On the Effect of Geometries Simplification on Geo-spatial Link Discovery

Abdullah Fathi Ahmed[1], Mohamed Ahmed Sherif[1,2], and Axel-Cyrille Ngonga Ngomo[1,2]

[1]Department of Computer Science, University Paderborn, 33098 Paderborn, Germany
afaahmed@mail.uni-paderborn.de
{sherif|ngonga}@upb.de
[2] Department of Computer Science, University of Leipzig, 04109 Leipzig, Germany
{sherif|ngonga}@informatik.uni-leipzig.de

12 September, 2018

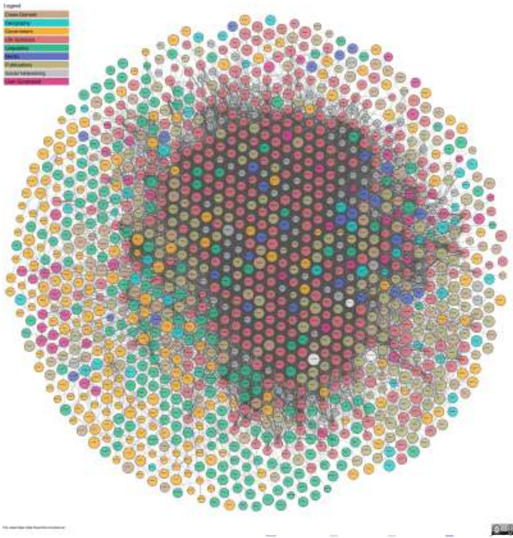
Outline

1. Motivation
2. Approach
3. Evaluation
4. Conclusion & Future Work

Motivation

Link Discovery among RDF geospatial data

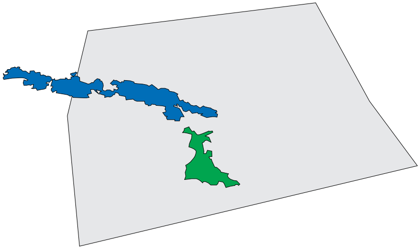
- Linked Data Cloud
  - `http://stats.lod2.eu`
  - 150+ billion triples
  - 46+ million links
  - Mostly `owl:sameAs`
- Large geospatial datasets
  - *LinkedGeoData* contains 20+ billion triples
  - *CLC* consists of 2+ million resources
  - *NUTS* contains up to 1500 points/resources



Motivation

Why is linking geospatial resources difficult?

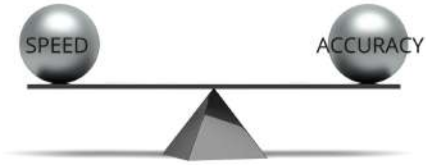
- Link Discovery
  - Given two knowledge bases S and T, find links of type $\mathcal{R}$ between S and T
  - Formally find $M = \{(s, t) \in S \times T : \mathcal{R}(s, t)\}$
  - Naïve computation of M requires quadratic time complexity
- Geo-spatial resources available on the LOD
  - Described using polygons
  - Large in number
  - Demands the computation of
    1. Topological relations
    2. point-set distance



Motivation

Why is linking of simplified geospatial data important?

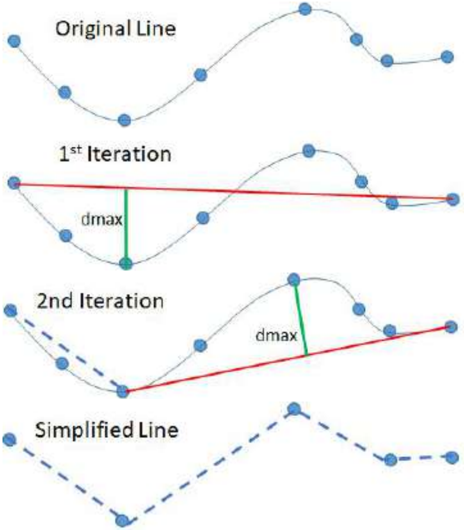
- Real-time applications
  - Structured machine learning
  - Cross-ontology QA
  - Reasoning
  - Federated Queries
  - ...
- The trade-off between
  - Runtime and
  - Accuracy



http://www.thepinsta.com

- **Input:** Polygonized curve with n vertices
- **Goal:** Find an approximating polygonized curve with m vertices, where $m < n$
- **Idea:** Approximate a line with a defined error tolerance $\epsilon > 0$
- Many algorithms exist
  - *Douglas-Peucker*
  - *Visvalingam-Whyatt*
  - ...

