface. At the top, there's a navigation bar with 'Abstra', 'Home', 'About', and 'Help'. The main content area is titled 'Here's what your dataset xmark1 contains!' and is divided into two sections: 'Description' and 'Entity/Relationship schema'.

Description:

Entities:

- A collection of 59486 bidder having the following properties: ⚙️
 - date (100%)
 - increase (100%)
 - time (100%)
- A collection of 25500 person having the following properties: ⚙️
 - name (100%)
 - emailaddress (100%)
 - phone (50%)
 - homepage (50%)
 - creditcard (50%)
 - profile (50%) ⚙️
 - interest (294%)
 - education (51%)
 - business (100%)
 - profile@income (100%)
 - gender (50%)
 - age (50%)
 - personid (100%)
 - address (50%) ⚙️
 - street (100%)
 - city (100%)
 - country (100%)
 - province (49%)
 - zipcode (100%)
- A collection of 21750 item having the following properties: ⚙️
 - incategory (378%)
 - payment (100%)
 - itemid (100%)
 - shipping (100%)
 - quantity (100%)

Entity/Relationship schema:

Abstraction of file:///data/abstra/abstraction_data/xmark1 ont (1392794 normalized nodes, 136 collections, 5 main collections, data coverage is 70%) with parameters PROP_P9, BOUND_SFLOOD, UPDATE_EXACT, ENABLE_SCORE

The diagram shows the following entities and their relationships:

- person (person) (25500)**: Properties: name, emailaddress, phone, homepage, creditcard, profile, personid, address.
- open_auction (open_auction) (12000)**: Properties: quantity, initial, current, reserve, type, interval, open_auctionid, privacy.
- closed_auction (closed_auction) (9750)**: Properties: price, date, quantity, type.
- bidder (bidder) (59486)**: Properties: date, increase, time.
- item (item) (21750)**: Properties: incategory, payment, itemid, shipping, quantity, description, mailBox, item@featured, name, location.

Relationships:

- personref**: Connects person and bidder.
- workfax_search**: Connects person and open_auction.
- annotation_auther**: Connects person and open_auction.
- itemref**: Connects open_auction and item.
- itemref**: Connects closed_auction and item.

Enumerate entity paths

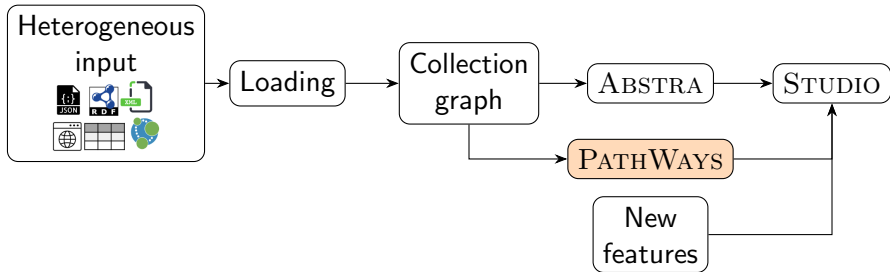
“Interesting **entity connections**”

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

⇒ 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**?

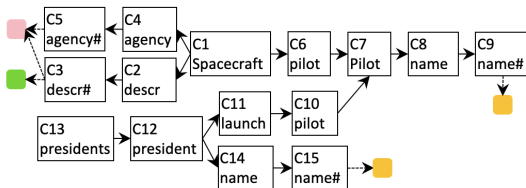
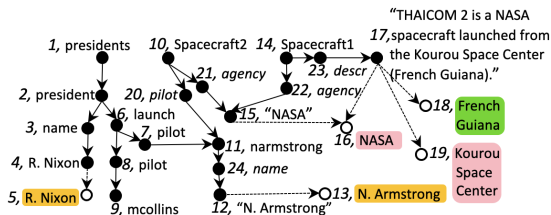
- 1 No query writing, nor prior knowledge
- 2 A **tabular, high-level output**, easy to grasp for **NTUs**
- 3 Do it **efficiently** even if the data graph is large

