# Effective and Scalable Spatial RDF Data Management and Growth

Nikos Mamoulis

Department of Computer Science and Engineering

University of Ioannina

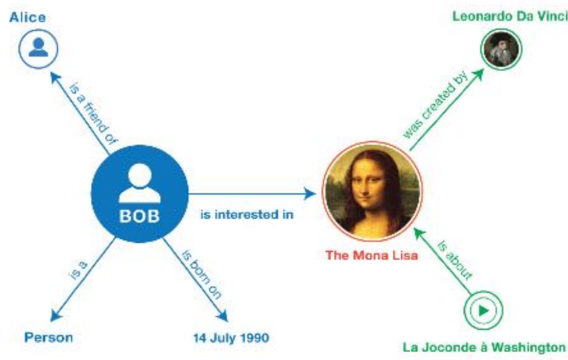
# Outline

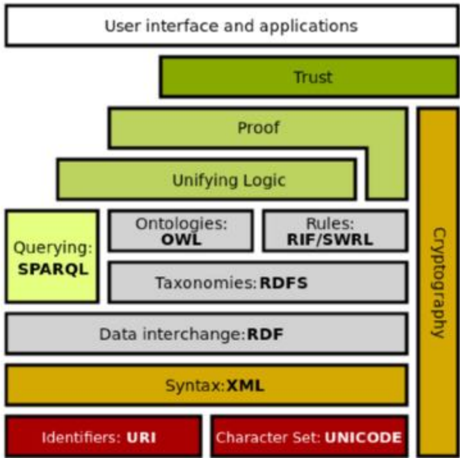
- RDF data and the Semantic Web
- Spatially enriched RDF data
- Spatial RDF data management

  [LMBT14, TLM+19]
- Keyword search on Spatial RDF data

  [SWM16,  CKF+20, WZSM20, KFM21]
- Linking geospatial data

  [KVM20, PMMK21]



# Resource Description Framework (RDF)

- Models information as a collection of <subject, predicate, object> triples

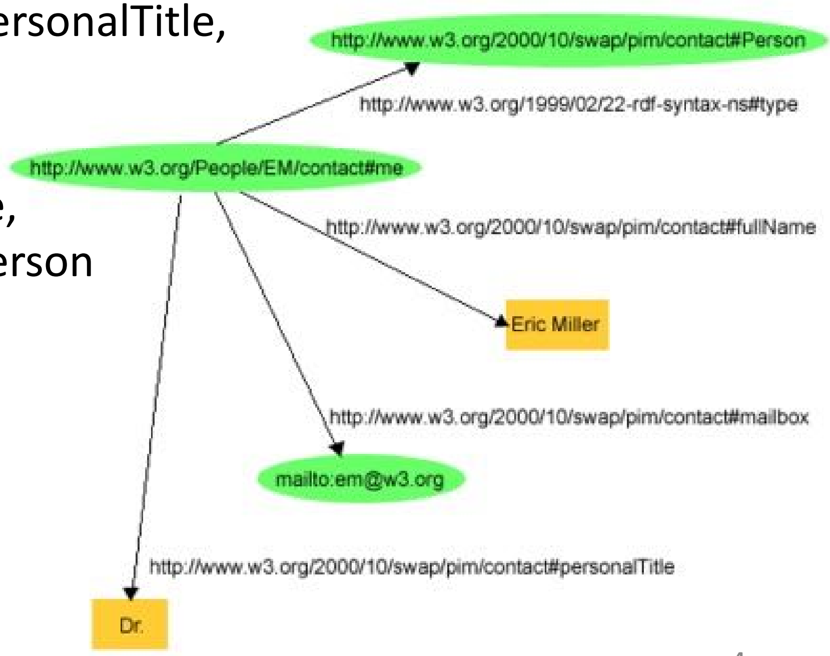
  Dresden hosted Wagner

  Wagner hasName "Richard Wagner"

- Subjects are *resources* (entities)
- Objects can be resources or *literals*
- Resources and predicates are identified by URIs

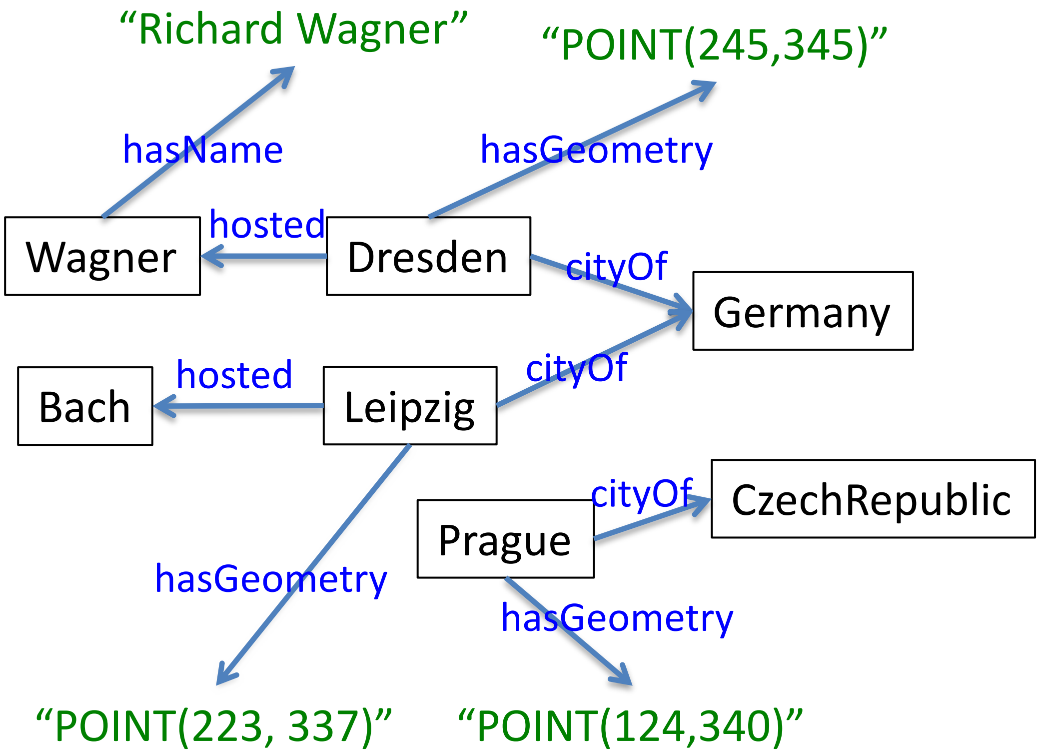
# Example (from Wikipedia)

- http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#fullName, "Eric Miller"
- http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#mailbox, mailto:e.miller123(at)example
- http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#personalTitle, "Dr."
- http://www.w3.org/People/EM/contact#me, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/10/swap/pim/contact#Person



# RDF data as a graph

| *subject* | *property* | *object* |
|---|---|---|
| Dresden | cityOf | Germany |
| Prague | cityOf | CzechRepublic |
| Leipzig | cityOf | Germany |
| Dresden | hosted | Wagner |
| Leipzig | hosted | Bach |
| Wagner | hasName | "Richard Wagner" |
| Wagner | performedIn | Leipzig |
| Dresden | hasGeometry | "POINT (...)" |
| Prague | hasGeometry | "POINT (...)" |
| Leipzig | hasGeometry | "POINT (...)" |
| ... | ... | ... |




In this example, URIs are simplified

# RDF in practice

- DBpedia - Extracts facts from Wikipedia articles and publishes them as RDF data.
- YAGO is a huge semantic knowledge base derived from Wikipedia, WordNet and GeoNames.
- Creative Commons - Uses RDF to embed license information in web pages and mp3 files.
- LinkedGeoData uses the information collected by the OpenStreetMap project and makes it available as an RDF knowledge base.
  It interlinks this data with other knowledge bases in the Linking Open Data initiative.

# Queries on RDF data

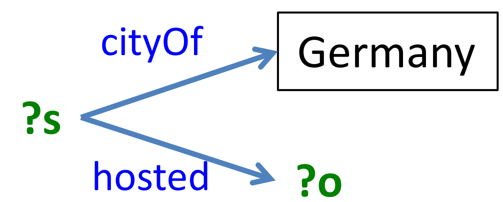
- SPARQL is the standard query language
- Queries in the form of subgraph patterns

**SPARQL query**

```
Select ?s ?o
Where {
        ?s cityOf Germany .
        ?s hosted ?o .

}
```

**graph representation of query**




**result**

| ?s | ?o |
|---|---|
| Dresden | Wagner |
| Leipzig | Bach |

# RDF-3X [Neumann & Weikum, 2008]

- A prototype open-source RDF store that efficiently supports generic SPARQL queries
- URIs and Literals are mapped to unique IDs, by a dictionary
- RDF data are represented as triples of IDs

**Dictionary**

| *ID* | *URI/literal* |
|---|---|
| 1 | Dresden |
| 2 | cityOf |
| 3 | Germany |
| 4 | Prague |
| 5 | CzechRepublic |
| 6 | Leipzig |
| ... | ... |

**ID-encoded triples**

| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 2 | 5 |
| 6 | 2 | 3 |
| ... | ... | ... |

# RDF-3X [Neumann & Weikum, 2008]

- A clustered B+-tree index for each SPO permutation

SPO

| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 2 | 5 |
| 6 | 2 | 3 |
| ... | ... | ... |

SOP

| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 2 | 5 |
| 6 | 2 | 3 |
| ... | ... | ... |

PSO

| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 2 | 5 |
| 6 | 2 | 3 |
| ... | ... | ... |

POS

| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 6 | 2 | 3 |
| 4 | 2 | 5 |
| ... | ... | ... |

OSP

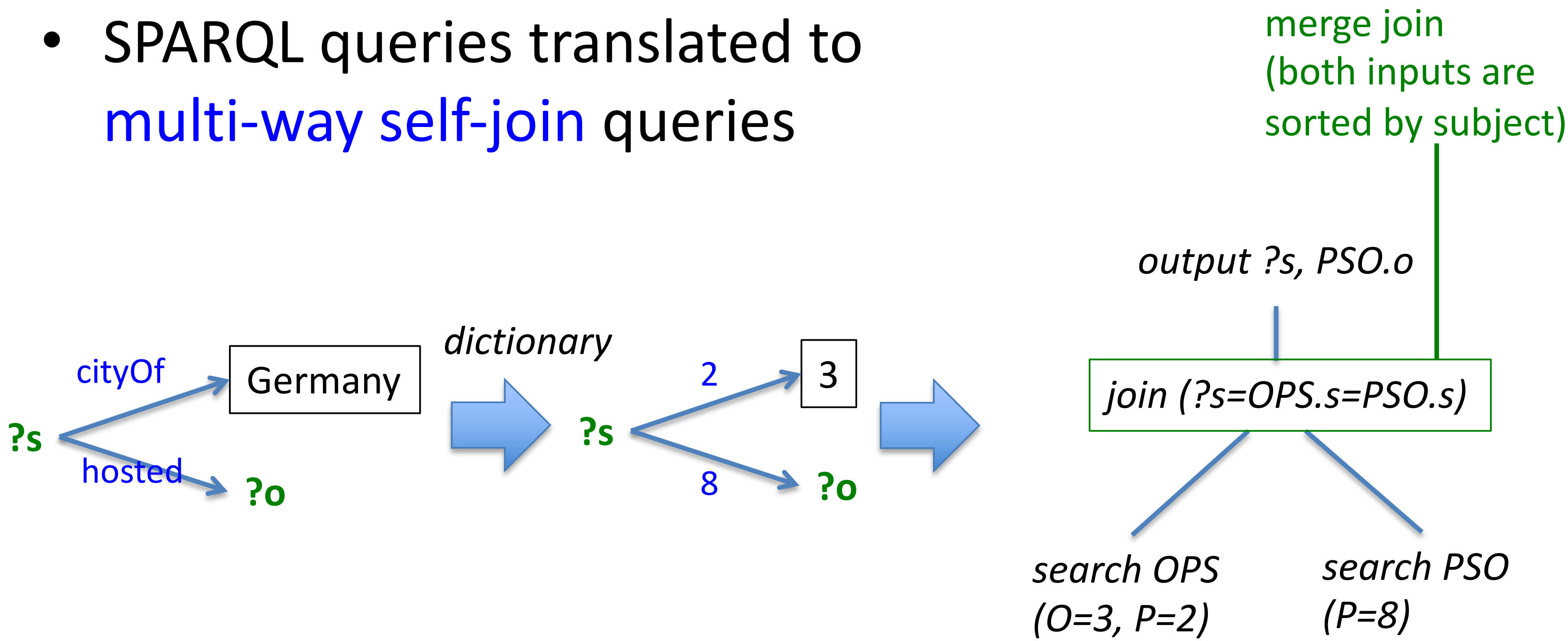
| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 6 | 2 | 3 |
| 4 | 2 | 5 |
| ... | ... | ... |