- Constract a line segment from the first point to the last point
- Find the point with farthest distance *dmax* from the line segment
- If ϵ *tolerance* $<$ *dmax* , the approximation is accepted
- otherwise, keep recursion



- Standard to describe the topological relations in 2D space.
- DE-9IM is based on the intersection matrix:

$$DE9IM(a,b)\begin{bmatrix} dim(I(g_1)\cap I(g_2)) & dim(I(g_1)\cap B(g_2)) & dim(I(g_1)\cap E(g_2)) \\ dim(B(g_1)\cap I(g_2)) & dim(B(g_1)\cap B(g_2)) & dim(B(g_1)\cap E(g_2)) \\ dim(E(g_1)\cap I(g_2)) & dim(E(g_1)\cap B(g_2)) & dim(E(g_1)\cap E(g_2)) \end{bmatrix}$$

- At least one shared point for a relation to be hold
- For the `disjoint` relation $\Rightarrow$ inverse of the `intersects` relation
- Accelerates the computation of any topological relation

- **Input** Two resources with input geometries g_s and g_t
- Compute the *orthodromic distance* $\delta(s_i, t_j)$ between pairwise point of g_s and g_t

$$\delta(s_i, t_j) = R\cos^{-1}\sin(\varphi_{s_i})\sin(\varphi_{t_j}) + \cos(\varphi_{s_i})\cos(\varphi_{t_j})\cos(\lambda_{s_i} - \lambda_{t_j}),$$

  p_i is a point on the surface (φ_i, λ_i), latitude φ_i and longitude λ_i,
- Many methods exist to compute the (g_s, g_t) point-set distance
  - Hausdorff

$$D_{Hausdorff}(g_s, g_t) = \max_{s_i \in g_s}\left\{\min_{t_j \in g_t}\left\{\delta(s_i, t_j)\right\}\right\}$$

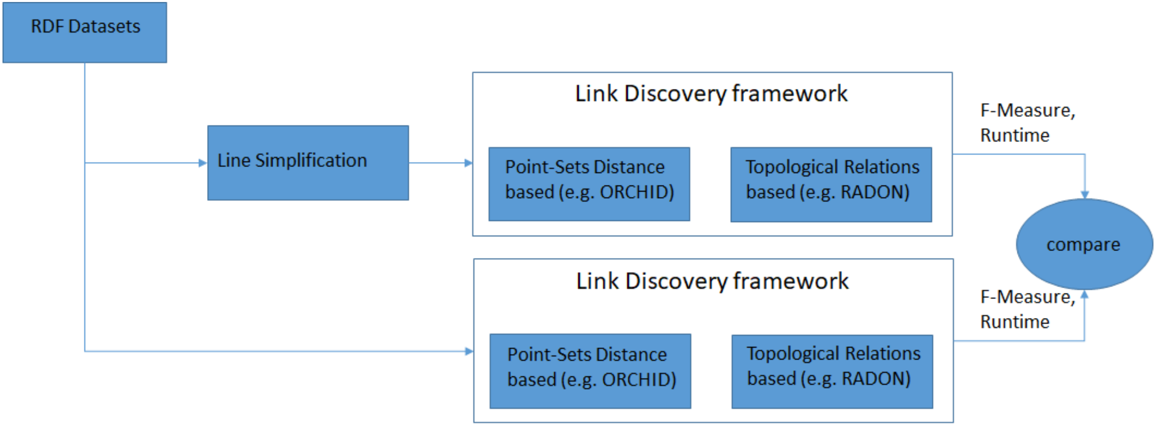
  - Mean

$$D_{mean}(g_s, g_t) = \delta\left(\frac{1}{n}\sum_{s_i \in g_s} s_i, \frac{1}{m}\sum_{t_j \in g_t} t_j\right)$$

- Line Simplification is independent from Link Discovery framework
- Stat of the art:
  - RADON for topological relation extraction
  - ORCHID for point set distance
- Topological relations:
  - `within`, `touches`, `overlaps`, `intersects`, `equals`, `crosses` and `covers`
- Point-sets distance:
  - `Hausdorff`, `Mean`, `Link`, `Min`, and `Sumofmin`

Evaluation

Overview



- Hardware
  - Oculus a cluster machine running OpenJDK 64-Bit 1.8.0161 on *Ubuntu* 16.04.3 LTS
  - Assigned 16 CPU (2.6 GHz Intel Xeon "Sandy Bridge") and 200 GB of RAM with timeout limit of 4 hours for each job
- Datasets
  - NUTS
  - CORINE Land Cover (CLC)

Q_1

How much performance (i.e., F-measure) each of the geospatial LD approaches loses, when to deal with the simplified geometries vs. when to deal with the original ones?

