Identifying, Relating and Querying Large Heterogeneous RDF Sources

Andre Valdestilhas[1]

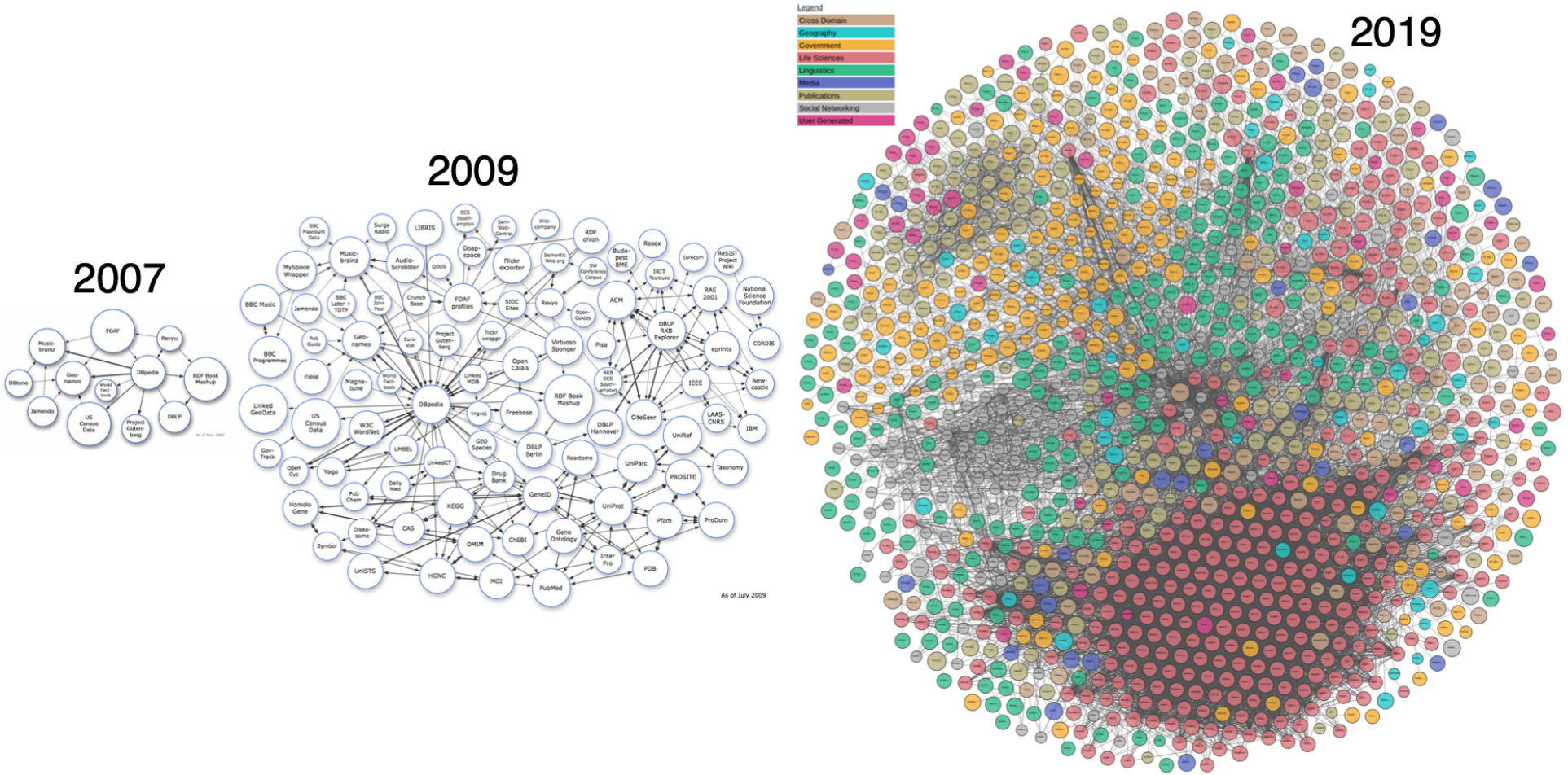
[1]AKSW Group, University of Leipzig, Germany

June 9, 2020

Outline

- Motivation
- Identifying
- Relating
- Querying
- Publications and statistics

Motivation: Large Heterogeneous RDF sources



- 221.7 billion triples (>5 terabytes)

LODSTATS
9,960
raw RDF datasets

SPARQL
Endpoint
559

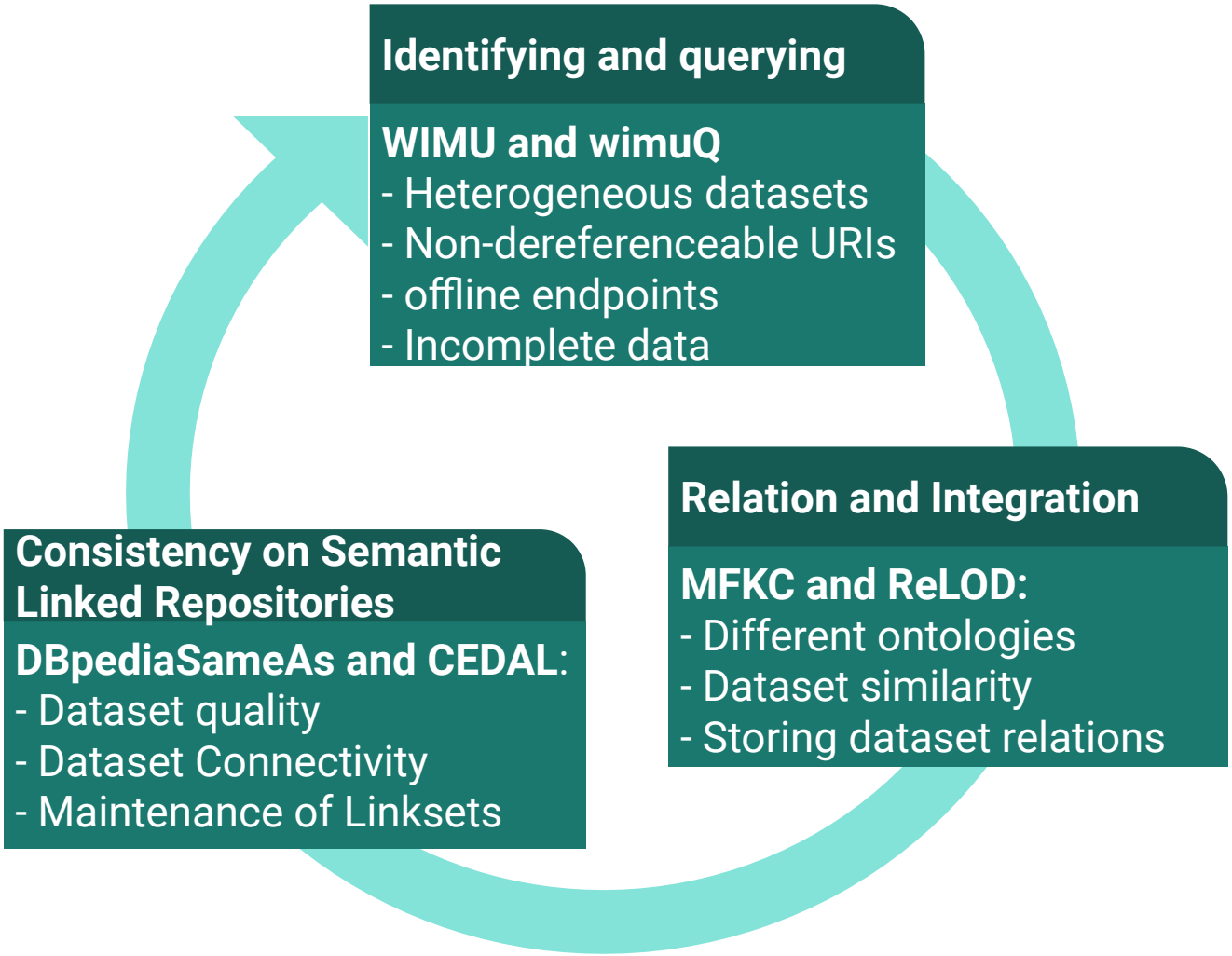
LODLaundromat
658,206
Datasets (HDT files)

How to Identify, Relate and Query?

