Empowering Investigative Journalism with Graph-based Heterogeneous Data Management

Angelos-Christos Anadiotis

Ecole Polytechnique and Institut Polytechnique de Paris

Conflicts of Interest database

"A conflict of interest is any situation where a public interest may interfere with a public or private interest, in such a way that the public interest may be, or appear to be, unduly influenced."

French transparency law, 2011

Biomedical domain

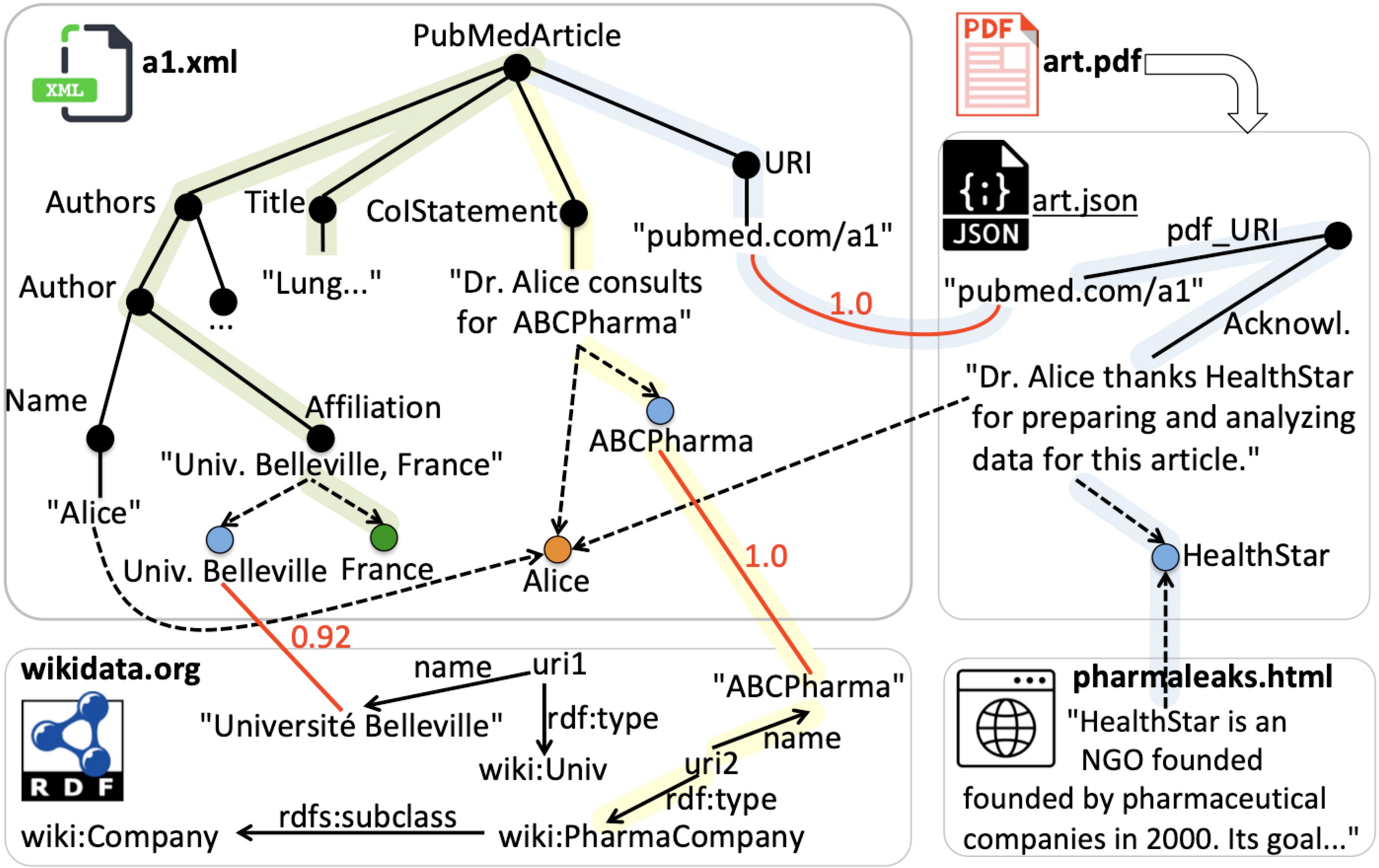
- **Experts in the biomedical area** advise national and international officials on decisions with impact on public health
- **Companies with interests in this area** may recruit experts likely to be auditioned by regulatory boards
- Goal: ***establish a database of CoIs*** where it would be easy to "*find the declared links of Dr. Alice with HealthStar*"