- The first set of experiments setup
  - RADON for discovering topological relations
  - *Douglas-Peucker* with simplification factors of $0.05, 0.09, 0.10$ and 0.2
  - NUTS dataset as a source and CLC dataset as a target

Relation/Factor	0.05	0.09	0.10	0.20	Average
Equals	1.00	1.00	1.00	1.00	1.00 ± 0.00
Intersects	0.99	0.97	0.97	0.94	0.97 ± 0.02
Contains	0.99	0.97	0.97	0.93	0.97 ± 0.03
Within	0.99	0.97	0.97	0.93	0.97 ± 0.03
Covers	0.99	0.97	0.97	0.93	0.97 ± 0.03
Coveredby	0.99	0.97	0.97	0.93	0.97 ± 0.03
Crosses	1.00	1.00	1.00	1.00	1.00 ± 0.00
Touches	1.00	1.00	1.00	1.00	1.00 ± 0.00
Overlaps	0.80	0.52	0.47	0.28	0.52 ± 0.21
Average	0.97 ± 0.07	0.94 ± 0.16	0.93 ± 0.17	0.90 ± 0.23	0.94 ± 0.03

F-measures results of applying RADON against geometries generated using the *Douglas-Peucker* line simplification algorithm.

Q_1

How much performance (i.e., F-measure) each of the geospatial LD approaches loses, when to deal with the simplified geometries vs. when to deal with the original ones?

- The second set of experiments setup:
  - ORCHID for measuring the distance between point-sets
  - *Douglas-Peucker* with simplification factors of $0.05, 0.09, 0.10$ and 0.2
  - NUTS dataset dedublicated to compute F-measure
  - NUTS dataset as a source and as a target (deduplication)

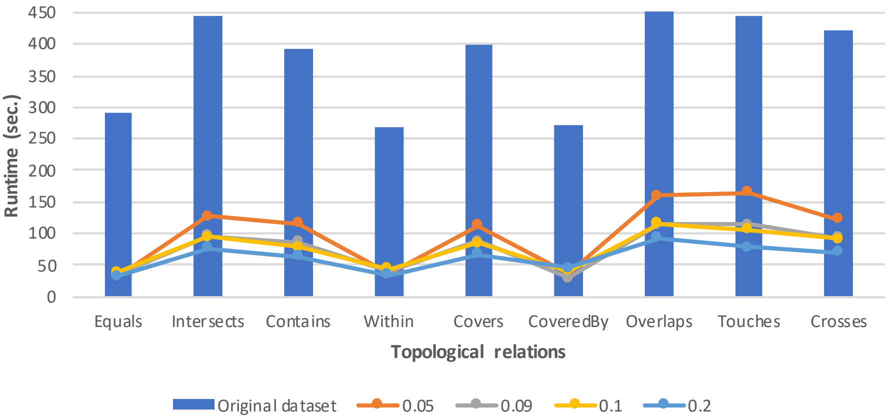
Measure/Factor	0.05	0.9	0.1	0.2	0.3	Average	$F_{original}$
Hausdorff	0.90	0.91	0.91	0.91	0.91	0.91 ± 0.00	0.88
Mean	0.94	0.94	0.94	0.94	0.94	0.94 ± 0.00	0.94
Min	0.14	0.16	0.16	0.21	0.25	0.18 ± 0.04	0.13
Link	0.95	0.95	0.94	0.94	0.94	0.94 ± 0.00	0.94
SumOfMin	0.95	0.95	0.94	0.94	0.94	0.94 ± 0.00	0.94
avarege	0.77 ± 0.36	0.78 ± 0.35	0.78 ± 0.35	0.79 ± 0.32	0.80 ± 0.31		0.77 ± 0.36

F-measures results of the point-set distance measures implementations in ORCHID using the *Douglas-Peucker* line simplification algorithm.

Q_2

How well each of the geospatial LD approaches scale (i.e., runtime speedup), and when to deal with the simplified geometries?