⇒ 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



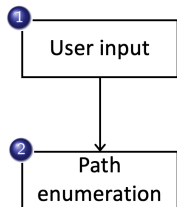
Collection (entity) path enumeration

1

User input

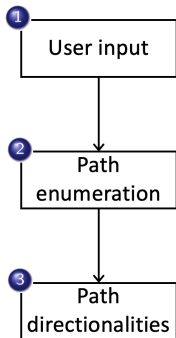
- (τ_1, τ_2) ; max path length; non-specific connections

Collection (entity) path enumeration



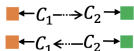
- (τ_1, τ_2) ; max path length; non-specific connections
- **Enumerate** all collection paths using the user input
- **Regardless of edge direction**

Collection (entity) path enumeration

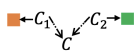


- (τ_1, τ_2) ; max path length; non-specific connections
- **Enumerate** all collection paths using the user input
- **Regardless of edge direction**

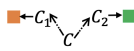
unidirectional



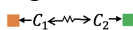
shared-sink



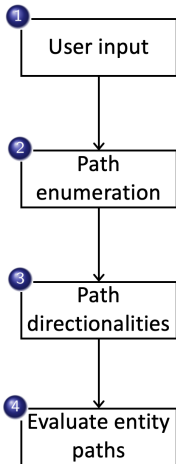
shared-root



general

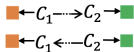


Collection (entity) path enumeration

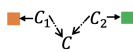


- (τ_1, τ_2) ; max path length; non-specific connections
- **Enumerate** all collection paths using the user input
- **Regardless of edge direction**

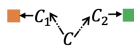
unidirectional



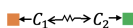
shared-sink



shared-root



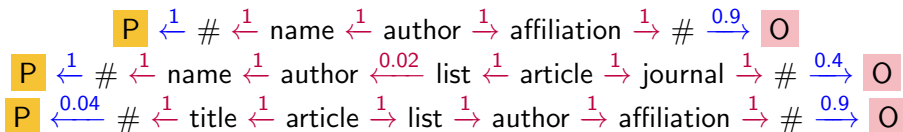
general



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

Collection paths interestingness

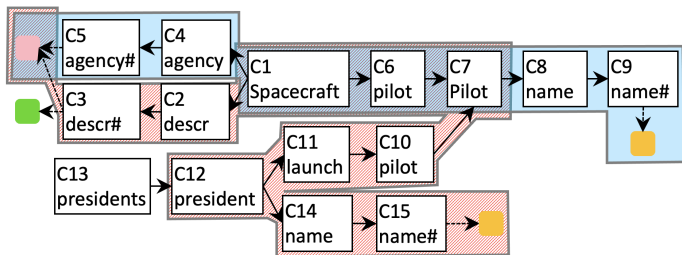
- Many enumerated paths are non-interesting
- Humans can “see/feel it”
- How to quantify the interestingness of a path?
 - 1 Compute the **reliability** r of its extracted NEs
 - How many NEs of a given type are extracted from a leaf collection?
 - 2 Compute the **force** f of each structural collection edge
 - 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 $r \geq \theta$



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 views
 - Saves computation time
- Collection paths are rewritten using views
 - Reduces the number of joins



Optimized data paths computation (2/2)

Greeditly 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}

■ \leftarrow agency# \leftarrow agency \leftarrow **Spacecraft** \rightarrow **pilot** \rightarrow **Pilot** \rightarrow name \rightarrow name# \rightarrow ■