Motivation: Linked data Lyfecicle

Auer, ISWC, 2012

Motivation: Methodology and contributions

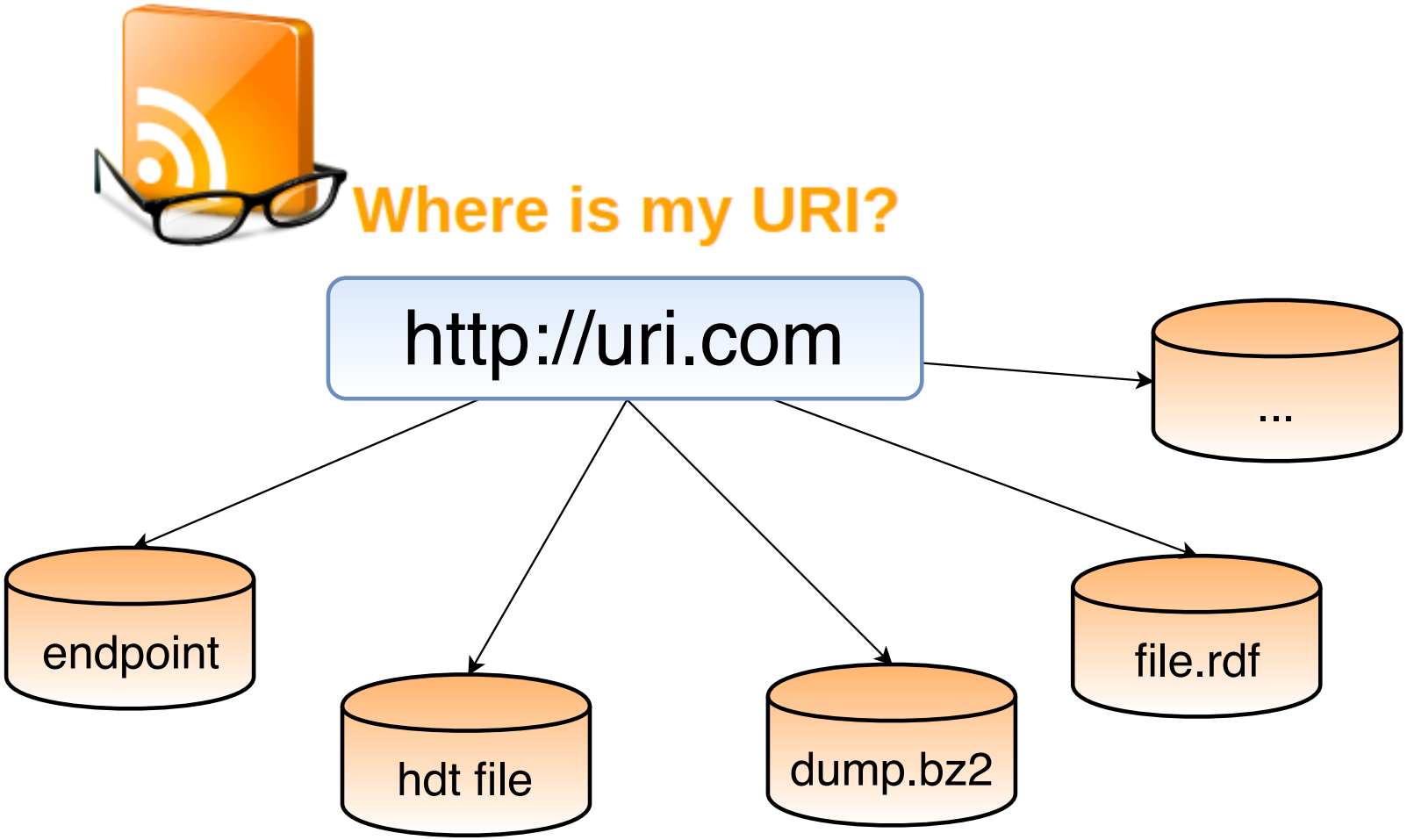


Identifying

Where is my URI? (WIMU)

Research Question

- Is there a way to know in which dataset a given URI was defined?



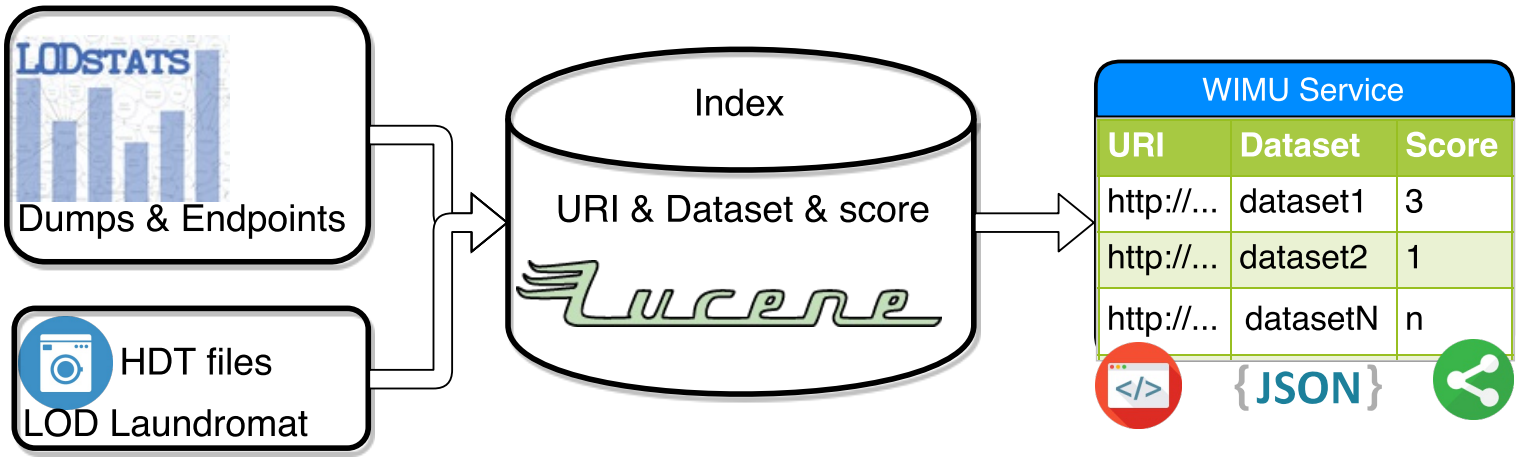
Where is my URI? (WIMU)

Approach

Goal

Index URIs and their use to enable Linked Data consumers to find relevant RDF data sources

- Rank the datasets proportionally to the number of literals
- Keep the provenance of the URI



Where is my URI? (WIMU)

Results - Valdestilhas, ESWC, 2018

	LOD Laundromat	LODStats	Total
URIs indexed	4,185,133,445	31,121,342	4,216,254,787
Datasets checked	658,206	9,960	668,166
Triples processed	19,891,702,202	38,606,408,854	58,498,111,056

- 69.8% datasets from LODStats present parser errors
- LODLaundromat presents 2.3% of parsing errors and 99% are indexed by WIMU