Biomedical domain

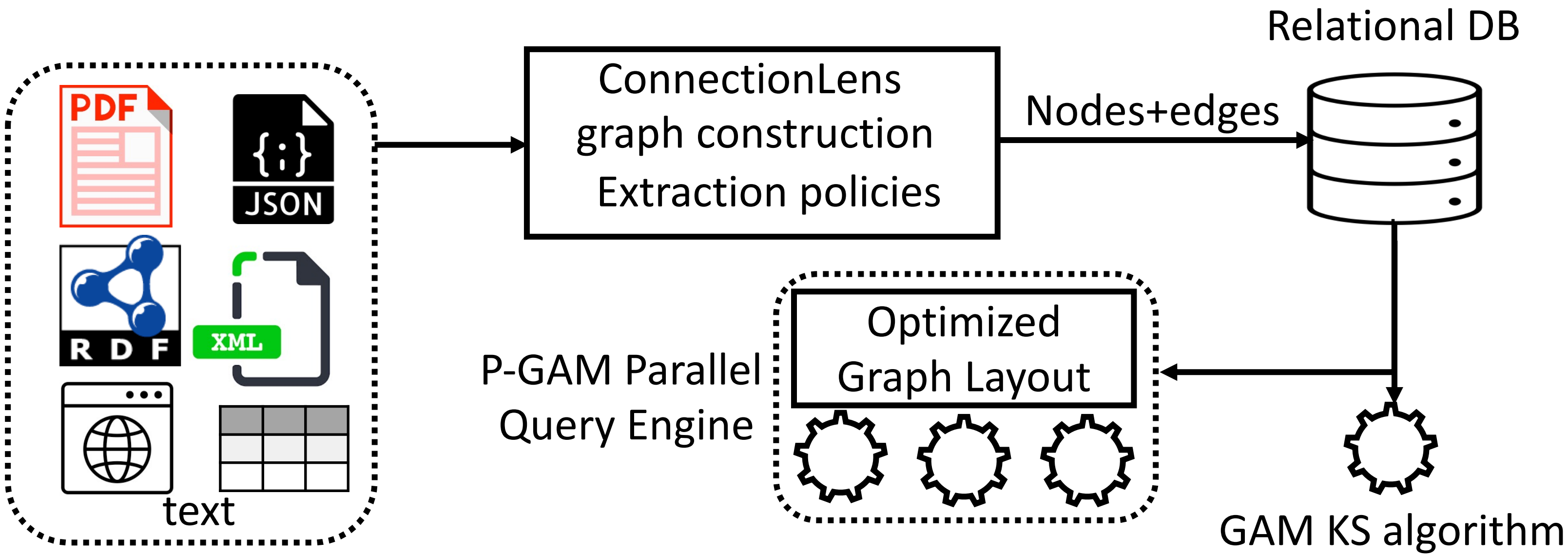
- **Experts in the biomedical area** advise national and international officials on decisions with impact on public health
- **Companies with interests in this area** may recruit experts likely to be auditioned by regulatory boards
- **Goal**: ***establish a database of CoIs*** where it would be easy to "*find the declared links of Dr. Alice with HealthStar*"