OPS

| *subject* | *property* | *object* |
|---|---|---|
| 1 | 2 | 3 |
| 6 | 2 | 3 |
| 4 | 2 | 5 |
| ... | ... | ... |

# RDF-3X [Neumann & Weikum, 2008]

- SPARQL queries translated to multi-way self-join queries




- RDF-3X favors plans that produce *interesting orders*, (merge joins are pipelined without intermediate sorts)

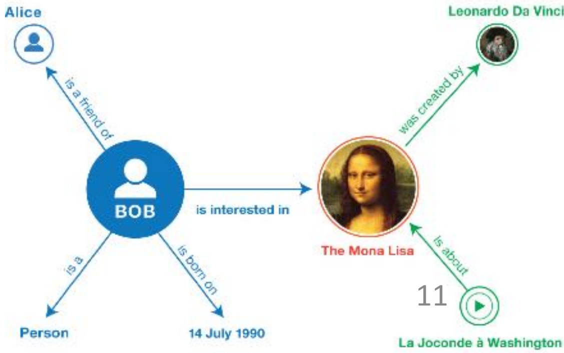
# Outline

- RDF data and the Semantic Web
- Spatially enriched RDF data
- Spatial RDF data management

  [LMBT14, TLM+19]
- Keyword search on Spatial RDF data

  [SWM16, CKF+20, WZSM20, KFM21]
- Linking geospatial data

  [KVM20, PMMK21]



# Spatial Information in RDF data

GeoSPARQL

- GeoSPARQL extends RDF and SPARQL to represent geographic information and support spatial queries
- Spatial entities linked to spatial literals (e.g. points, polygons) via predicates such as **`hasGeometry`**

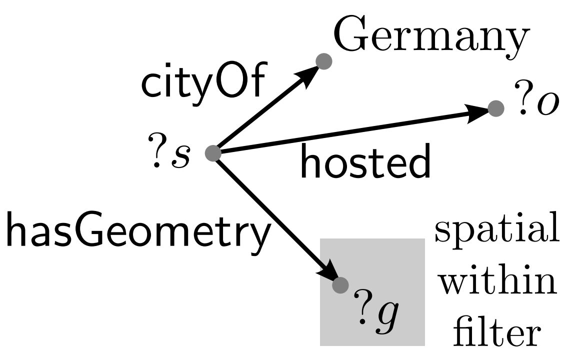
| *subject* | *property* | *object* |
|---|---|---|
| ... | ... | ... |
| Dresden | hasGeometry | "POINT (...)" |
| Prague | hasGeometry | "POINT (...)" |
| Leipzig | hasGeometry | "POINT (...)" |
| ... | ... | ... |

spatial entities

# Examples of GeoSPARQL queries

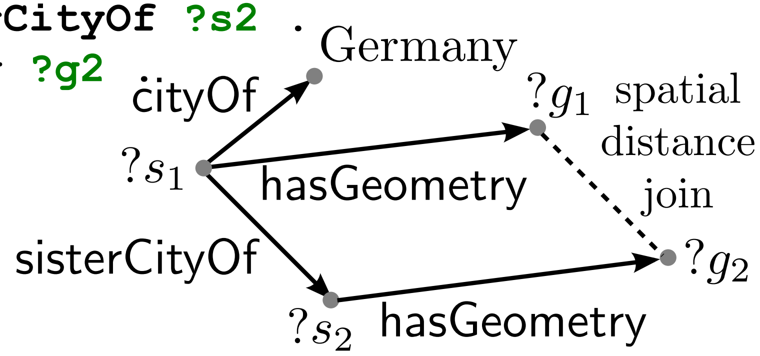
**spatial range predicate**

```
Select ?s ?o
Where ?s cityOf Germany .
?s hosted ?o . ?s hasGeometry ?g .
Filter WITHIN(?g, "POLYGON(...)");
```

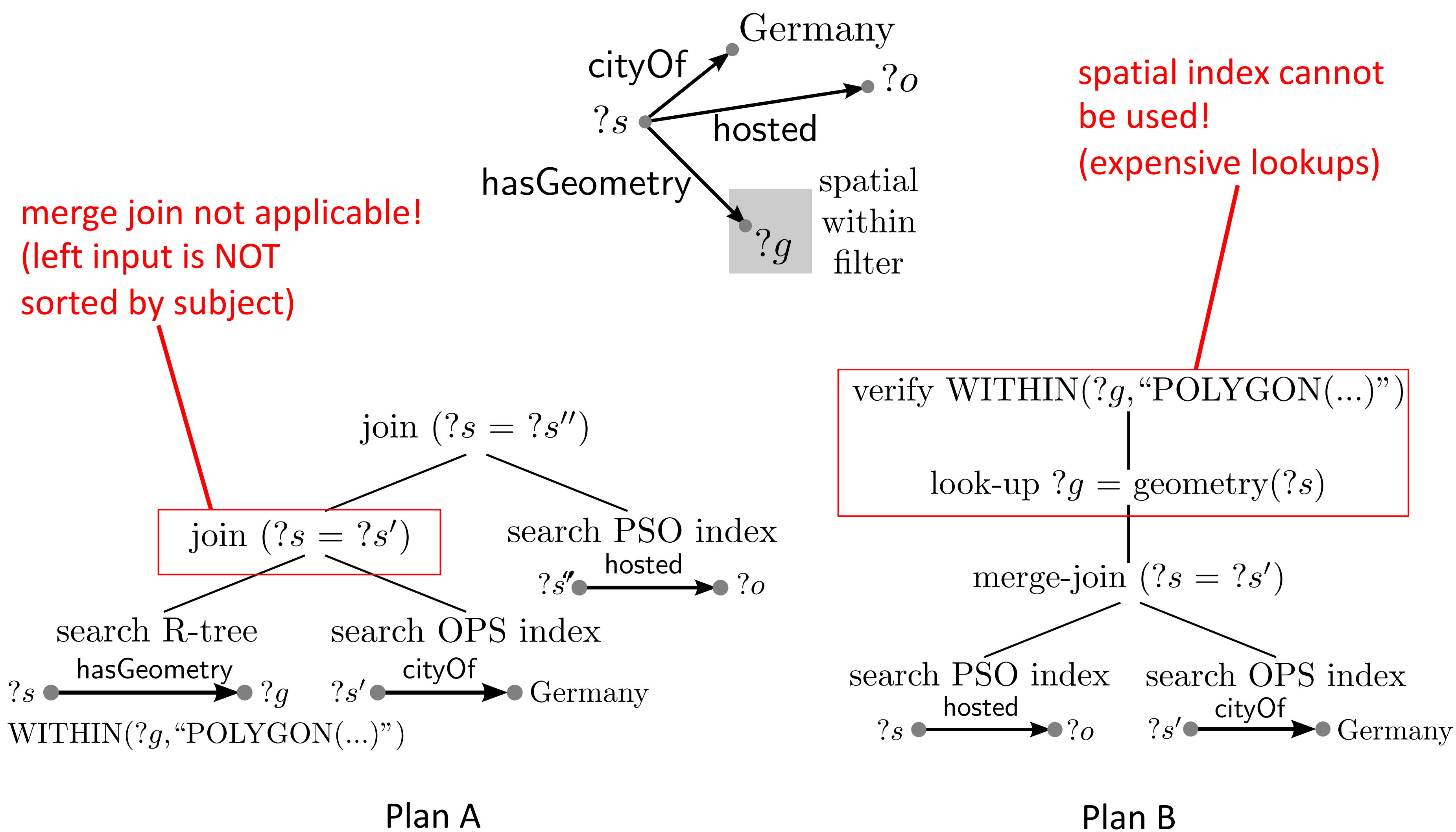



**spatial join predicate**

```
Select ?s1 ?s2
Where ?s1 cityOf Germany . ?s1 sisterCityOf ?s2 .
?s1 hasGeometry ?g1 . ?s2 hasGeometry ?g2
Filter DISTANCE(?g1, ?g2) < "300km";
```



# Previous work on GeoSPARQL support



# Observation

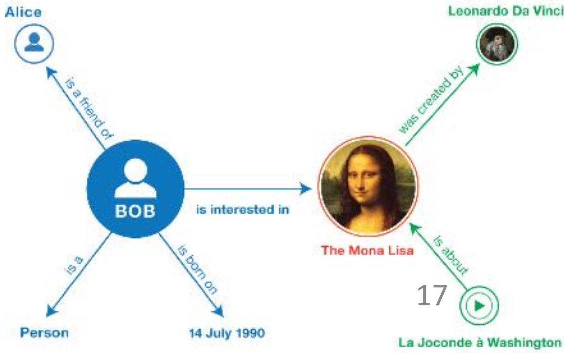
- The IDs given to resources or literals at the dictionary mapping do not carry any semantics

## Idea 

- Encode the (approximate) locations of spatial entities into the IDs

# Outline

- RDF data and the Semantic Web
- Spatially enriched RDF data
- Spatial RDF data management
  [LMBT14, TLM+19]
- Keyword search on Spatial RDF data
  [SWM16, CKF+20, WZSM20, KFM21]
- Linking geospatial data
  [KVM20, PMMK21]