Relating

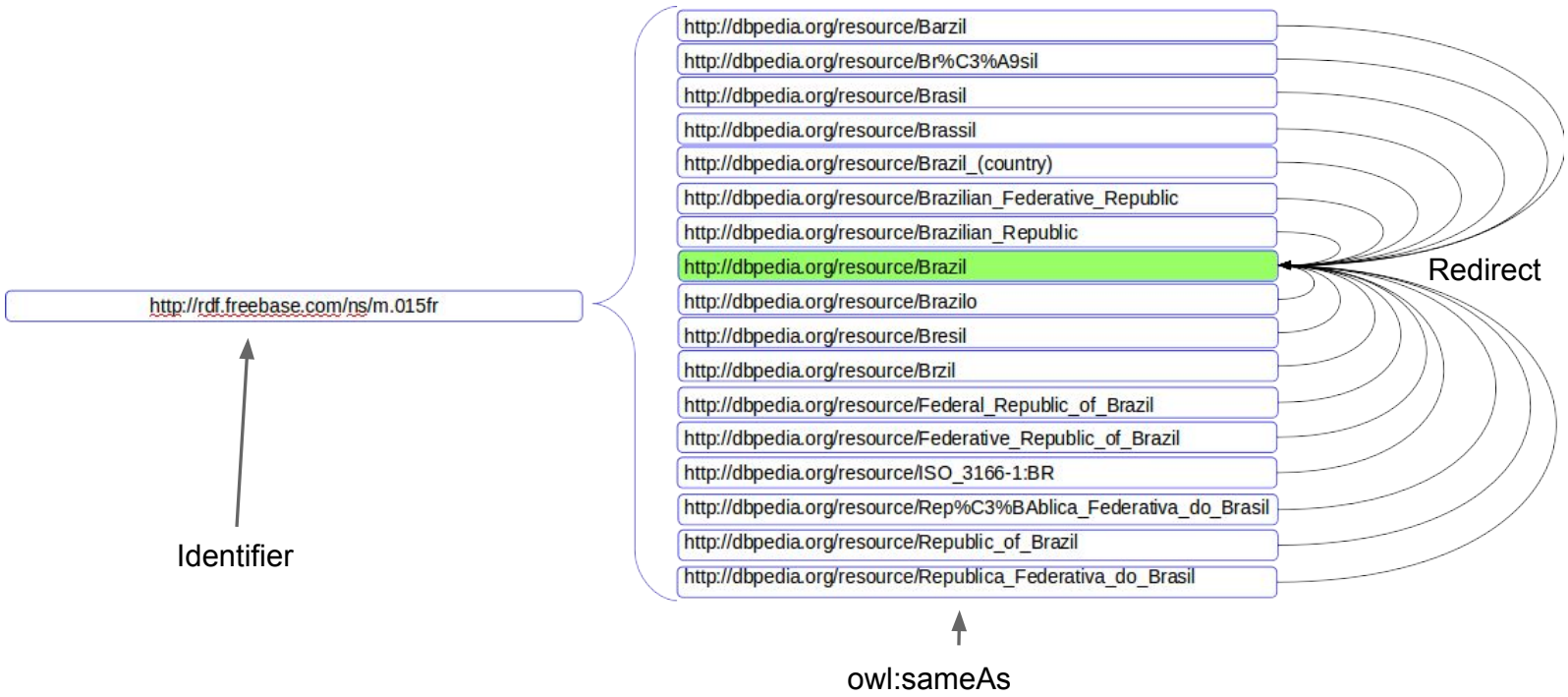
Consistency and Similarity

Consistency

DBpediaSameAs

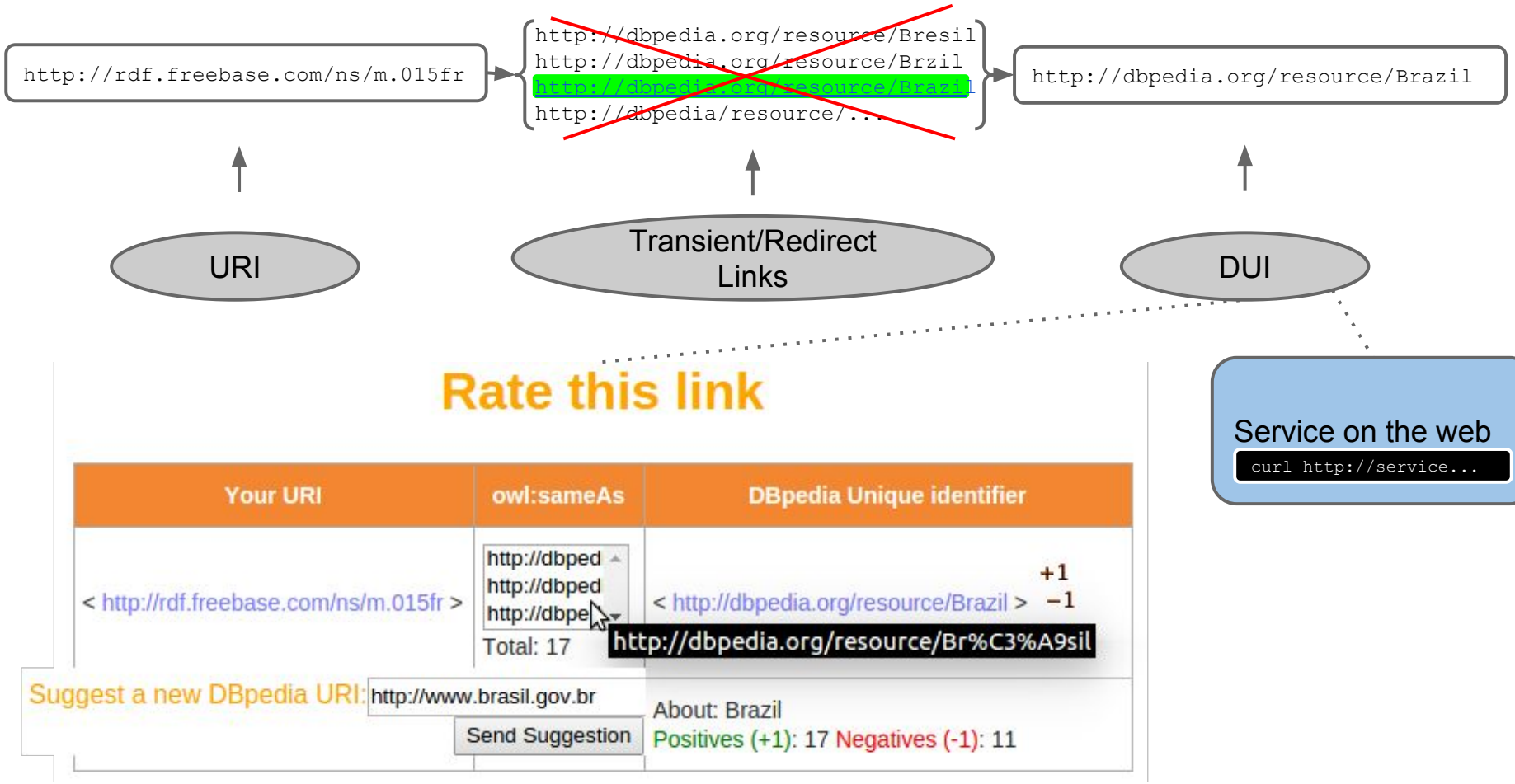
Research Question

- How to tackle heterogeneity in DBpedia identifiers?