Usually available, but *technically buried* information

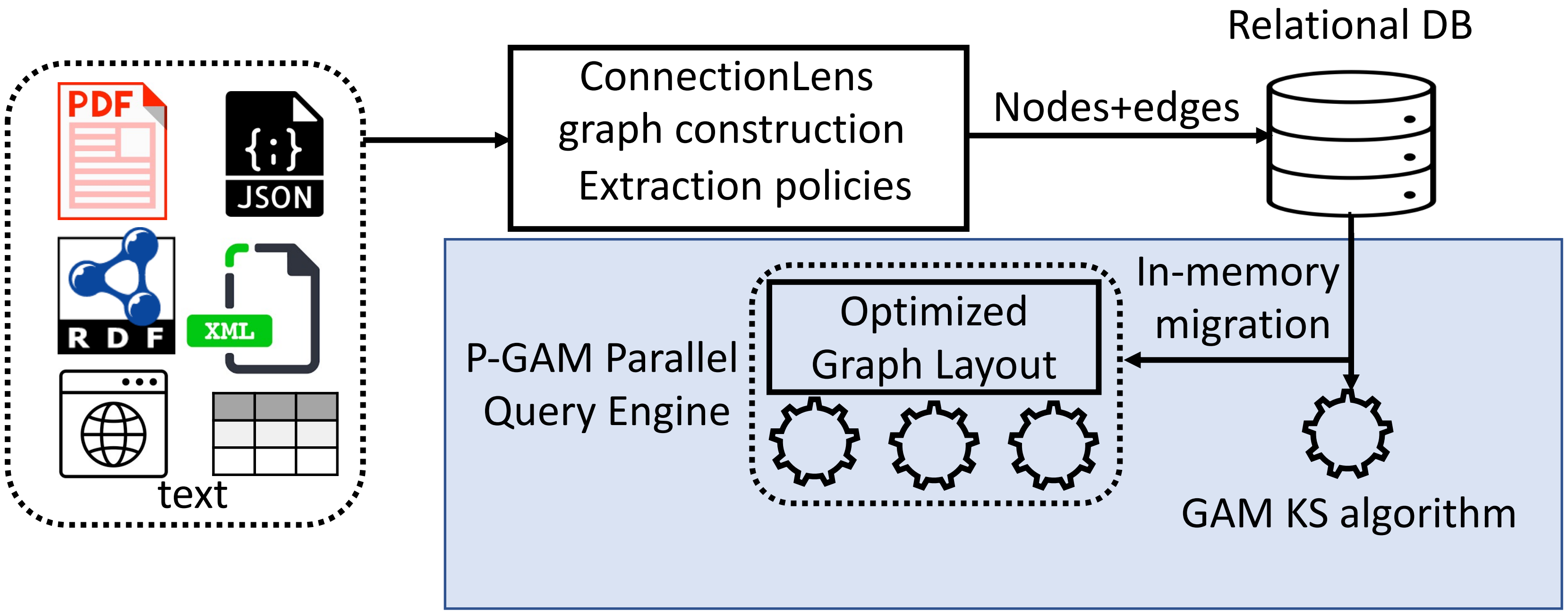
Landscape of heterogeneous data



ConnectionLens graph processing pipeline



ConnectionLens graph processing pipeline



Querying the graph

Problem statement

- Given the graph **G = (N, E)** built out of the datasets D and a query keywords **Q= {w_1, ...,w_m}**, return the k highest-score minimal answer trees
- An **answer tree** is a set of edges which (i) form a tree, and (ii) for each w_i, contain at least one node whose label matches w_i
- We are interested in **minimal answer trees**, that is:
  - Removing an edge from the tree should make it lack some query keywords w_i
  - If a query keyword w_i matches the label of more than one nodes in the answer tree, then all these matching nodes must be equivalent

Problem statement

- Given the graph **G = (N, E)** built out of the datasets D and a query keywords **Q= {w_1, ...,w_m}**, return the k highest-score minimal answer trees
- An **answer tree** is a set of edges which (i) form a tree, and (ii) for each w_i, contain at least one node whose label matches w_i
- We are interested in **minimal answer trees**, that is:
  - Removing an edge from the tree should make it lack some query keywords w_i
  - If a query keyword w_i matches the label of more than one nodes in the answer tree, then all these matching nodes must be equivalent

Related to GSTP + bidirectional edges

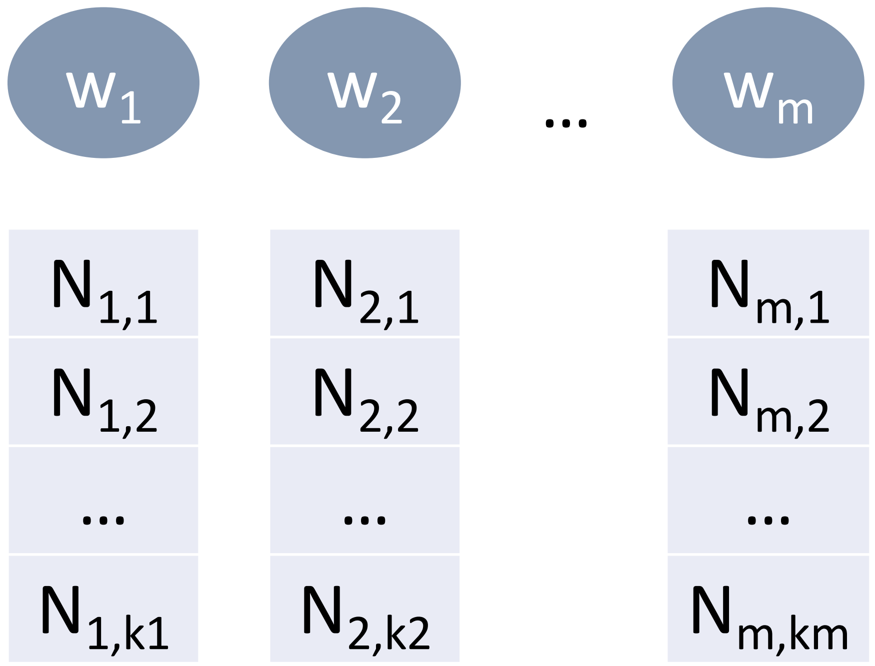
Return k highest-score trees among those found

Grow and Aggressive Merge

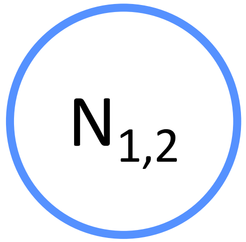
Grow

w_1	w_2	…	w_m
$N_{1,1}$	$N_{2,1}$		$N_{m,1}$
$N_{1,2}$	$N_{2,2}$		$N_{m,2}$
…	…		…
$N_{1,k1}$	$N_{2,k2}$		$N_{m,km}$