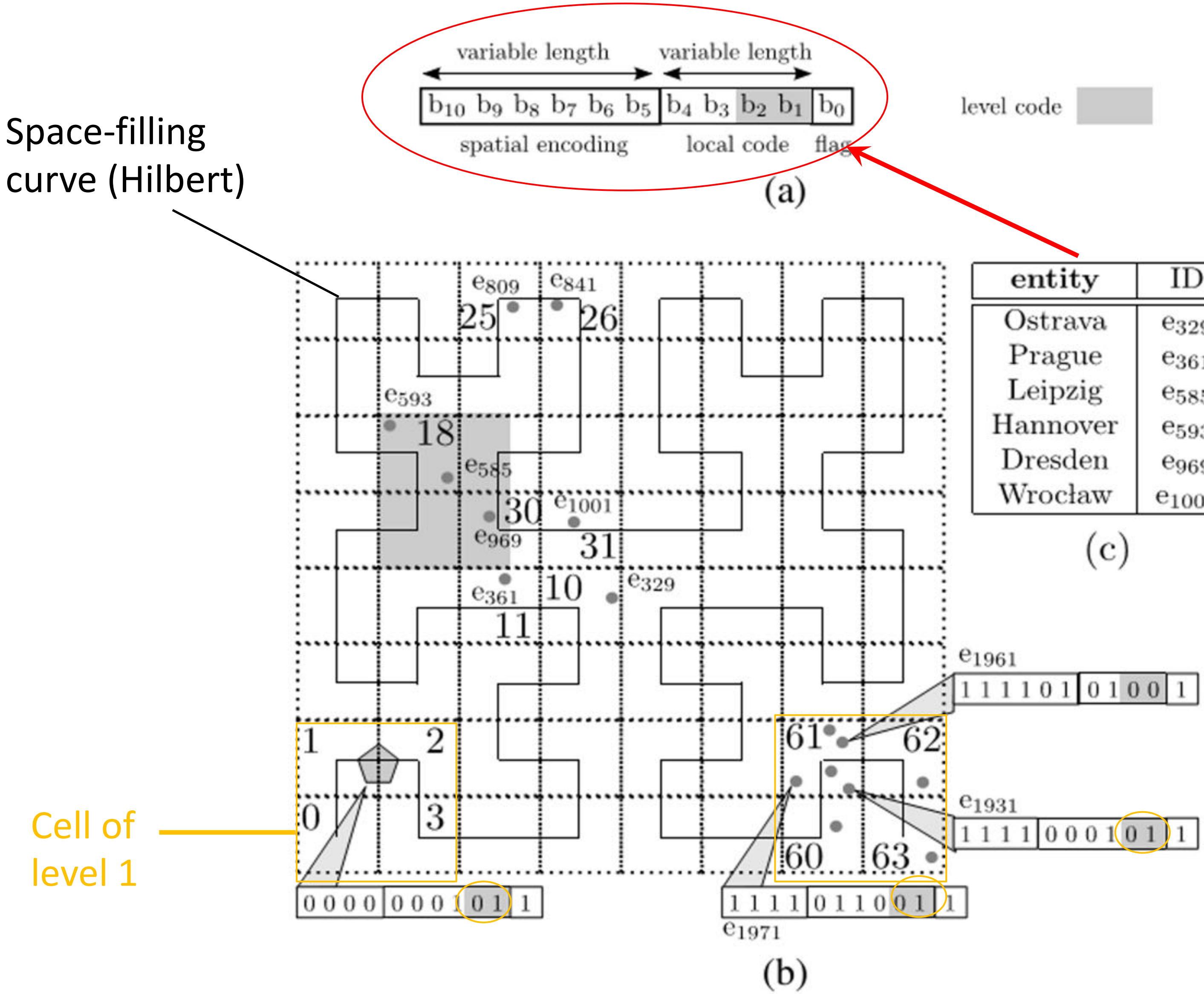
# Our Contributions

- An encoding scheme for RDF spatial entities
- Spatial filters can be applied on the fly during graph pattern matching
- Two spatial join algorithms that operate directly on the IDs of the entities (spatial merge-join, spatial hash-join)
- Two k-NN algorithms that use the IDs to avoid fetching geometries as much as possible
- An extension of RDF-3X that includes the above
- Extension of RDF-3X's query optimizer to consider all these new operators

[LMBT14] J. Liagouris, N. Mamoulis, P. Bouros, and M. Terrovitis, "An Effective Encoding Scheme for Spatial RDF Data," *Proceedings of the VLDB Endowment (**PVLDB**)*, 7(12):1271-1282, 2014.

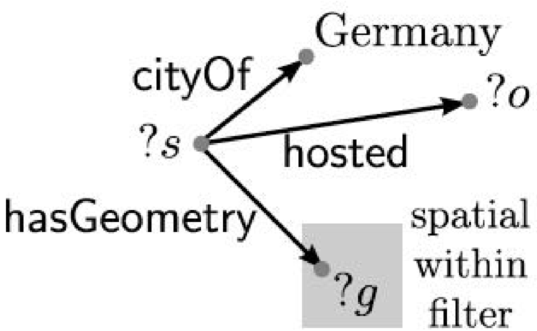
[TLM+19] K. Theocharidis, J. Liagouris, N. Mamoulis, P. Bouros, and M. Terrovitis, "SRX: Efficient Management of Spatial RDF Data," *The **VLDB Journal***, 28(5): 703-733, October 2019.

# Our Encoding Scheme

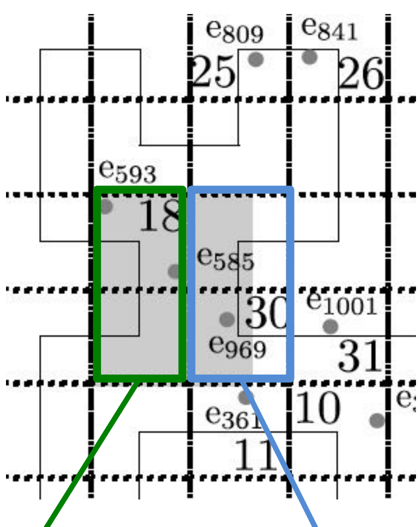



| entity | ID |
|---|---|
| Ostrava | $e_{329}$ |
| Prague | $e_{361}$ |
| Leipzig | $e_{585}$ |
| Hannover | $e_{593}$ |
| Dresden | $e_{969}$ |
| Wrocław | $e_{1001}$ |

# Spatial Selection Filter

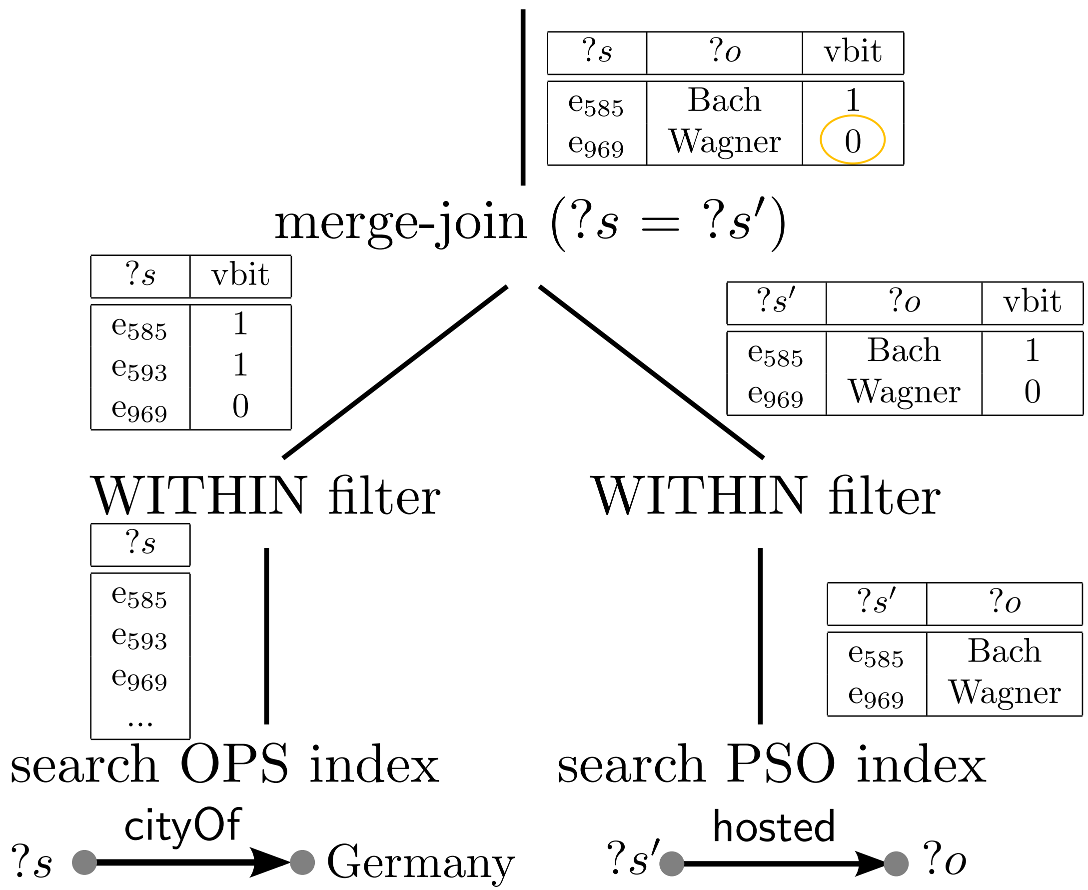



Query




guaranteed region (vbit=1)

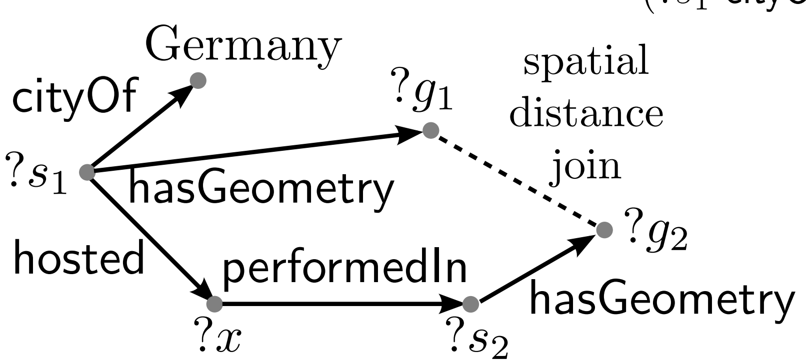
candidate region (vbit=0)

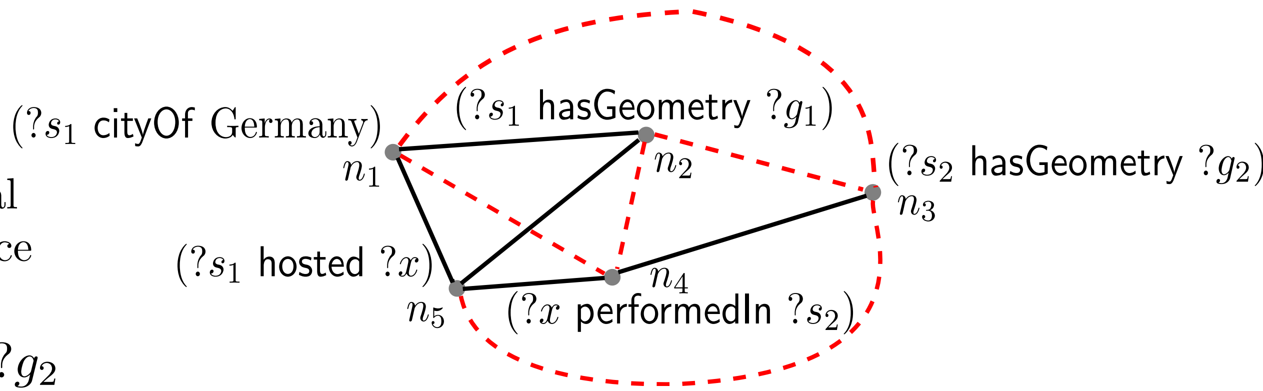



Evaluation Plan

# Query Optimization

- Extension of RDF-3X query optimizer to consider
  - Selectivity of spatial operations (use of spatial statistics)
  - New spatial join algorithms (spatial merge join)
  - Application of spatial filters
- Join graph is augmented accordingly
  - E.g. pairs of nodes ($n_1$,$n_4$) can be spatially joined