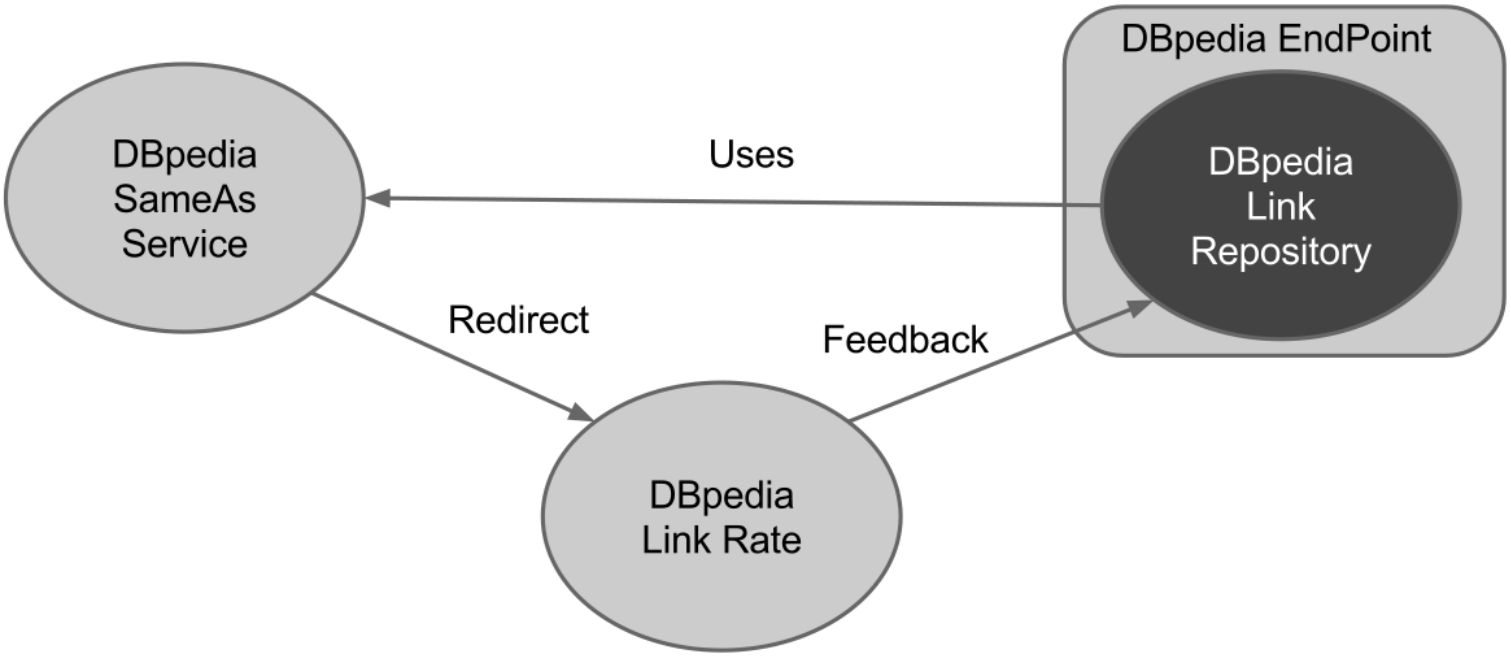
DBpediaSameAs

Approach



DBpediaSameAs

Contribution - Valdestilhas, Semantics, 2016



Open Question

- How to tackle the inconsistency in Linksets?

Consistency Error Detection Algorithm (CEDAL)

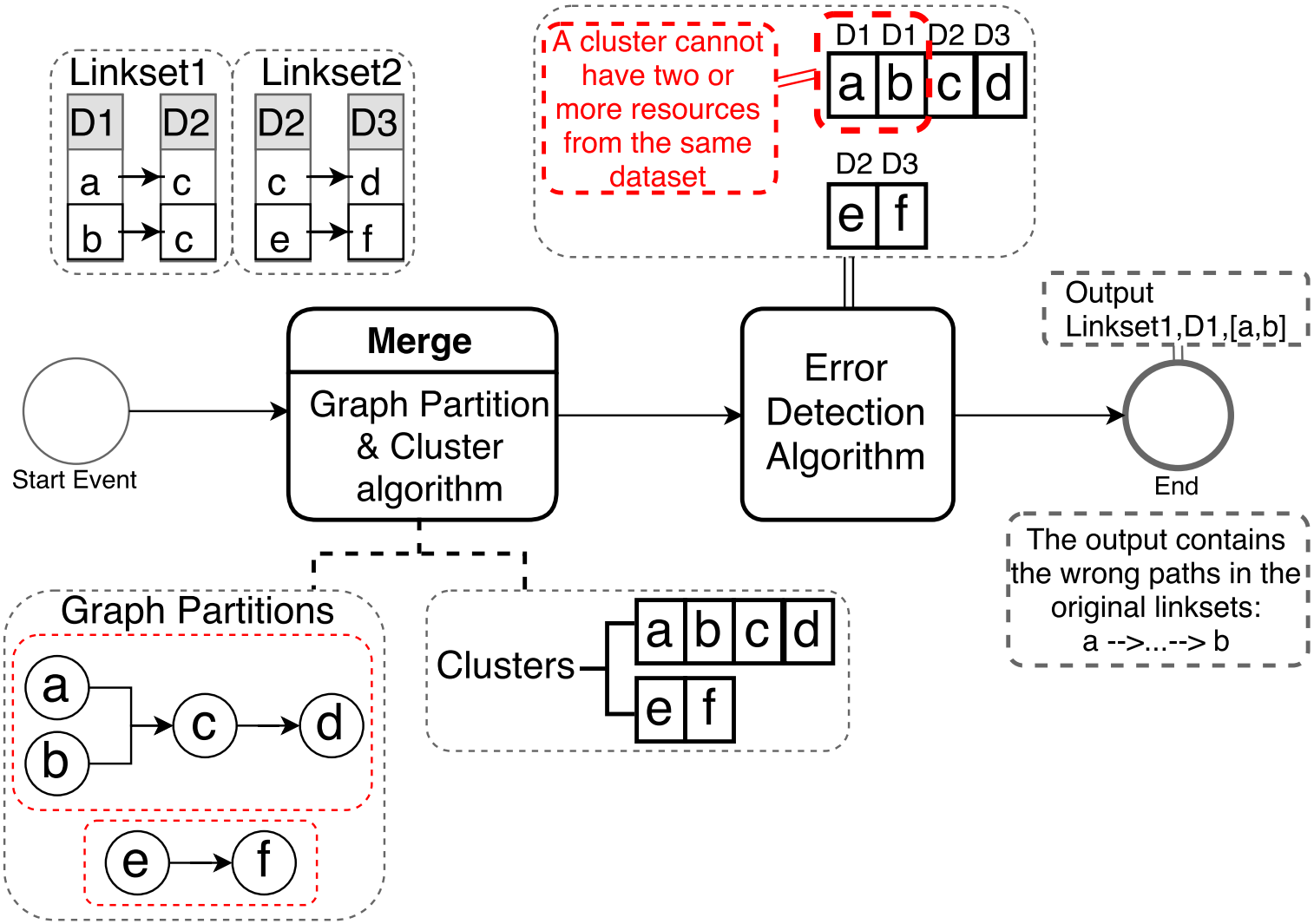
Research Question

- Is there a time-efficient algorithm to detect erroneous links in large-scale link repositories?
- Is there an approach to discover whether a linkset is consistent without computing all closures required by the property axiom?