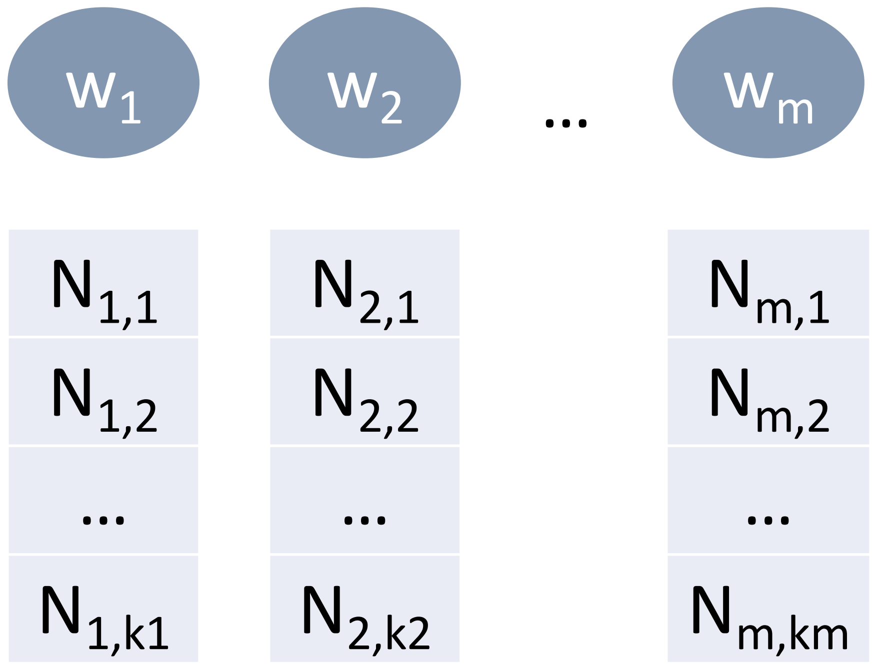
Grow and Aggressive Merge



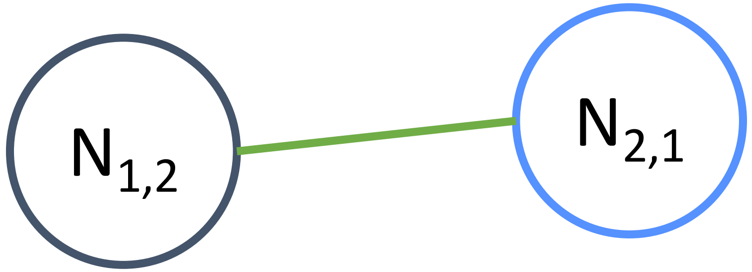
Grow



Grow and Aggressive Merge



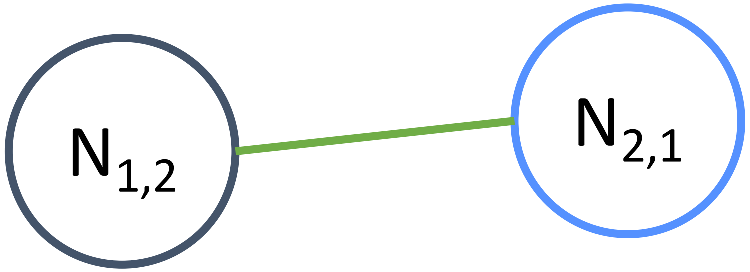
Grow



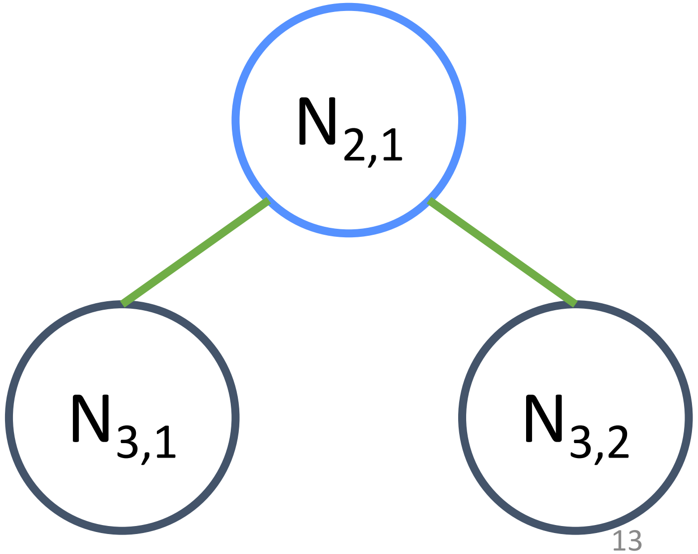
Grow and Aggressive Merge

w_1	w_2	…	w_m
$N_{1,1}$	$N_{2,1}$		$N_{m,1}$
$N_{1,2}$	$N_{2,2}$		$N_{m,2}$
…	…		…
$N_{1,k1}$	$N_{2,k2}$		$N_{m,km}$