# Remarks

- Our encoding-based approach can be easily incorporated into any RDF triple store
- The average performance gains of the Encoding-based approach with respect to previous work are:
  - Queries with WITHIN predicates:
    - LGD: **53%** with cold cache and **68%** with warm cache
    - YAGO2: **35%** with cold cache and **60%** with warm cache
  - Queries with DISTANCE predicates (spatial joins):
    - LGD: **65%** with cold cache and **75%** with warm cache
    - YAGO2: **19%** with cold cache **21%** with warm cache
- The overhead in the optimization time is negligible with respect to the original query engine

yago select knowledge

Linked GeoData

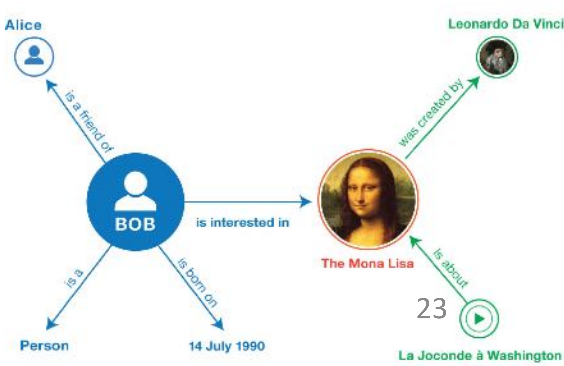
# Outline

- RDF data and the Semantic Web
- Spatially enriched RDF data
- Spatial RDF data management

  [LMBT14, TLM+19]
- Keyword search on Spatial RDF data

  [SWM16,  CKF+20, WZSM20, KFM21]
- Linking geospatial data

  [KVM20, PMMK21]

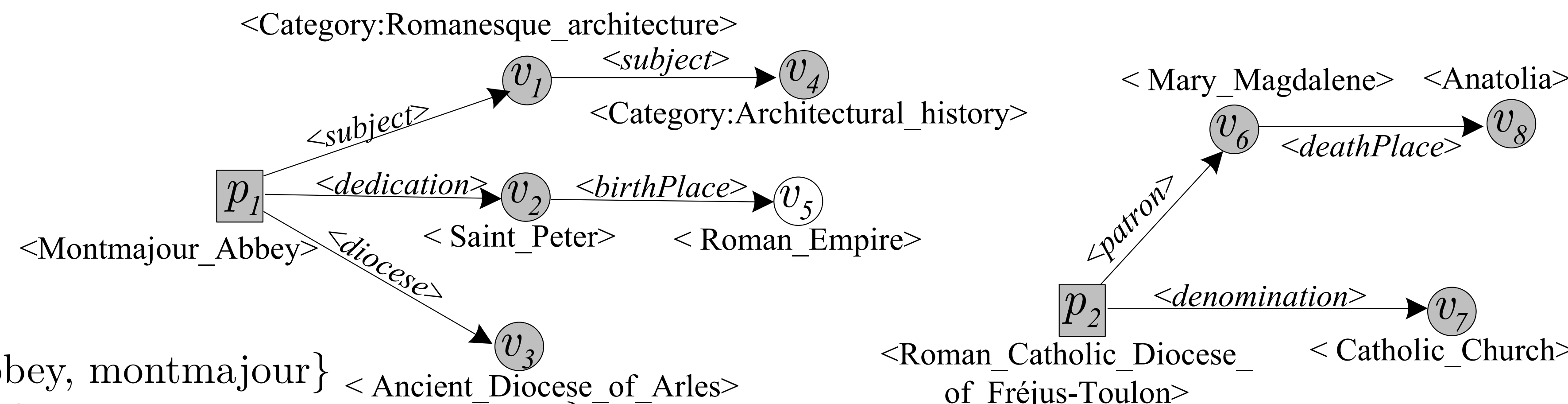


# Keyword search on RDF data

- Motivation: SPARQL is not user-friendly, unless you know the data
  - User must know predicates, URIs of subjects
- Keyword search only asks the user to input a set of keywords
  - Results are small RDF subgraphs that contain all keywords
  - Special case of keyword search in general graphs

# Keyword search on RDF data

- Data preparation:
  - Each entity is associated to a textual description extracted from its URI, predicates, and literals.
  - Literals + connecting predicates are eliminated
  - Resulting graph only includes entities and their relationships




$p_1$: {abbey, montmajour}
$v_1$: {architecture, romanesque, subject}
$v_2$: {catholic, dedication, peter, roman, saint}
$v_3$: {ancient, arles, diocese}
$v_4$: {architectural, history, subject}
$v_5$: {ancient, birthplace, empire, roman}

$p_2$: {catholic, diocese, roman}
$v_6$: {mary, magdalene, patron}
$v_7$: {catholic, church, denomination, history}
$v_8$: {anatolia, ancient, deathPlace, history}

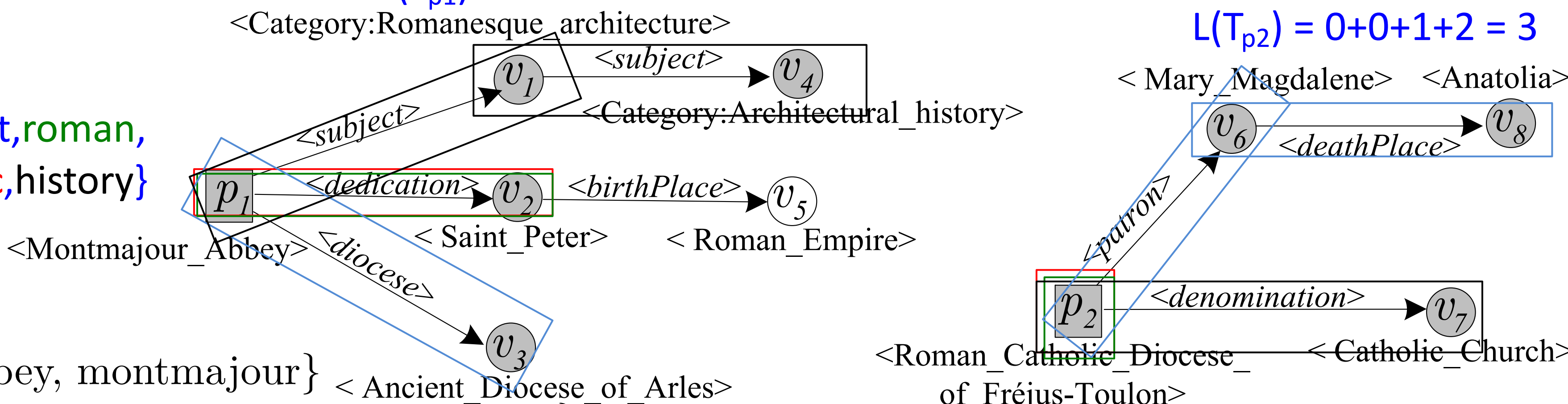
# Keyword search on RDF data

- A query result is a sub-tree of the RDF graph
- The sum of the lengths of the paths that connect the keyword instances to the root define a looseness score (the smaller the better)

Query: {ancient, roman, catholic, history}

$L(T_{p1}) = 1+1+1+2 = 5$

$L(T_{p2}) = 0+0+1+2 = 3$




$p_1$: {abbey, montmajour}
$v_1$: {architecture, romanesque, subject}
$v_2$: {catholic, dedication, peter, roman, saint}
$v_3$: {ancient, arles, diocese}
$v_4$: {architectural, history, subject}
$v_5$: {ancient, birthplace, empire, roman}

$p_2$: {catholic, diocese, roman}
$v_6$: {mary, magdalene, patron}
$v_7$: {catholic, church, denomination, history}
$v_8$: {anatolia, ancient, deathPlace, history}

# Our Contributions

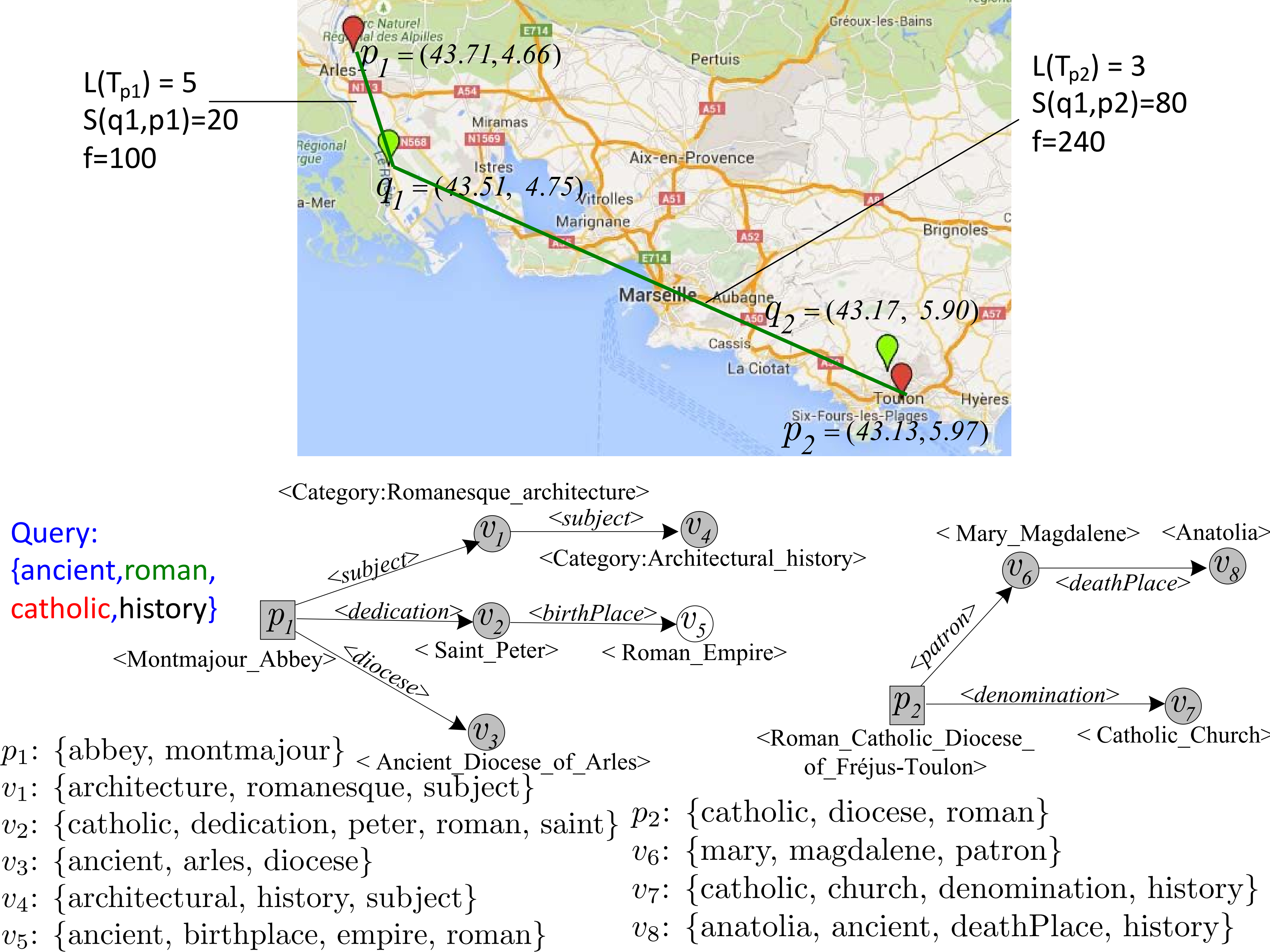
- A definition of spatial keyword search on RDF data
- Pruning rules based on reachability indexes and dynamic bounds
- A data preprocessing technique that further accelerates search
- Extensions that add temporal semantics to queries and support diversification/proportionality of results

[SWM16] J. Shi, D. Wu, and N. Mamoulis "Top-k Relevant Semantic Place Retrieval on Spatial RDF Data," ***SIGMOD*** 2016.
[WZSM20] D. Wu, H. Zhou, J. Shi, and N. Mamoulis, "Top-k Relevant Semantic Place Retrieval on Spatio-temporal RDF Data," *The* ***VLDB Journal***, 29(4): 893-917, July 2020.
[CKF+20] Z. Cai, G. Kalamatianos, G. J. Fakas, N. Mamoulis, and D. Papadias, "Diversified spatial keyword search on RDF data," *The* ***VLDB Journal***, 29(5): 1171-1189, September 2020.
[KFM21] G. Kalamatianos, G. J. Fakas, and N. Mamoulis, "Proportionality in Spatial Keyword Search," ***SIGMOD*** 2021.