Consistency Error Detection Algorithm (CEDAL)

Approach

- Semantics of URI + Efficient Graph Partition (Union-find)



Consistency Error Detection Algorithm (CEDAL)

Results - Valdestilhas, WI-IAT, 2017

Comparison of results with respect to the provenance of the links from LinkLion(http://www.linklion.org/)

Framework	**Errors**	**Resources**	**Errors (%)**	**M1**
sameas.org	3,792,326	28,130,994	13.5	0.865
LIMES	1,130	27,819	4.1	0.951
Silk	5,933	208,300	2.8	0.972
DBpedia Extraction Framework	12,615	914,180	1.4	0.986
All frameworks	**3,812,004**	**29,281,293**	**13.0**	**0.870**

Similarity

Most Frequent K Characters (MFKC)

Research Question

Instance Matching

- Given D_s, D_t and relation R
- Find $M = \{(s,t) \in D_s \times D_t : R(s,t)\}$

Research questions

- How to find a better way to compute M directly?
  - Compute approximation of M as follows
  - $M' = \{(s,t) \in D_s \times D_t : \sigma(s,t) \geq \theta\}$
- How to improve the naïve computation of M' $\in O(n^2)$
  - Find effective ways to compute $\sigma(s,t) \geq \theta$ quickly

Most Frequent K Characters (MFKC)

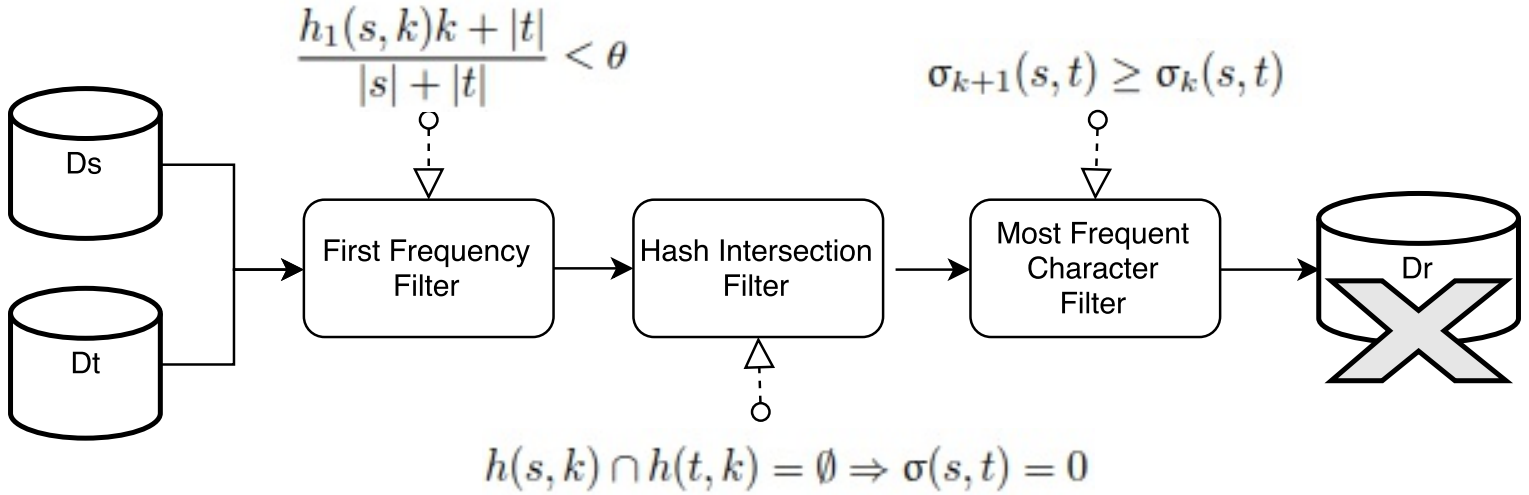
Approach

$$(s, t, k, \theta) = \frac{\sum_{c_i \in s,k \cap t,k} c_i, s + c_i, t}{|s| + |t|} \geq \theta \quad (1)$$

Goal

Derive efficient approach for $M = \{(s, t) \in S \times T : \sigma(s, t, K) \geq \theta\}$

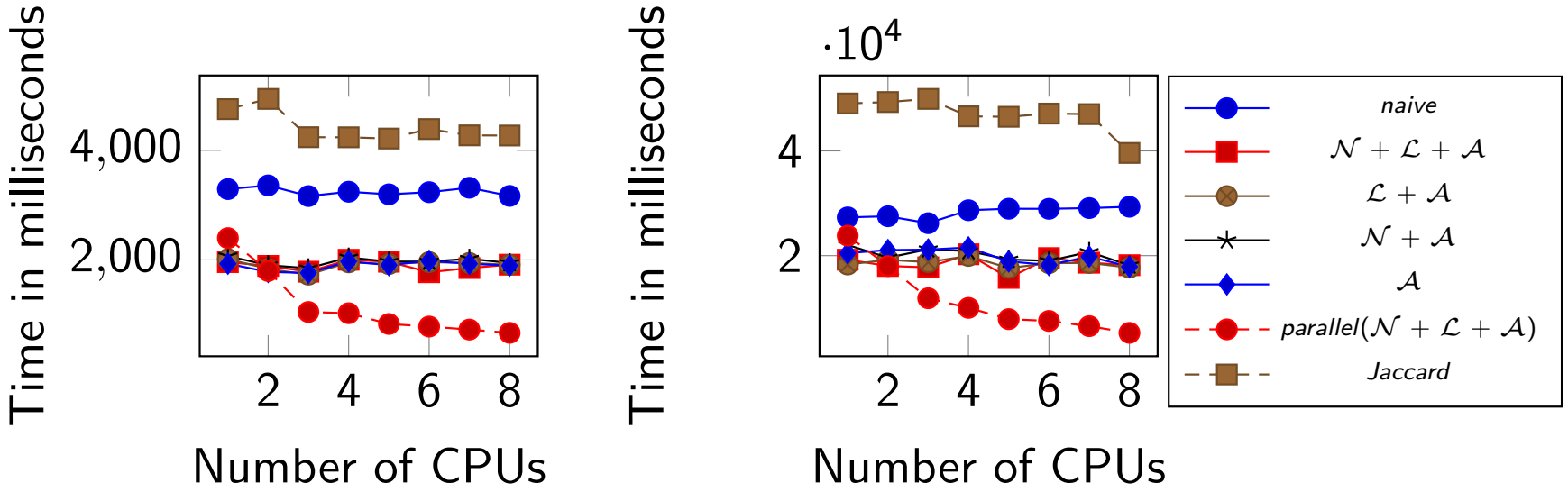
1. Hash intersection filter
2. Frequency-based filter
3. Most frequent character filter



Most Frequent K Characters (MFKC)