Grow



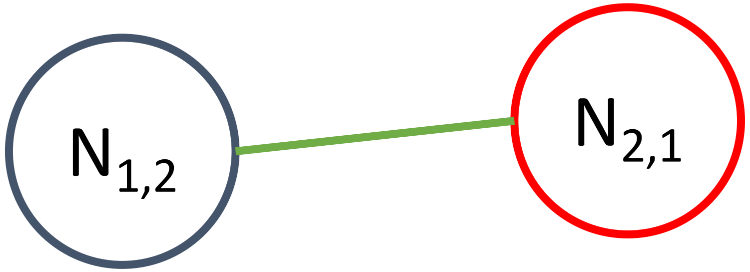
Merge



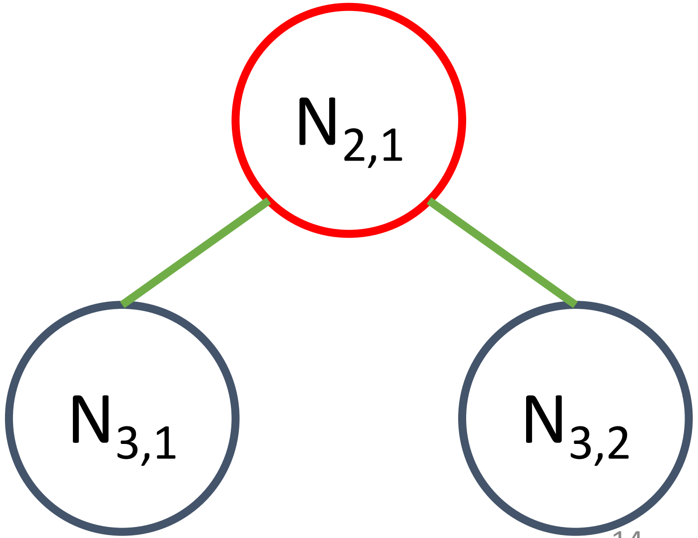
Grow and Aggressive Merge

w_1	w_2	…	w_m
$N_{1,1}$	$N_{2,1}$		$N_{m,1}$
$N_{1,2}$	$N_{2,2}$		$N_{m,2}$
…	…		…
$N_{1,k1}$	$N_{2,k2}$		$N_{m,km}$