# Spatial Keyword search on RDF data

- Applicable when users search for places near them using keywords
  - Query q: a location q.λ and a set of keywords q.w
- Search for RDF subgraphs:
  - rooted at vertices that are places (spatial entities)
  - contain all query keywords
- Relevance score $f(T_p)$ of subgraph $T_p$ rooted at place p is done based on a function that combines
  - looseness score of $T_q$ w.r.t. query keywords q.w
  - spatial distance S(q.λ,p) of p to query location q.λ

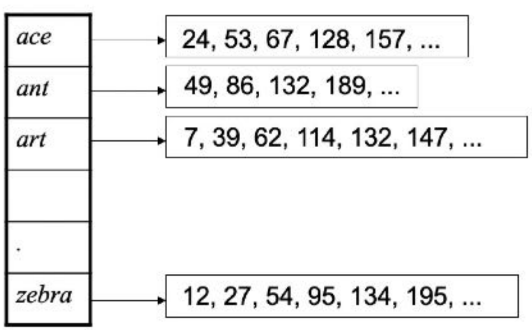
$$f(T_p) = L(T_p) \cdot S(q.\lambda, p)$$

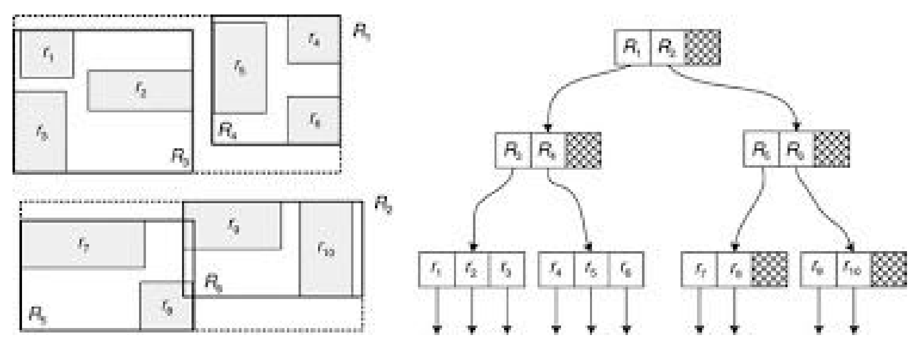


L(Tp1) = 5
S(q1,p1)=20
f=100
L(Tp2) = 3
S(q1,p2)=80
f=240
p1 = (43.71, 4.66)
q1 = (43.51, 4.75)
q2 = (43.17, 5.90)
p2 = (43.13, 5.97)
Query:
{ancient, roman, catholic, history}
<Category:Romanesque_architecture>
<subject>
<Category:Architectural_history>
<dedication>
<birthPlace>
<diocese>
<Montmajour_Abbey>
< Saint_Peter>
< Roman_Empire>
< Ancient_Diocese_of_Arles>
< Mary_Magdalene>
<Anatolia>
<deathPlace>
<patron>
<denomination>
<Roman_Catholic_Diocese_of_Fréjus-Toulon>
< Catholic_Church>
p1: {abbey, montmajour}
v1: {architecture, romanesque, subject}
v2: {catholic, dedication, peter, roman, saint}
v3: {ancient, arles, diocese}
v4: {architectural, history, subject}
v5: {ancient, birthplace, empire, roman}
p2: {catholic, diocese, roman}
v6: {mary, magdalene, patron}
v7: {catholic, church, denomination, history}
v8: {anatolia, ancient, deathPlace, history}

# Data Representation and Indexing

- RDF data are represented in their native graph form (i.e. use of adjacency lists) in memory
  - keyword queries require graph browsing
- Index textual content of graph vertices by an inverted file
- Spatially index place vertices by an R-tree

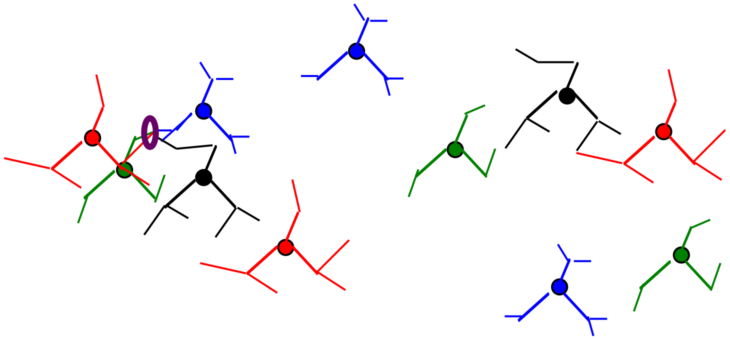


# Previous work: Bottom-up algorithm for keyword search in graphs

- For each query keyword w find vertices that contain w
- Apply concurrent breadth first search from each vertex
- As soon as ALL searches from different keywords meet, start getting results

Query:
{ancient,roman,
catholic,history}



# Top-down algorithm for top-k spatial keyword search

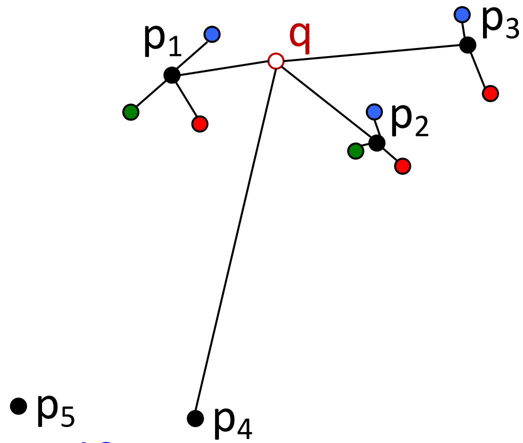
- Bottom-up algorithm (previous work)
  - does not guarantee that the trees found first include place vertices
  - does not necessarily find places near the user's location
- Top-down algorithm (ours):
  - considers place vertices in increasing spatial distance to query location (use spatial R-tree index)
  - for each such vertex p computes tree $T_p$ and $L(T_p)$, by BFS form p
  - stops when next vertices have no chance to outrank current top-k results (use of bounds)

# Top-down algorithm for top-k spatial keyword search

- The score θ of the k-th best place so far is the termination bound
  - If the for the next place p, $S(q,p) \geq \theta$, terminate

$S(q,p_1)=3$
$L(p_1)=5$
$f = 15$

$S(q,p_2)=4$
$L(p_2)=3$
$f = 12$




$S(q,p_3)=5$
$L(p_3)=?$

$S(q,p_4)=13$

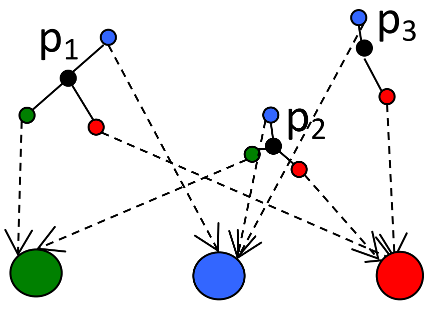
current top-1: $\cancel{p_1}$ $p_2$ with score $\cancel{15}$ 12

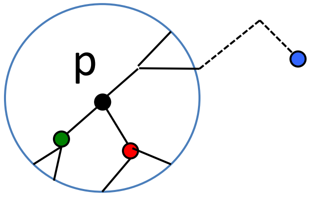
# Improvements

(1) Avoid BFS from place vertices from which some of the query keywords cannot be reached (use of reachability index)

(2) During BFS use a dynamically computed lower bound for its looseness L(p) to potentially prune p

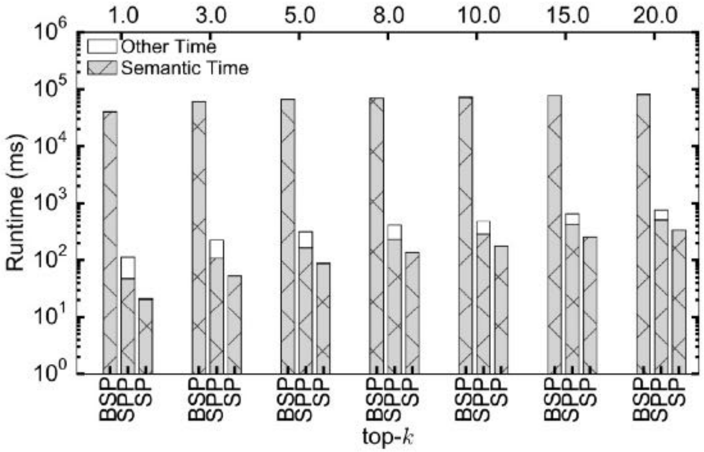
(3) Precompute for each place and for the R-tree nodes a summary of nearby words



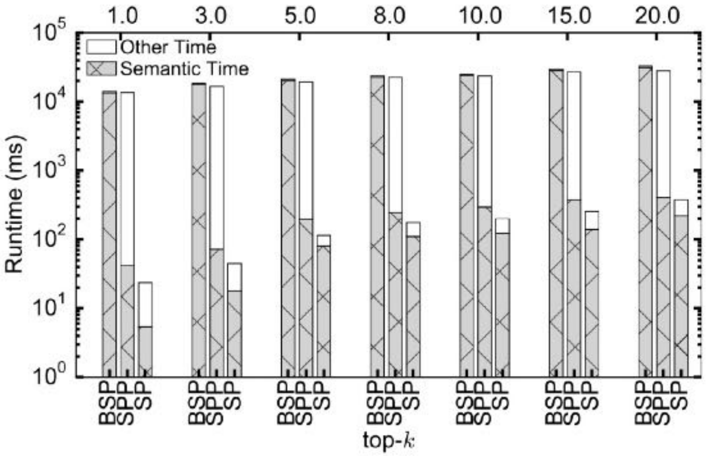



# Remarks

- Computation of spatial distances and spatial NN search is extremely cheap compared to graph browsing and incremental ranking of places by looseness
- This cost imbalance motivates the design of our algorithms and preprocessing techniques



runtime (DBpedia)



runtime (Yago)