Results - Valdestilhas, ISWC, 2017

Runtime experiments, according to the number of CPU cores

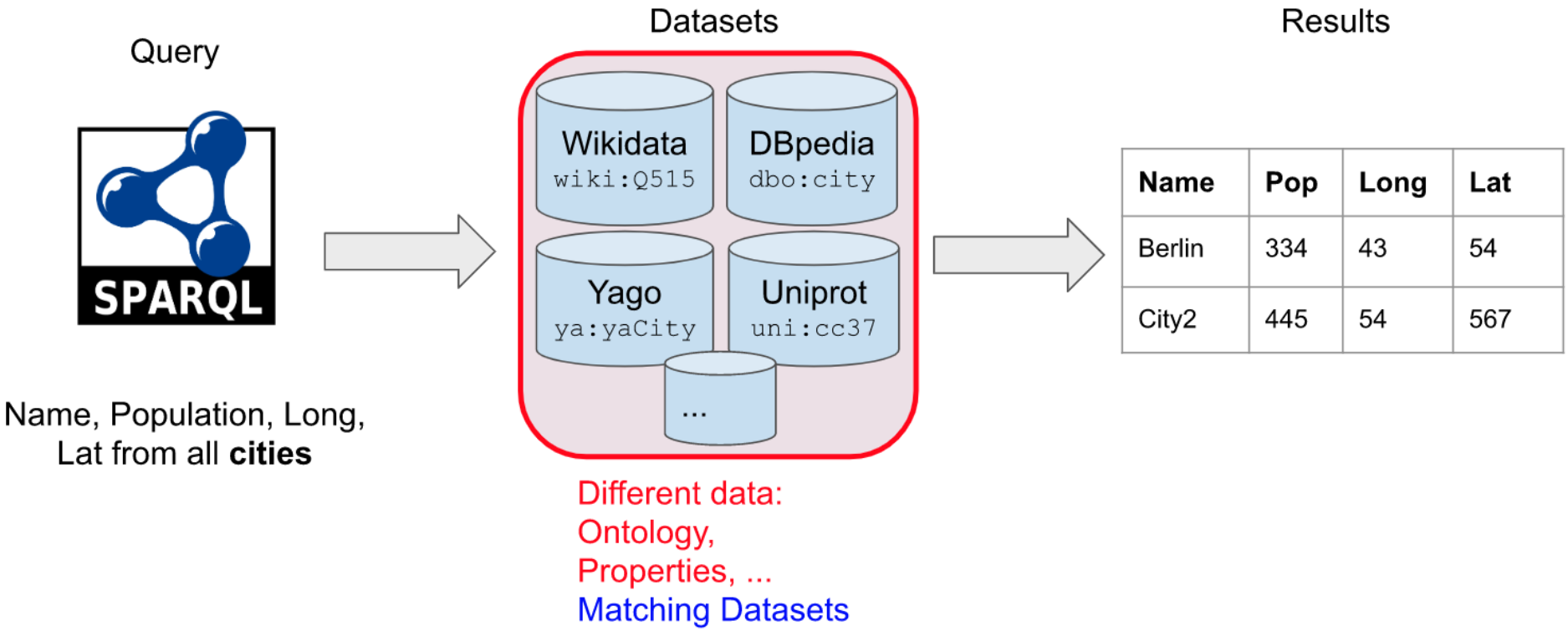


(a) $|D_s \times D_t| = 1,000,000$

(b) $|D_s \times D_t| = 10,000,000$

ReLOD - *The incremental LOD Dataset relation index*

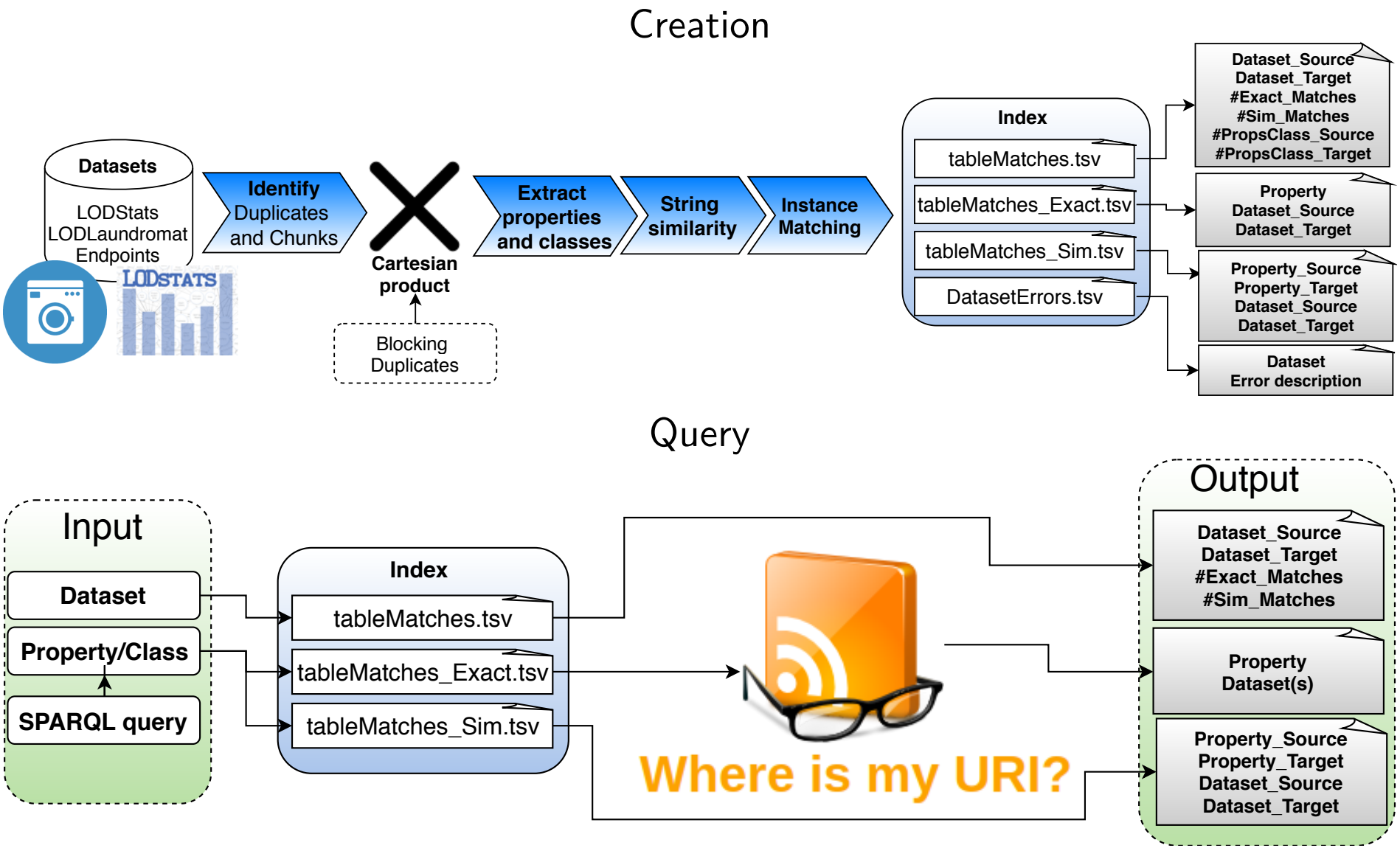
Research Questions



- How to identify and quantify similar datasets for a given Dataset?
- How many datasets are most likely to execute a given SPARQL query?
- How the detection of duplicated and chunk datasets can help in the process of matching a large amount of datasets?
- How ReLOD increase the number of datasets identified by wimuQ?

ReLOD - *The incremental LOD Dataset relation index*

Approach



ReLOD - *The incremental LOD Dataset relation index*

Results - Valdestilhas, SWJ, 2020

Top 10 datasets containing exact the same URI and containing the most similar URIs according to our similarity approach(MFKC) [1]

Dataset	**#ExactMatch**	**#sim > 0.8**	**#PropClass**
swdf	22	64	288
yago	6	65	373546
dblp	6	9	41
linkedgeodata	5	617	11799
wiktionary	4	6	31
geonames	4	12	27
wordnet	3	26	69
wikidata	0	5	427
freebase	0	55	17587

[1] #PropClass represents the total number of properties and classes from the dataset.