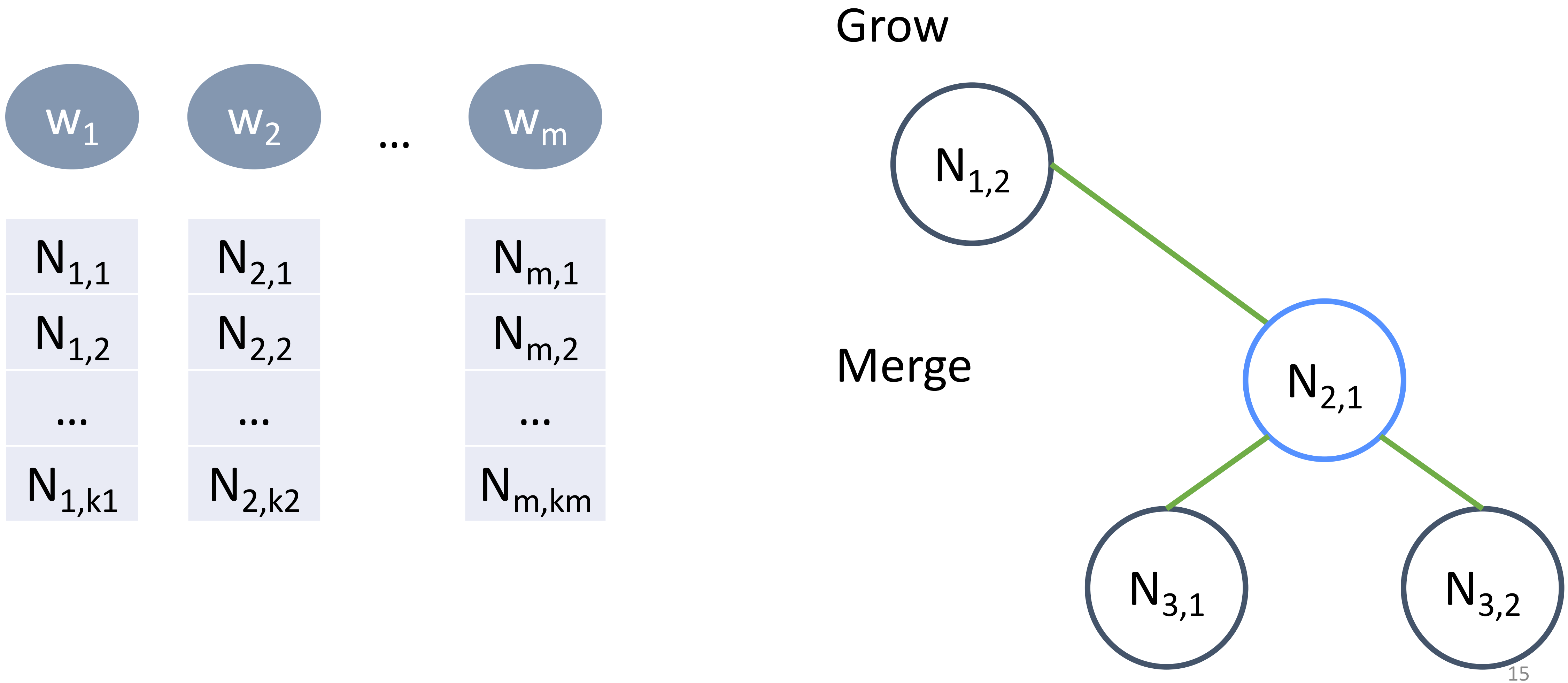
Grow



Merge



Grow and Aggressive Merge

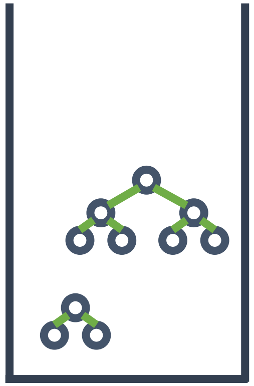


Which tree to Grow or to Merge?

- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size

Which tree to Grow or to Merge?

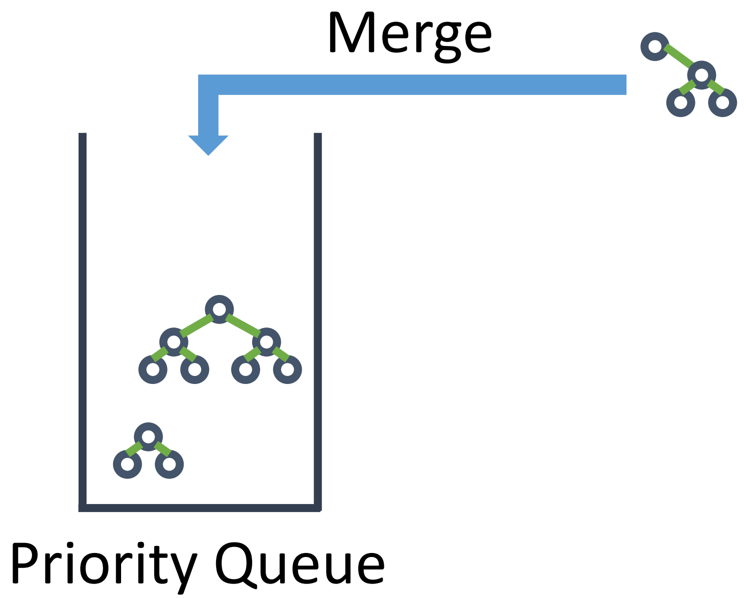
- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size



Priority Queue

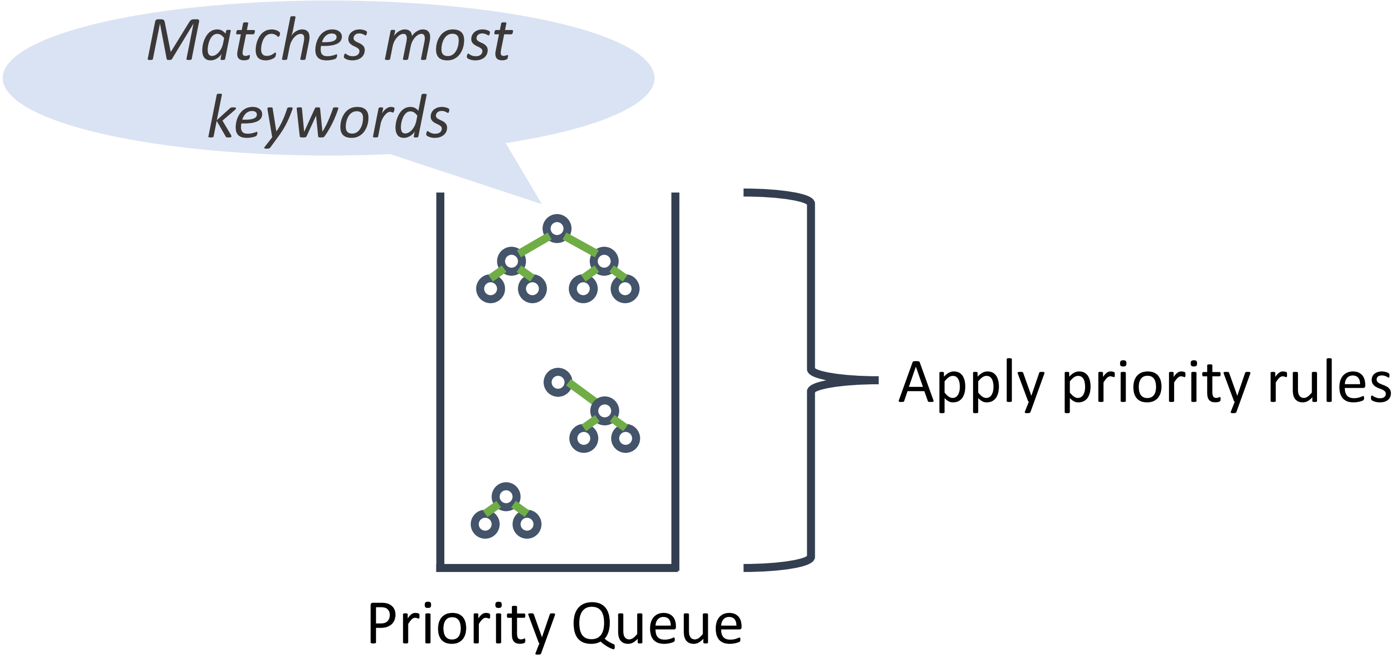
Which tree to Grow or to Merge?

- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size



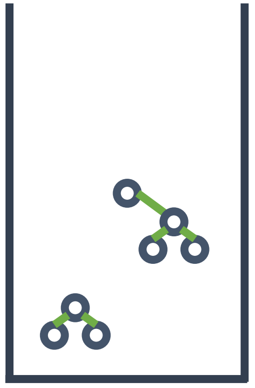
Which tree to Grow or to Merge?

- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size

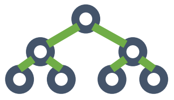


Which tree to Grow or to Merge?

- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size



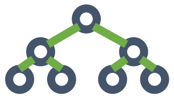
Priority Queue

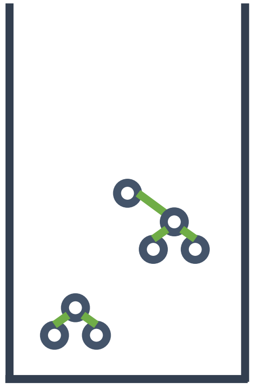


1. Grow answer tree
2. Merge with same-rooted answer trees

Which tree to Grow or to Merge?

- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size



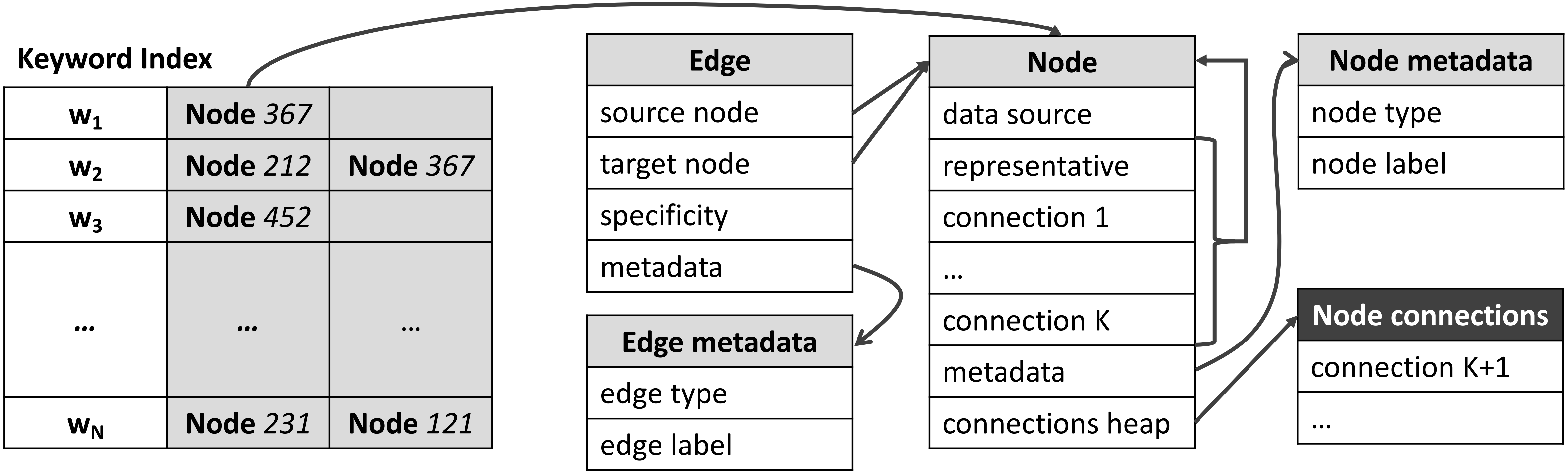


1. Grow answer tree
2. Merge with same-rooted answer trees

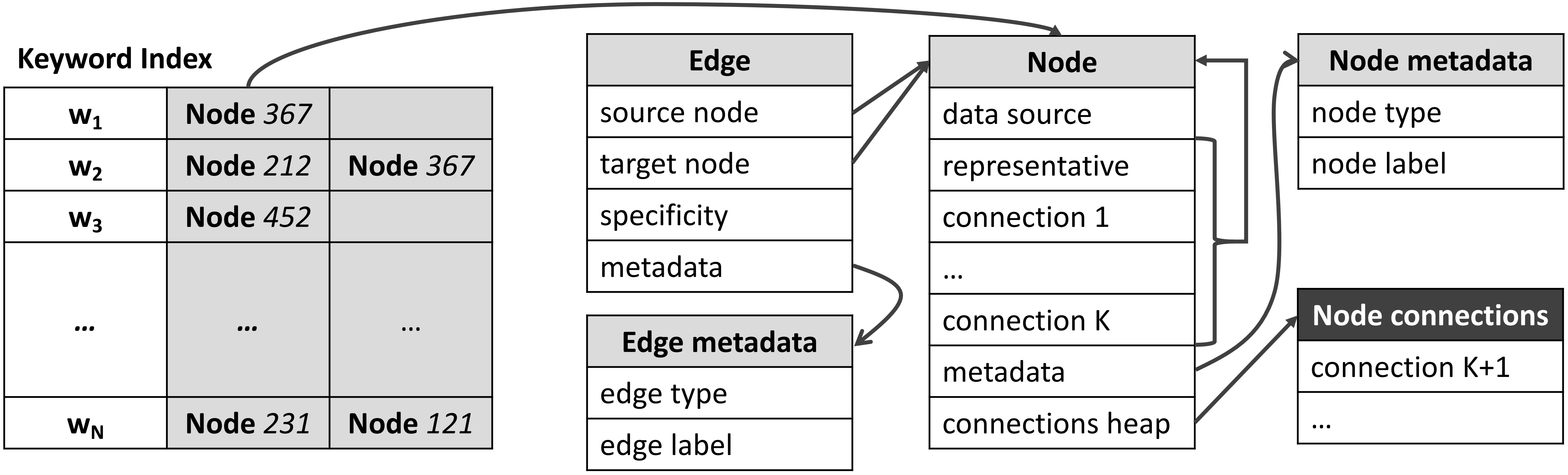
Priority Queue

Mixed BFS/DFS approach of graph search

In-memory graph layout



In-memory graph layout



Row-major, CPU-cache-friendly data layout

Duplicate work elimination

- The same answer tree may be created following different combinations of Grow and Merge
  - ➢Duplicate work
- Maintain a history of explored trees
- Every answer tree is inserted only once:
  - in the history of explored trees
  - in the priority queue

Parallel search

- Cannot partition the graph:
  - expensive, and we do not know which parts we will need
  - no assumption on the shape of the graph
- DFS/BFS alternation incurs mixed scalability requirements
- P-GAM bottlenecks
  - size of intermediate results

Parallel search

- Cannot partition the graph:
  - expensive, and we do not know which parts we will need
  - no assumption on the shape of the graph
- DFS/BFS alternation incurs mixed scalability requirements
- P-GAM bottlenecks
  - size of intermediate results

Shared-everything

Concurrent data structures

Experimental evaluation – CoI application

- 450,000 PubMed bibliographic notices (2019, 2020)
- 42,000 PDF articles transformed to JSON
- 781 HTML pages describing relationships between people and organizations