- The third sets of experiments setup:
  - Same setting as in the first sets of experiments
    - RADON for discovering topological relations
    - *Douglas-Peucker* with simplification factors of $0.05, 0.09, 0.10$ and 0.2
    - NUTS dataset as a source and CLC dataset as a target



Runtimes of RADON's implementation of topological relations LD using the *Douglas-Peucker* algorithm

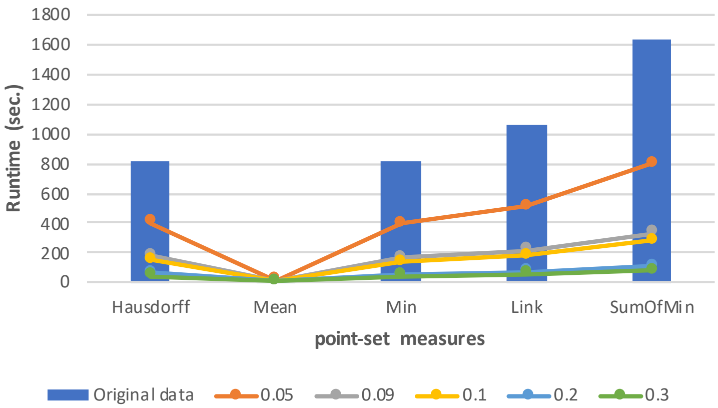
Q_2

How well each of the geospatial LD approaches scale (i.e., runtime speedup), and when to deal with the simplified geometries?

- The fourth set of experiments:
  - Same setting in the second sets of experiments
    - ORCHID for measuring the distance between point-sets
    - *Douglas-Peucker* with simplification factors of $0.05, 0.09, 0.10$ and 0.2
    - NUTS dataset dedublicated to compute F-measure
    - NUTS dataset as a source and as a target (deduplication)

Evaluation

Runtime Analysis



Runtimes of ORCHID's implementation of set-points distance measure LD using the *Douglas-Peucker* algorithm

Q_3

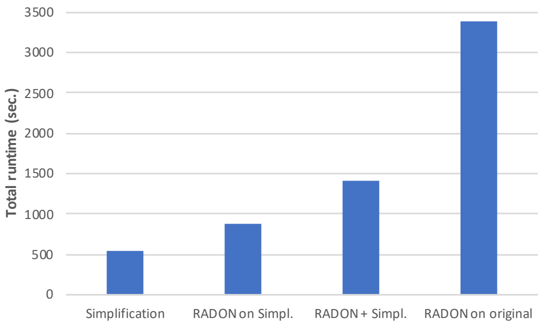
Which relation is the most/least affected by the simplification process?

- Same setting in the first and second sets of experiments
- The relation `overlaps` has the most affected F-measure when using *Douglas-Peucker* algorithm, (see Table 14)
- The `equals`,`crosses` and `touches` are not affected at all by any simplification (see Tables 14
- In the case of point-set measures, the F-measure of `min` measure is the most affected, (see Table 16)

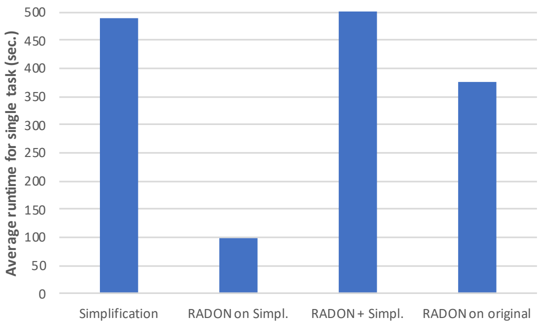
Q_4

What is the run time cost of simplification?

- Same setting in the third and fourth sets of experiments



Total Runtime for all topological relations.



Average Runtime of a single topological relation.

Runtime of RADON on the original data vs. simplified data using the *Douglas-Peucker* algorithm.

Conclusion & Future Work

- Conclusion
  - Studied the usage of line simplification as a preprocessing step of LD approaches over geospatial RDF knowledge bases
  - Studied the behaviour of two categories of geospatial linking approaches (i.e., the topological relations and point-set distances)
  - On average, F-measure of 0.94 using the *Douglas-Peucker* algorithm and 0.69 using the *Visvalingam-Whyatt* algorithm has been achieved.
  - Gain up to 19.8$\times$ speedup using *Douglas-Peucker* algorithm and up to 67.3$\times$ using the *Visvalingam-Whyatt*
- Future Work
  - Guarantee the minimum F-measure loss
  - Determine fittest line simplification algorithm and its best parameter to achieve a better trad-off between F-measure and Runtime

Thank you for your Attention!

Abdullah Fathi Ahmed
`afaahmed@mail.uni-paderborn.de`
`https://github.com/dice-group/Orchid`

Acknowledgment

This work has been supported by LEDS project, Eurostars Project SAGE (GA no. E!10882), the H2020 projects SLIPO (GA no. 731581) and HOBBIT (GA no. 688227) as well as the DFG project LinkingLOD (project no. NG 105/3-2) and the BMWI Project GEISER (project no. 01MD16014).