Querying

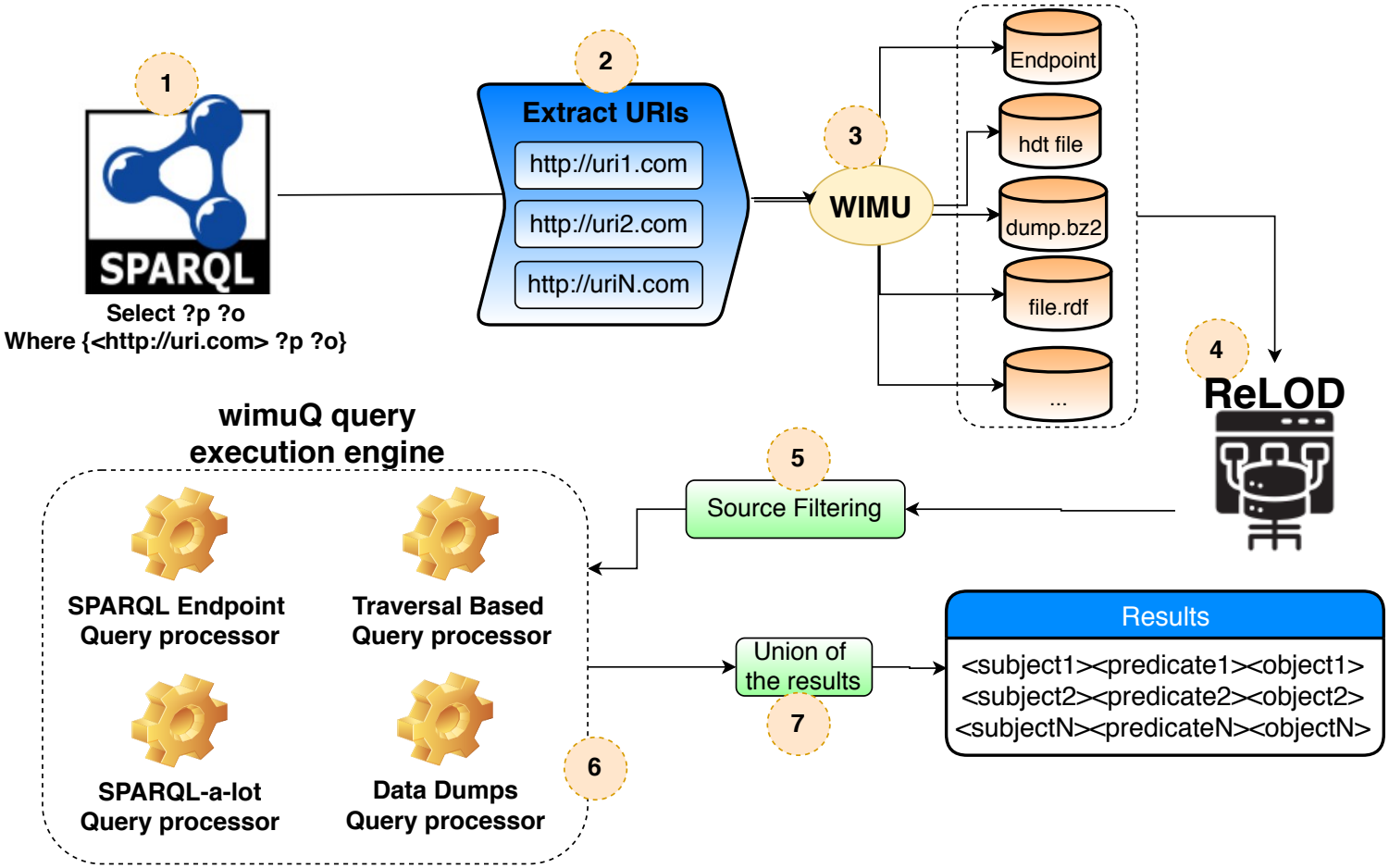
WimuQ - More Complete Resultset Retrieval from Large Heterogeneous RDF Sources

Research Questions

- Is it possible to Identify automatically relevant sources from heterogeneous RDF data, even with non-dereferenceable URIs, improving the resultset retrieval?
- Given a SPARQL query, how to know in which dataset(s) the query can be executed?

WimuQ - More Complete Resultset Retrieval from Large Heterogeneous RDF Sources

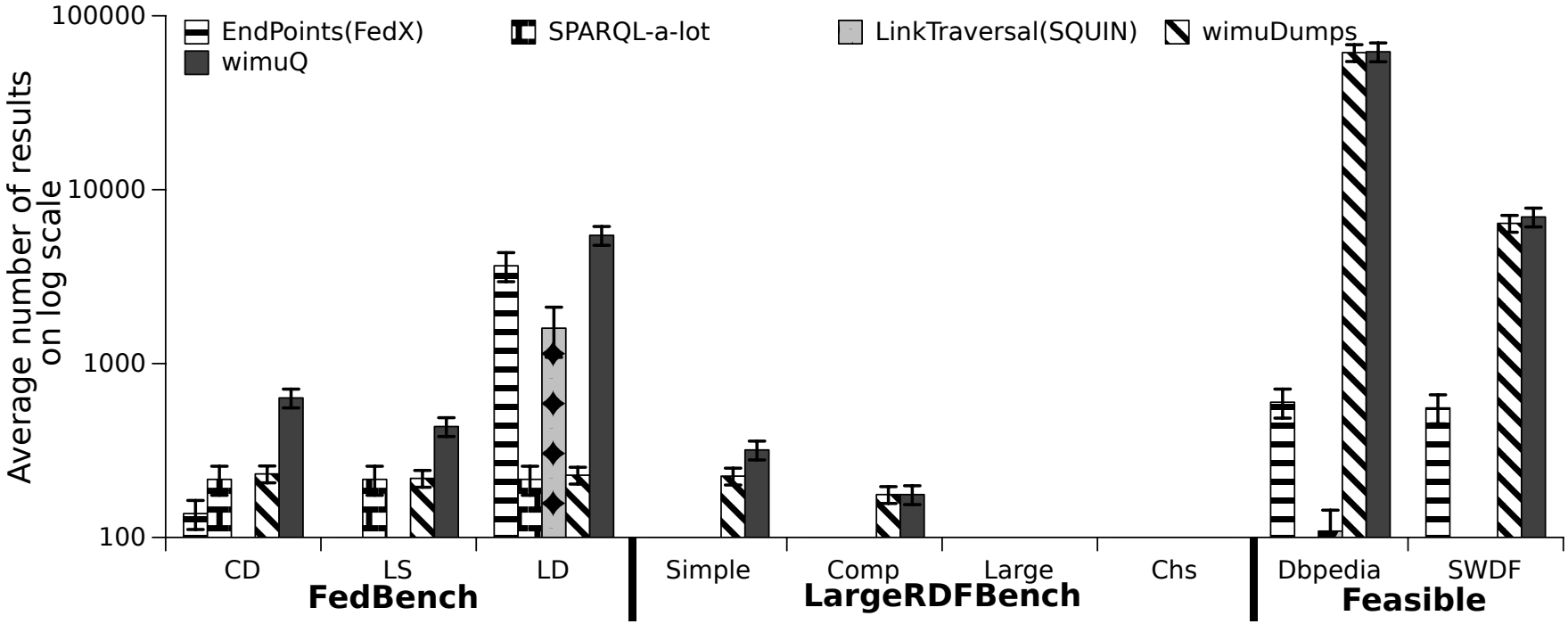
Approach



WimuQ - More Complete Resultset Retrieval from Large Heterogeneous RDF Sources

Results - Valdestilhas, KCap, 2019

Coverage: Overall 76% queries with results(Zero results=non-public endpoints/data - non-indexed)