- Load the graph in the main memory
- Query thresholds:
  - 1000 solutions
  - 1 minute of execution time

CoI application results (anonymized)

#	Keywords	T^1	T^{last}	T	S	#DS
1	A1, A2	200	4840	4840	1000	1-6, **<u>5</u>**
2	A3, I1	1263	20547	60000	13	2-4, **<u>2</u>**, **<u>3</u>**
3	A5, A6, I3	2602	4203	60000	15	6, 8, **<u>8</u>**
4	A8, I2, I4	667	51186	60000	63	4-7, **<u>6</u>**
5	A9, H3, I2	264	59831	60000	516	3-8, **<u>5</u>**
6	H2, I1, P1	1267	60212	60000	148	6-8, **<u>6</u>**
7	A5, A10, I2	19077	23160	60000	9	8, **<u>8</u>**
8	A9, I1, I4, I5	6327	55762	60000	38	8-9, 11, **<u>8</u>**
9	A7, I1, I6, P1	1857	3057	60000	8	7, 8, **<u>7</u>**, **<u>8</u>**
10	A7, A8, I1, I2, I4	3389	28237	60000	4	7-8, 11, **<u>11</u>**

Conclusion

- ConnectionLens introduces an end-to-end pipeline for constructing and querying graphs from heterogeneous data
- In-memory storage engine stores the graph data required for querying
- P-GAM queries the graph in parallel

Find out more about our work

- A. -C. Anadiotis, O. Balalau, C. Conceição, H. Galhardas, M. Y. Haddad, I. Manolescu, T. Merabti, J. You. Graph integration of structured, semistructured and unstructured data for data journalism. Information Systems (accepted for publication).
- A. -C. Anadiotis, O. Balalau, T. Bouganim, F. Chimienti, H. Galhardas, M. Y. Haddad, S. Horel, I. Manolescu, Y. Youssef. Empowering Investigative Journalism with Graph-based Heterogeneous Data Management. IEEE Data Engineering Bulletin (accepted for publication).
- A. -C. Anadiotis, O. Balalau, T. Bouganim, F. Chimienti, H. Galhardas, M. Y. Haddad, S. Horel, I. Manolescu, Y. Youssef. Discovering Conflicts of Interest across Heterogeneous Data Sources with ConnectionLens. Demonstration in CIKM 2021.

anr agence nationale de la recherche AU SERVICE DE LA SCIENCE

MINISTÈRE DES ARMÉES Liberté Égalité Fraternité

SourcesSay project

https://sourcessay.inria.fr