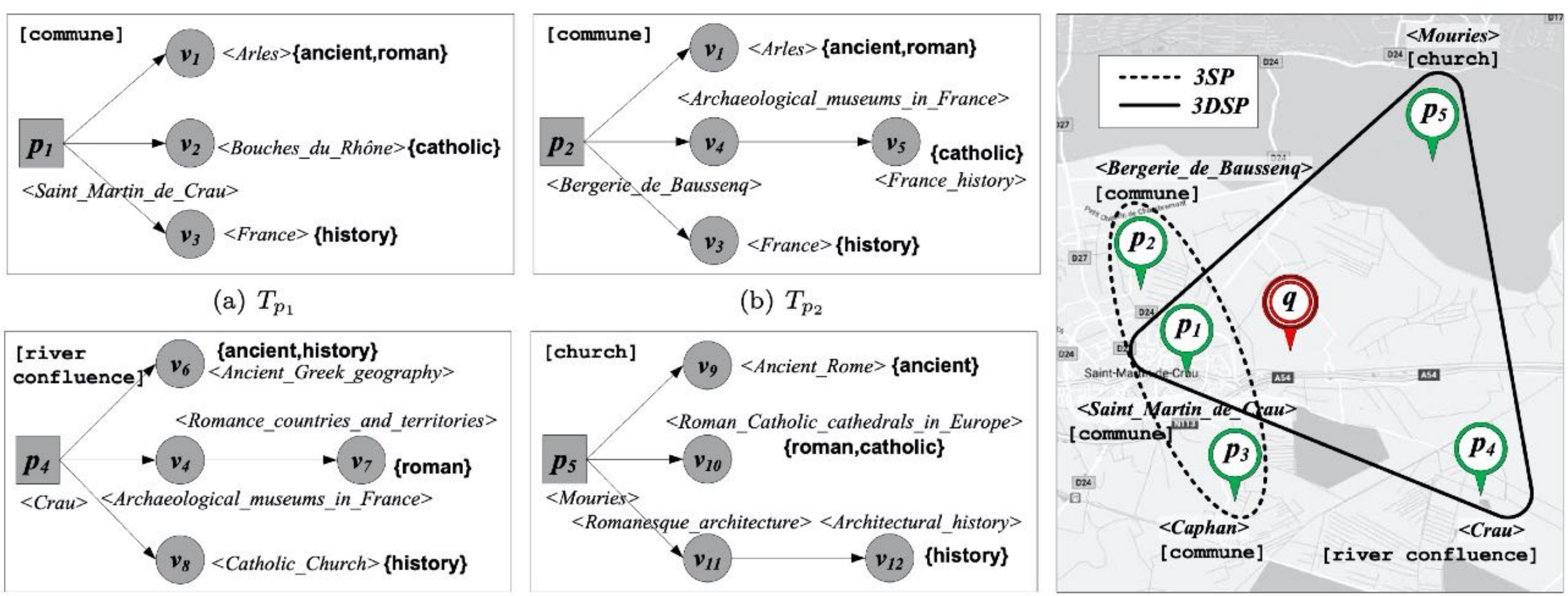
# Extensions:
# (1) support of temporal semantics

- Each RDF entity may be associated with a validity timestamp or interval
- Queries are associated with a search timestamp or interval
- A vertex is temporally relevant to the query if it is valid during the search period
  - Temporal relevance is considered when constructing the subgraph $T_p$ rooted at a candidate place p
- Inverted index is extended to contain temporal information
- Search bounds are revised to consider time

[WZSM20] D. Wu, H. Zhou, J. Shi, and N. Mamoulis, "Top-k Relevant Semantic Place Retrieval on Spatio-temporal RDF Data," *The **VLDB Journal***, 29(4): 893-917, July 2020.

# Extensions:
# (2) diversified search [CKF+20]

- Avoid putting in the query result two or more places
  - near each other and in the same direction (spatial diversity)
  - their $T_p$'s share many graph vertices (contextual diversity)
- NP-hard problem
- Efficient greedy algorithms with quality guarantees




(a) $T_{p_1}$ (b) $T_{p_2}$

(c) $T_{p_4}$ (d) $T_{p_5}$ (e) Map of Places

# Extensions:
# (3) proportionality in search [KFM21]

- Extend diversification to choose places proportionally based on spatial direction and context
- Applicable to different definitions of context (not just RDF subgraphs)
- Efficient algorithms for proportional subset selection:
  - Contextual
  - Spatial
- Used within a greedy search framework

***Spatial OS$_1$***

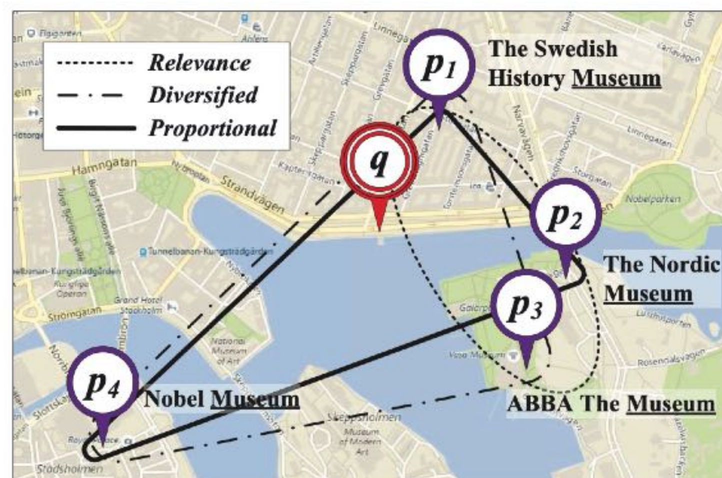
**Place:** Swedish History Museum
**Year established:** 1866
**Type:** History museum
**Type:** Nordic museum
**Type:** National museum
**Collection size:** 10 million
**Director:** K. Hauptman
**Opening days:** Everyday
**Collection:** Archaeological
**Collection:** Viking collection
**Collection:** Jewellery works
...

***Spatial OS$_2$***

**Place:** The Nordic Museum
**Year established:** 1873
**Type:** History museum
**Type:** Nordic museum
**Collection size:** 1.5 million
**Location:** Stockholm
**Opening days:** Eveyday
**Collection:** Buildings
**Collection:** Farms
**Collection:** Viking collection
**Collection:** Jewellery works
...

***Spatial OS$_4$***

**Place:** Nobel Museum
**Year established:** 2001
**Type:** Natural science
**Type:** Literature museum
**Type:** Peace museum
**Collection size:** 3500
**Director:** Erika Lanner
**Opening days:** Everyday
**Collection:** Discovery
**Collection:** Laureates works
**Collection:** Photos
...



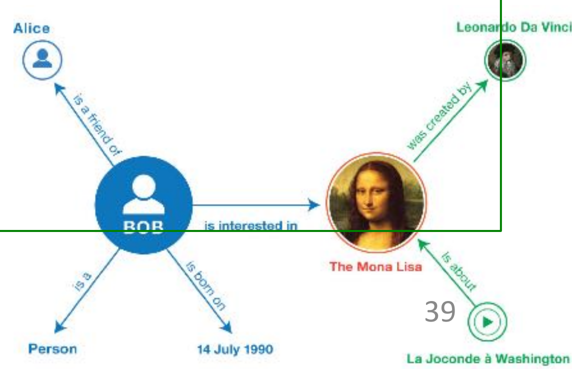
# Outline

- RDF data and the Semantic Web
- Spatially enriched RDF data
- Spatial RDF data management

  [LMBT14, TLM+19]
- Keyword search on Spatial RDF data

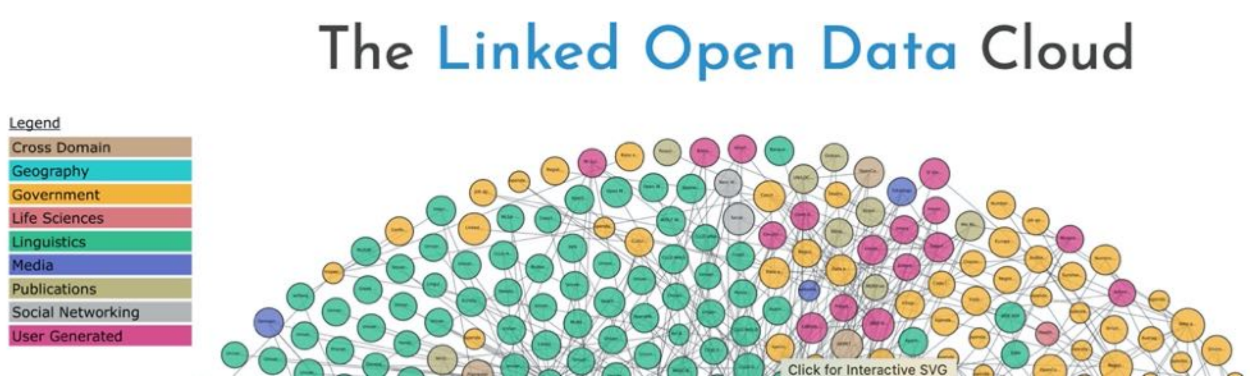
  [SWM16,  CKF+20, WZSM20, KFM21]
- Linking geospatial data

  [KVM20, PMMK21]



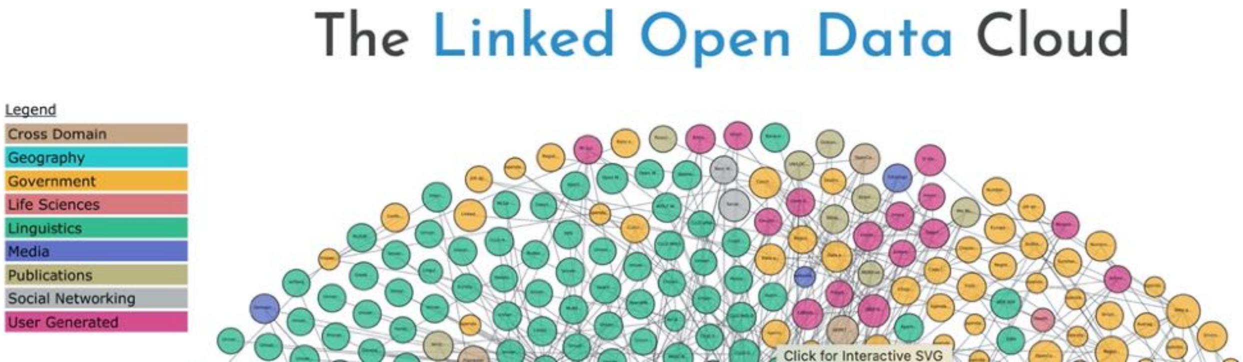
# Link Discovery

- Web of Data (WoD)
  - A global data space where entities across the web are more discoverable and easier reusable
- Link Discovery Problem
  - Establish links between entities in two different datasets for which some relation exists (e.g., *sameAs*, *seeAlso*)



# Related Classes for Link Discovery

- Fact: The **Linked Open Data** Cloud is huge
  - **1,255** datasets with **16,174** links (as of May 2020)
- Data providers usually link their data only with well-known datasets (DBpedia, Geonames)
- Entities in a dataset can be divided to classes
- Problem: find pairs of classes likely to include many pairs of linked entities