■ \leftarrow agency# \leftarrow agency \leftarrow **v** \rightarrow name \rightarrow name# \rightarrow ■

```
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/>

The screenshot shows the Pathways web application interface. At the top, there is a navigation bar with 'Pathways', 'Home', 'About', and 'Help'. Below this, there are two main sections: 'Load a Pathways result from database' and 'Run Pathways on a dataset'. The 'Load a Pathways result from database' section has a search input field containing 'pathways_pubmedcoi' and two dropdown menus for entity types. The 'Run Pathways on a dataset' section has a form for entering a database name and two dropdown menus for entity types, with a 'Run Pathways' button below. Below these sections is a 'Result' section with a table of data paths. The table has columns for ID, Name#val, Name, Author, AuthorList, PubmedArticle, CoiStatement, and CoiStatement#val. The first four rows of the table are visible, showing authors like Giampiero Mazzaglia and Paolo Angelo Cortes, and associated entities like Bayer and Pfizer. Below the table, there are several links to other data path results, such as 'CoiStatement#val - CoiStatement - PubmedArticle - AuthorList - Author - Affiliation - Affiliation#val (480 data paths)'. At the bottom, there is a footer with contact information for Nelly Barret and other team members.

Pathways Home About Help

Load a Pathways result from database

pathways_pubmedcoi

- (PERSON, ORGANIZATION), max 100 paths of max size 20 [Load](#)
- (PERSON, LOCATION), max 100 paths of max size 20 [Load](#)

Run Pathways on a dataset

Enter a database name:

Left entity type:

Right entity type:

[Run Pathways](#)

Result

Sort queries by length Sort queries by number of associated data paths Hide/show queries without associated data paths

▶ Name#val - Name - Author - AuthorList - PubmedArticle - CoiStatement - CoiStatement#val (860 data paths)

ID	Name#val	Name	Author	AuthorList	PubmedArticle	CoiStatement	CoiStatement#val
2901	Giampiero Mazzaglia	Name	Author	AuthorList	PubmedArticle	CoiStatement	... Bayer ...
2931	Giampiero Mazzaglia	Name	Author	AuthorList	PubmedArticle	CoiStatement	... Pfizer ...
5531	Paolo Angelo Cortes	Name	Author	AuthorList	PubmedArticle	CoiStatement	... Bayer ...
5561	Paolo Angelo Cortes	Name	Author	AuthorList	PubmedArticle	CoiStatement	... Pfizer ...

▶ CoiStatement#val - CoiStatement - PubmedArticle - AuthorList - Author - Affiliation - Affiliation#val (480 data paths)

▶ Name#val - Name - Author - AuthorList - PubmedArticle - ArticleTitle - ArticleTitle#val (8 data paths)

▶ Name#val - Name - Author - Affiliation - Affiliation#val (71 data paths)

▶ CoiStatement#val - CoiStatement - PubmedArticle - ArticleTitle - ArticleTitle#val (12 data paths)

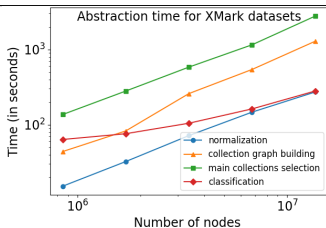
Authors: Nelly Barret @ Inria, Antoine Gauquier @ IMT Nord Europe, Jean Law @ Ecole Polytechnique, Ioana Manolescu @ Inria

Main contact: nelly.barret@inria.fr

Quick overview of experiments

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

ABSTRA



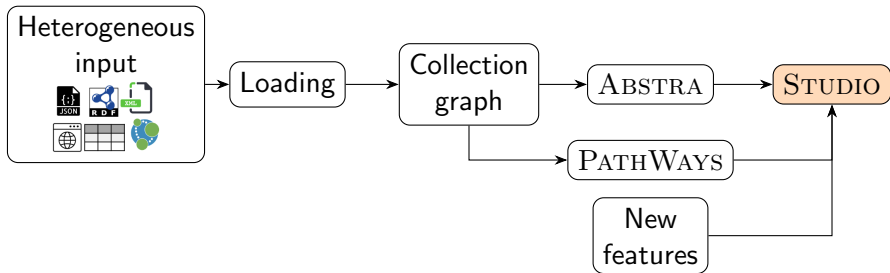
- User study
- Comparison to schemas

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	61x
(τ_P, τ_L)	37.29	19.38	2x
(τ_L, τ_O)	151.29	20.47	7x
(τ_P, τ_P)	152.59	44.27	3x
(τ_L, τ_L)	169.64	71.63	2x
(τ_O, τ_O)	317.92	23.24	13x

- # 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)

Connection Studio Projects



Sort by