Approaches and the best coverage

FedBench 55% endpoints

LargeRDFBench 81% wimuDumps

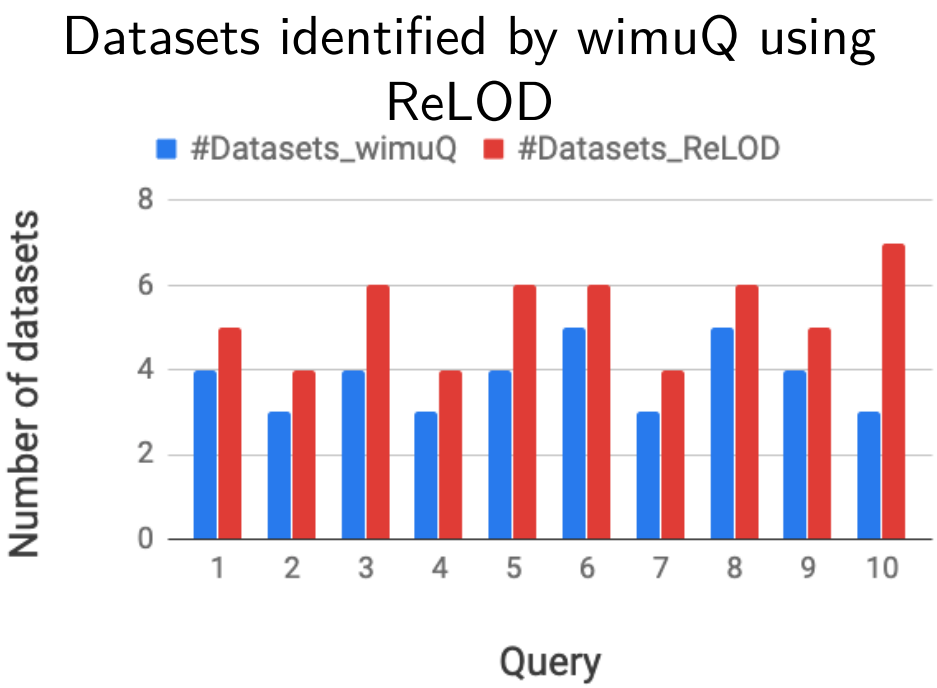
FEASIBLE 98% wimuDumps

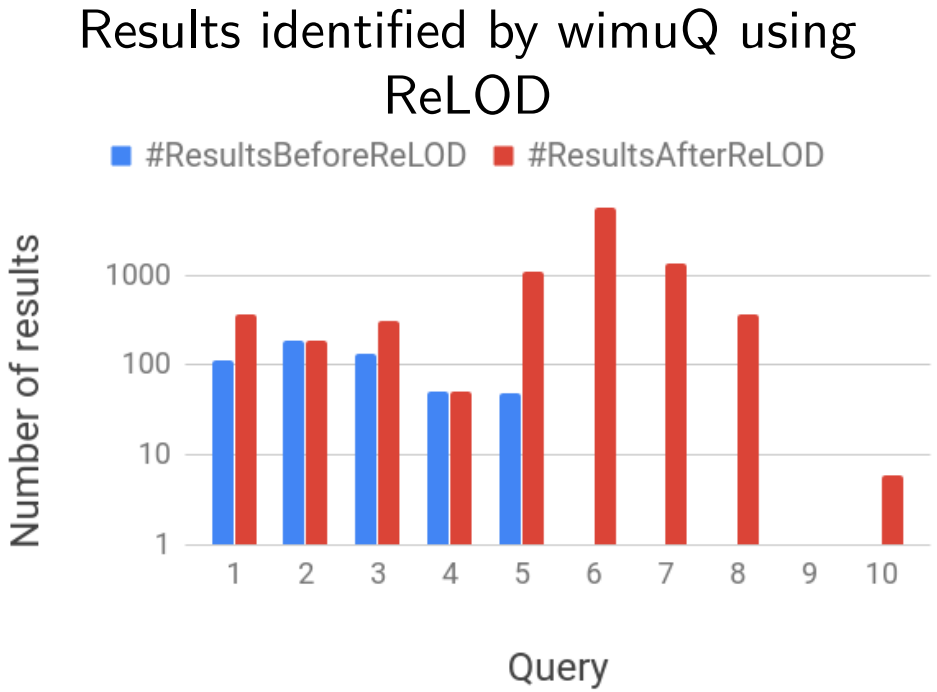
Observation

The combination of those query processing engines gives more resultset retrieval

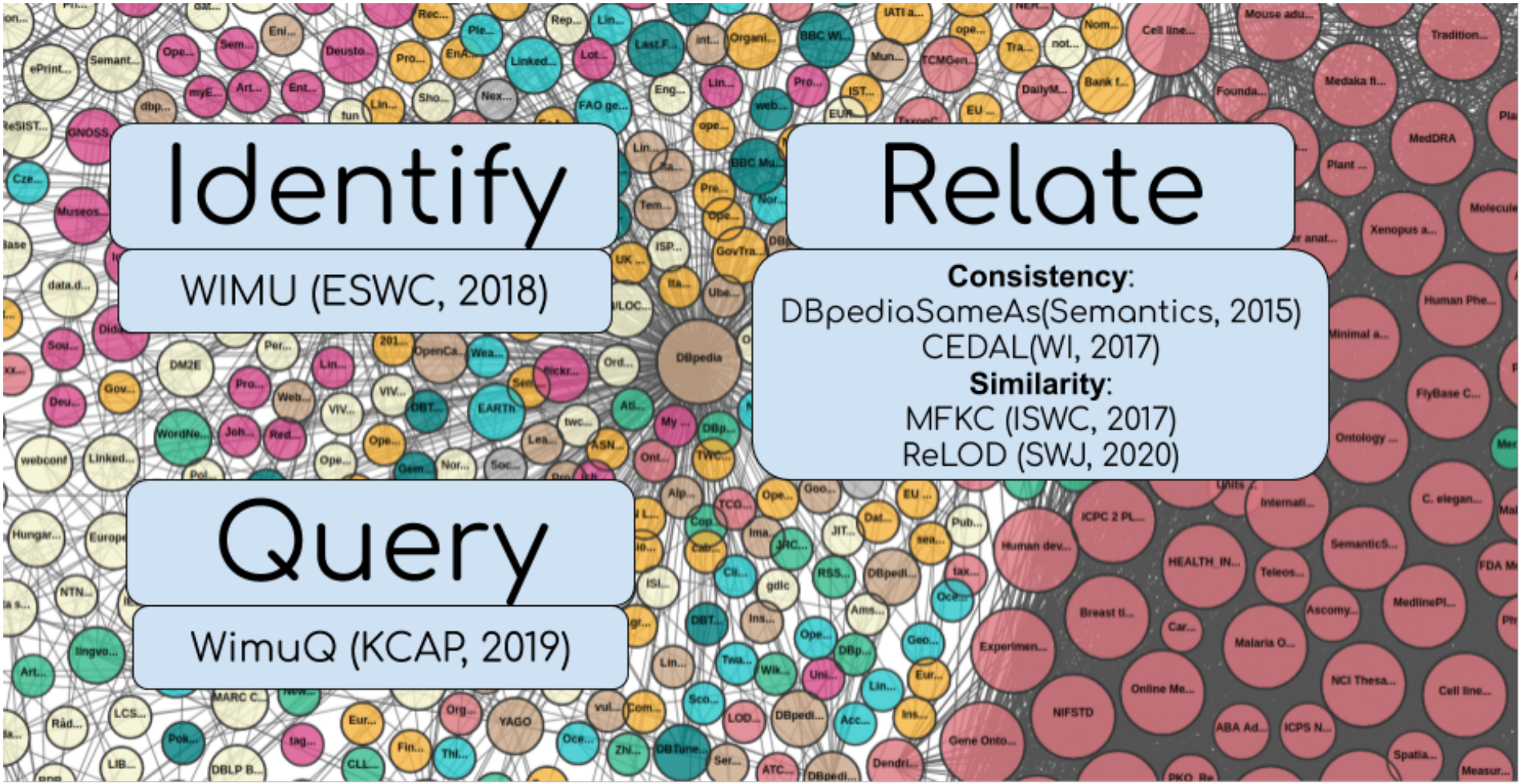
Improvements of wimuQ using ReLOD

Results - Valdestilhas, SWJ, 2020





Conclusion


Identify
WIMU (ESWC, 2018)
Relate
Consistency:
DBpediaSameAs(Semantics, 2015)
CEDAL(WI, 2017)
Similarity:
MFKC (ISWC, 2017)
ReLOD (SWJ, 2020)
Query
WimuQ (KCAP, 2019)

Publications and Statistics

- 6 papers First author
- 6 papers as co-author
- 1 Project proposal
- 2 awards
- Google h-index 5

That's all Folks!

Thanks!

Questions?

Contact: valdestilhas@informatik.uni-leipzig.de

Special thanks to my PhD. advisor Prof. Dr. rer. nat. Thomas Riechert