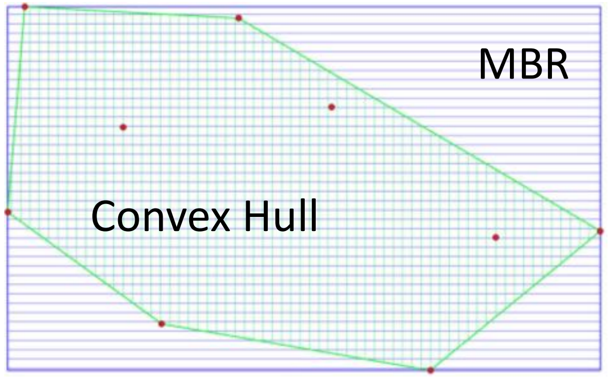
# Spatial search for related classes

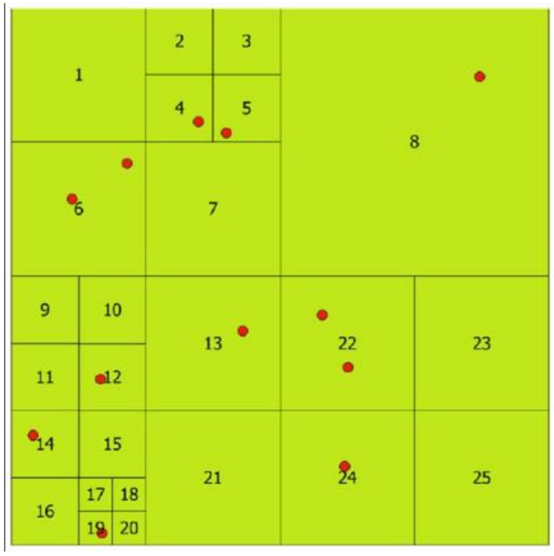
- Hypothesis: Pairs of classes with similar spatial distribution are likely to include semantically related instances
- Approach:
  - compute a degree of **geospatial relatedness** between classes; use pairs of large relatedness as candidates for link discovery
  - summarizes classes' spatial distributions for efficient computation of geospatial relatedness

[KVM20] V. Kopsachilis, M. Vaitis, N. Mamoulis, and D. Kotzinos, "Recommending Geo-semantically Related Classes for Link Discovery," ***Journal on Data Semantics***, 9(4): 151-177, December 2020.

# Computing geospatial relatedness

- Filter: if MBRs or CHs of class pairs do not overlap, consider them as irrelevant
- Approximation: summarize class by the IDs of quadrants occupied by its instances
  - Determine quadtree based on objects from all classes
- Compare class summaries to determine relatedness

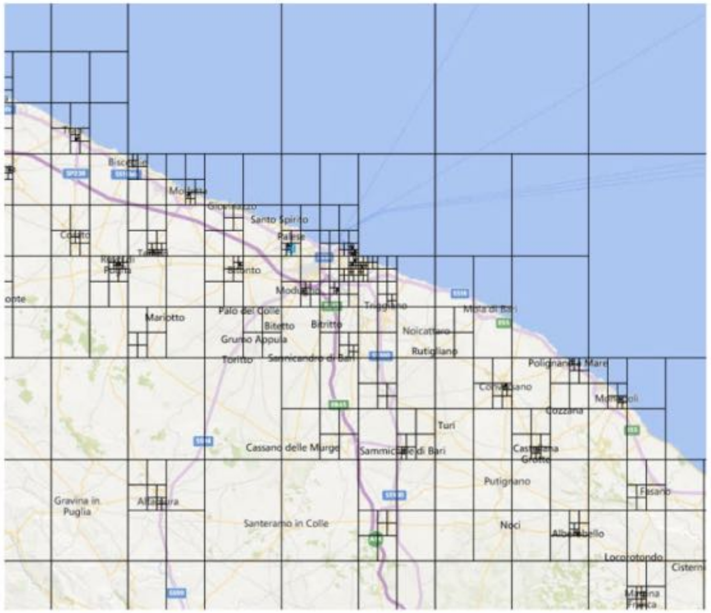



$S = \{4,5,6,8,12,13,14,19,22,24\}$

# Class Recommender for link discovery

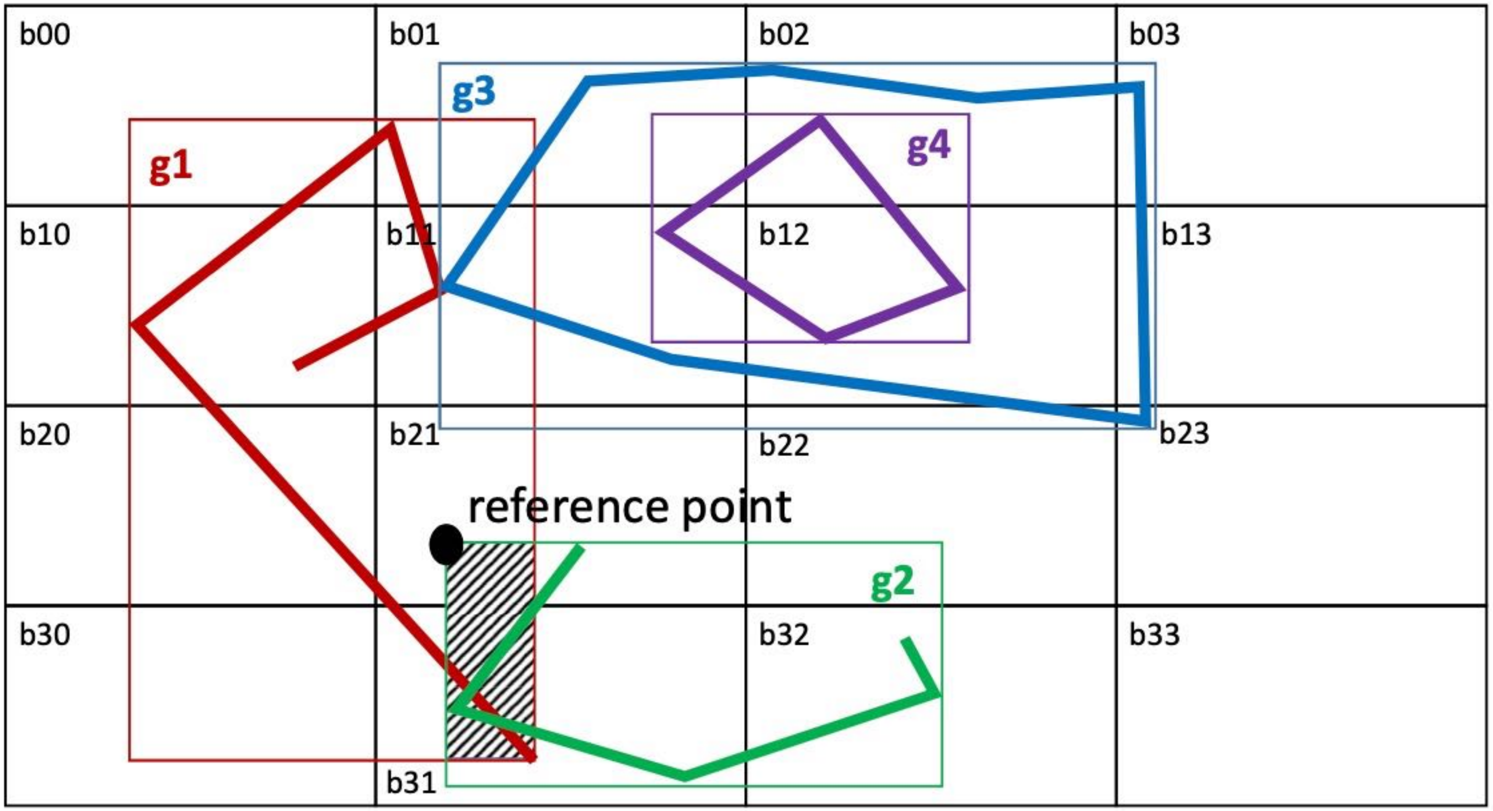
- *Spatial Class Locator* identifies and catalogs available spatial classes in the Web of Data
- *QuadTree Construction* generates a QuadTree summarizing the distribution of all classes
- *Spatial Class Summarization* creates summaries for the catalogued classes
- *Recommendation Algorithm* recommends a ranked list of classes to a source class



# Geospatial Interlinking

- Problem: Identify topological relations between geometries from two classes (datasets)
  - Facilitates geospatial link discovery
- Challenge: number of candidate pairs is $O(n^2)$
- Previous work (Space tiling)
  - divide space using a grid
  - only pairs of objects which co-occur in tiles are candidates for a topological relation (e.g. *inside*)
  - verify candidate pairs by computing their spatial relation

# Space tiling




LineString $g_1$ *intersects* LineString $g_2$

LineString $g_1$ *touches* Polygon $g_3$

Polygon $g_3$ *touches* Polygon $g_4$

# Limitations of previous work [Silk-spatial, RADON]

- Only one topological relation is evaluated for each pair of input classes
- Pairs are verified at no particular order
- Our contributions:
  - Holistic Geospatial Interlinking
  - Progressive Geospatial Interlinking (pay-as-you-go)
    - weighting schemes to define the best order
  - Parallel Geospatial Interlinking

[PMMK21] G. Papadakis, G. Mandilaras, N. Mamoulis, and M. Koubarakis, "Progressive, Holistic Geospatial Interlinking," *30th International Conference on World Wide Web (**WWW**)*, Ljubljana, Slovenia, April 2021.

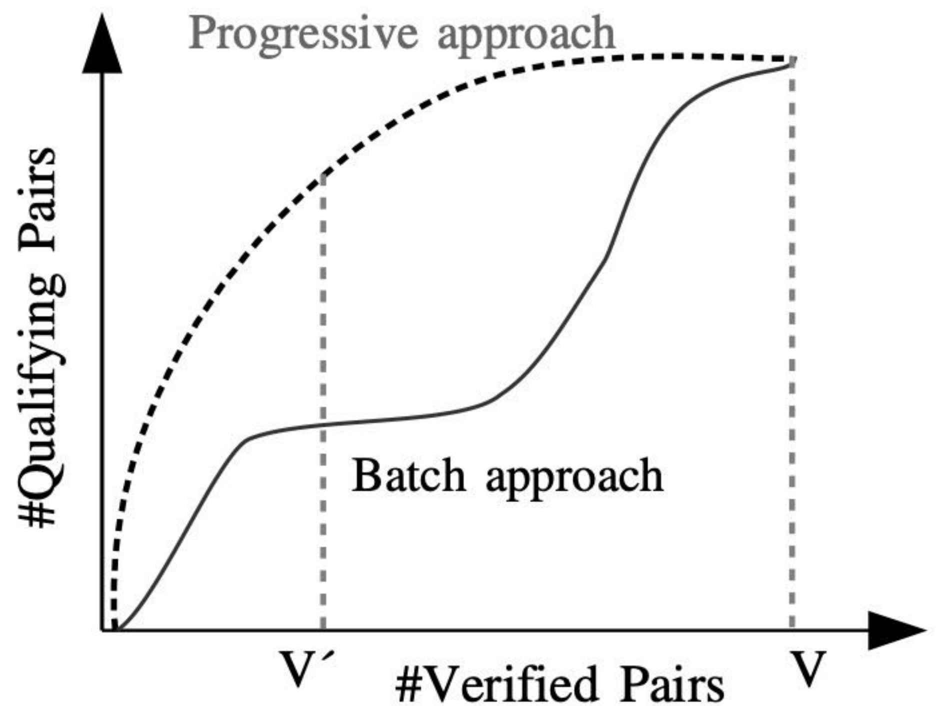
# Holistic Geospatial Interlinking

- Problem: Given a source dataset $S$, a target dataset $T$, and the set $R$ of topological relations, derive the set of all links
  - $LR = \{(s,r,t) \mid s \in S \wedge t \in T \wedge r \in R \wedge r(s,t)\}$
- Geospatial Interlinking At large (GIA.nt)
  - Index smallest dataset using a grid
  - For each geometry g in the other dataset:
    - Find grid cells that intersect with g and verify all relations for all objects in them
  - Lower space complexity compared to RADON
  - Faster filtering step compared to RADON
  - Examines each pair just once for all relations