<p>Project Cac ✕</p> <p>1 files Created on: 2023-07-13 11:32:13 Latest file addition: 2023-07-13 11:32:13</p> <p><input type="button" value="MANAGE"/></p>	<p>Project Cac40 ✕</p> <p>1 files Created on: 2023-07-05 16:12:38 Latest file addition: 2023-07-05 16:12:38</p> <p><input type="button" value="MANAGE"/></p>	<p>Project Hatvp Cac ✕</p> <p>2 files Created on: 2023-07-11 16:03:48 Latest file addition: 2023-07-11 16:39:39</p> <p><input type="button" value="MANAGE"/></p>	<p>Project Hatvp Cac40 ✕</p> <p>2 files Created on: 2023-07-05 15:46:07 Latest file addition: 2023-07-05 16:25:52</p> <p><input type="button" value="MANAGE"/></p>
<p>Project Hatvpsmall ✕</p> <p>No files uploaded yet, add one!</p> <p><input type="button" value="MANAGE"/></p>	<p>Project Pubmed ✕</p> <p>1 files Created on: 2023-07-05 09:46:07 Latest file addition: 2023-07-05 09:46:07</p> <p><input type="button" value="MANAGE"/></p>	<p>Project Recac40 ✕</p> <p>1 files Created on: 2023-07-12 23:24:56 Latest file addition: 2023-07-12 23:24:56</p> <p><input type="button" value="MANAGE"/></p>	

Future work: STUDIO

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

Explore

Connection Studio

Uploaded files Project: Hatvp Cac

Uploaded files

+ ADD ?

⚙ DISPLAY ADVANCED OPTIONS

ID	File	Path	Type	Creation date	
1	hatvp-cleaned.xml	file:/Users/nelly/Documents/boulot/theseNelly/connection-lens/./connection-studio/demo-CFI	XML	2023-07-11 16:03:48+02	
2	Cac40.csv	file:/Users/nelly/Documents/boulot/theseNelly/connection-lens/./connection-studio/demo-CFI	CSV	2023-07-11 16:39:39+02	

Future work: STUDIO

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

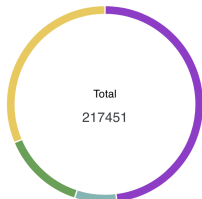
Explore

Connection Studio Statistics

Project: Hatvp Cac

Entities distribution by type

< Identified entities >



● Number of dates ● Number of Persons ● Number of Places
● Number of Organizations ● Number of hashtags

Entity cloud



Future work: STUDIO

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

The screenshot displays the STUDIO query builder interface. It features five rows of path definitions, each with a 'Starting variable' and an 'Ending variable'. The paths are:

- Path 1: `declaration.general.declarer.name#val` (Starting: `decla`, Ending: `deputyName`)
- Path 2: `declaration.financialInterest.items.item` (Starting: `decla`, Ending: `item`)
- Path 3: `item.company#val.extract:o` (Starting: `item`, Ending: `companyName`)
- Path 4: `item.nbShares#val` (Starting: `item`, Ending: `nbShares`)
- Path 5: `row.company_name.#val.extract:o` (Starting: `csvline`, Ending: `companyName`)

Buttons for 'EVALUATE THE QUERY' and 'SAVE CHANGES' are visible. Below the paths, there are 'Join' options with radio buttons for 'Required' and 'Optional', and trash icons.

At the bottom, a table displays the results of the query:

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

Explore **Connection Studio** Search Project: Recac40

Airbus Engie 🔍

⚙️ DISPLAY ADVANCED OPTIONS

Always display edge labels

RESULTS NEIGHBORS

N°2
13 Nodes found
From 1 source
Score: 0.25

N°3
7 Nodes found
From 1 source
Score: 0.25

N°4

- Data node
- Location
- Organization

Future work: STUDIO

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

Extraction model: Stanford Extractor

File language: English

Extraction policy:

```
declaration.general.declarant.nom#val Person,
declarations.declaration.origine#val NoExtract
```

Split long texts: False

[SAVE PARAMETERS](#)

2998	als thom
2998	alsthom
2998	alsthom atlantique
2998	alstom


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