# Progressive Geospatial Interlinking

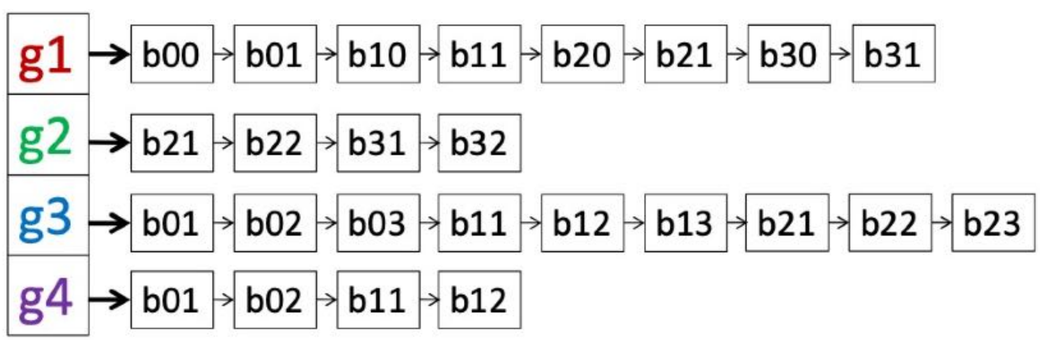
- Idea: Define a verification order for pairs
  - pairs with higher probability to satisfy a relationship go first
- Same eventual quality, improved early quality
- Measure: Progressive Geometry Recall (PGR)

$$PGR(R) = \sum_{i=1}^{|P|} p_Q^i / |P_Q^{BU}|$$



# Progressive Geospatial Interlinking

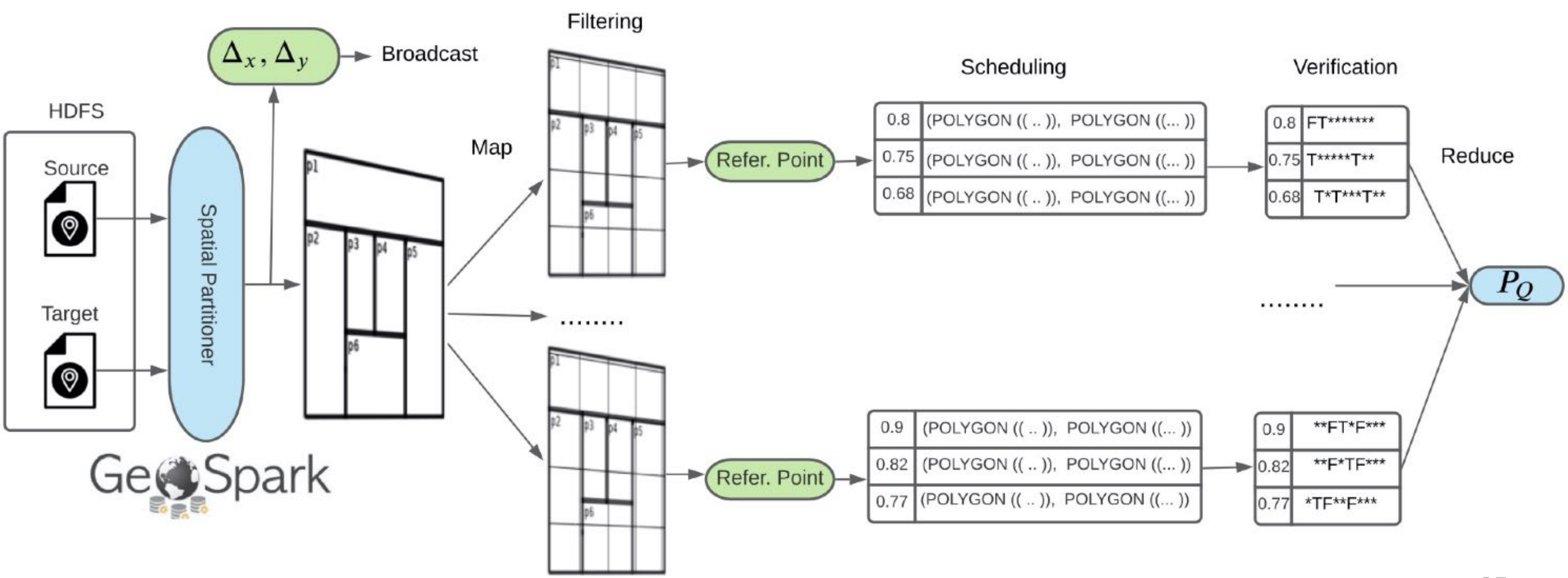
- Methodology:
  - Find all pairs (s,t) that pass the filter step and add them to a priority queue
    - Prioritization based on probability to pass verification step
    - Several hit probability weighting schemes evaluated
      - Number of tiles shared by s and t, Jaccard similarity, etc.
    - A *geometry index* is used to compute hit probabilities




  - Progressively verify top pairs

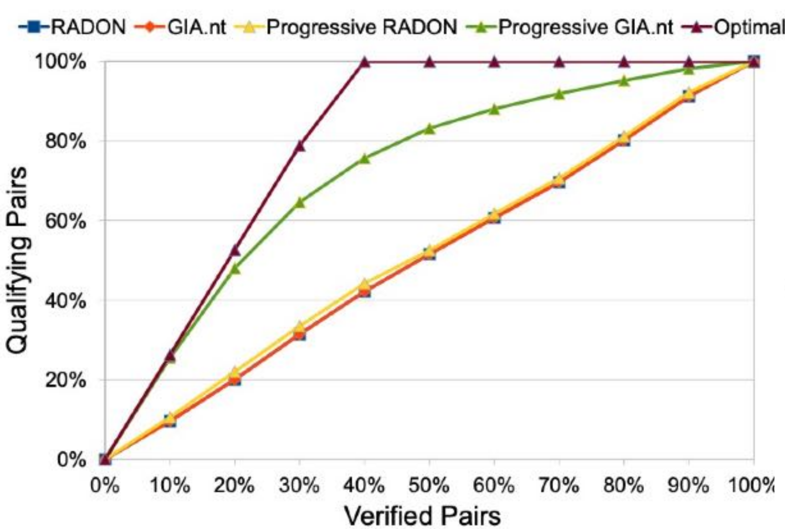
# Parallel Progressive GIA.nt

- load both datasets as RDDs and partition them using the same QuadTree partitioner
- Join RDDs with the same partition-id
- Divide budget among data partitions proportionally to load
- Aggregate all qualifying pairs during the Reduce phase

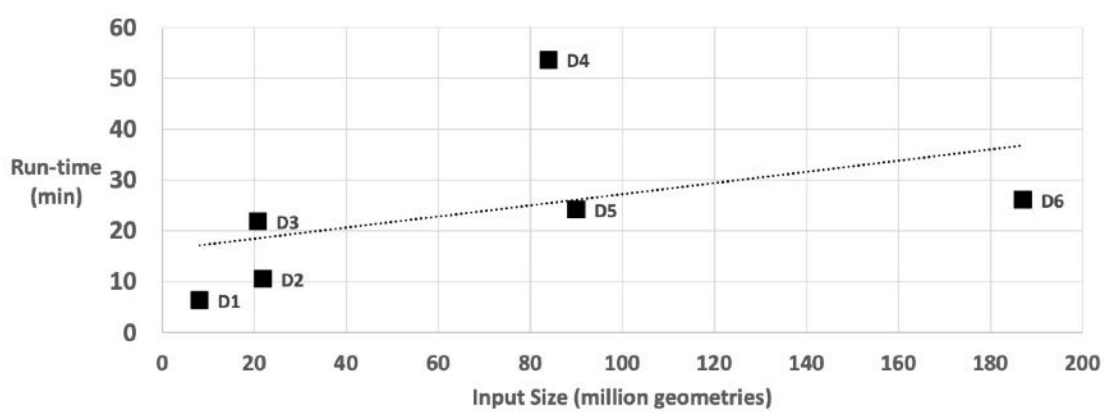


# Experimental Results

- We join pairs of public datasets from spatialhadoop.cs.umn.edu
- Verification of pairs is the bottleneck
- Holistic verification reduces overall cost by 80%
- Progressive version of GIA.nt is very effective
- Parallel processing scales gracefully




progressiveness




parallel processing

# Conclusions

- Huge volumes of semantic data available
- Significant presence of spatial information in data
- Need for efficient support of queries with spatial semantics (GeoSPARQL, spatial keyword search)
- Need for continuous enrichment and growth of existing data
  - by link discovery between different datasets
  - by geospatial interlinking of different datasets

# References


[KFM21] G. Kalamatianos, G. J. Fakas, and N. Mamoulis, "Proportionality in Spatial Keyword Search," Proceedings of the *ACM Conference on Management of Data (**SIGMOD**)*, Xi'an, China, June 2021.

[PMMK21] G. Papadakis, G. Mandilaras, N. Mamoulis, and M. Koubarakis, "Progressive, Holistic Geospatial Interlinking," *30th International Conference on World Wide Web (**WWW**)*, Ljubljana, Slovenia, April 2021.

[KVM20] V. Kopsachilis, M. Vaitis, N. Mamoulis, and D. Kotzinos, "Recommending Geo-semantically Related Classes for Link Discovery," ***Journal on Data Semantics***, 9(4): 151-177, December 2020.

[CKF+20] Z. Cai, G. Kalamatianos, G. J. Fakas, N. Mamoulis, and D. Papadias, "Diversified spatial keyword search on RDF data," *The **VLDB Journal***, 29(5): 1171-1189, September 2020.

[WZSM20] D. Wu, H. Zhou, J. Shi, and N. Mamoulis, "Top-k Relevant Semantic Place Retrieval on Spatio-temporal RDF Data," *The **VLDB Journal***, 29(4): 893-917, July 2020.

[TLM+19] K. Theocharidis, J. Liagouris, N. Mamoulis, P. Bouros, and M. Terrovitis, "SRX: Efficient Management of Spatial RDF Data," *The **VLDB Journal***, 28(5): 703-733, October 2019.

[SWM16] J. Shi, D. Wu, and N. Mamoulis "Top-k Relevant Semantic Place Retrieval on Spatial RDF Data," Proceedings of the *ACM Conference on Management of Data (**SIGMOD**)*, pp. 1977-1990, San Francisco, CA, June 2016.

[LMBT14] J. Liagouris, N. Mamoulis, P. Bouros, and M. Terrovitis, "An Effective Encoding Scheme for Spatial RDF Data," *Proceedings of the VLDB Endowment (**PVLDB**)*, 7(12):1271-1282, 2014.